



# A Framework for Autonomous Wind Farms: Wind Direction Consensus

Jennifer Annoni<sup>1</sup>, Christopher Bay<sup>1,2,3</sup>, Kathryn Johnson<sup>1,2</sup>, Emiliano Dall’Anese<sup>3</sup>, Eliot Quon<sup>1</sup>,  
Travis Kemper<sup>1</sup>, and Paul Fleming<sup>1</sup>

<sup>1</sup>National Renewable Energy Laboratory, Golden, CO, 80401, USA

<sup>2</sup>Colorado School of Mines, Golden CO, 80401, USA

<sup>3</sup>University of Colorado - Boulder, Boulder, CO, 80309, USA

*Correspondence to:* Jennifer Annoni ([jennifer.annoni@nrel.gov](mailto:jennifer.annoni@nrel.gov))

## Abstract.

Wind turbines in a wind farm typically operate individually to maximize their own performance and do not take into account information from nearby turbines. In an autonomous wind farm, enabling cooperation to achieve farm-level objectives, turbines will need to use information from nearby turbines to optimize performance, ensure resiliency when other sensors fail, and adapt to changing local conditions. A key element of achieving an autonomous wind farm is to develop algorithms that provide necessary information to ensure reliable, robust, and efficient operation of wind turbines in a wind plant using local sensor information that is already being collected, such as supervisory control and data acquisition (SCADA) data, local meteorological stations, and nearby radars/sodars/lidars. This article presents a framework for implementing an autonomous wind farm that incorporates information from local sensors in real time to better align turbines in a wind farm. Oftentimes, measurements made at an individual turbine are noisy and unreliable. By incorporating measurements from multiple nearby turbines, a more robust estimate of the wind direction can be obtained at an individual turbine. Results indicate that this estimate of the wind direction can be used to improve the turbine’s knowledge of the wind direction and could decrease dynamic yaw misalignment, decrease the amount of time a turbine spends yawing due to a more robust input to the yaw controller, and increase resiliency to faulty wind-vane measurements.

## 15 1 Introduction

The wind industry continues to seek methods to decrease the levelized cost of energy (LCOE) by using advances in science, engineering, and computation (Lindenberg (2009)). Control systems have the potential to contribute to this LCOE reduction by incorporating local measurements and optimizing large wind farms in real time. Specifically, control systems can be used to achieve an autonomous wind plant, which includes turbines that self-organize into groups, monitor their status and the status of other turbines, and control their performance to maximize the economic and reliable performance of a large-scale wind plant (Kroposki et al. (2017)). A framework for autonomous wind farms can be achieved by representing a wind farm as a network of wind turbines. The network topology can be used to advance the state of the art in wind farm controls in topics ranging from distributed optimization and control to fault detection and short-term forecasting. In this sense, wind farms can



take advantage of the network topology to implement scalable, reconfigurable, and resilient control strategies in real time. The approach presented in this article is a robust algorithm that takes advantage of the topology of a wind farm and incorporates local measurements from nearby turbines to determine the wind direction at an individual turbine in real time. Improving the wind direction measurement at the turbine can reduce unnecessary yaw movements and minimize dynamic yaw misalignments.

5     Currently, turbines typically rely on wind vanes and anemometers mounted on the back of the nacelle to provide measurements to their controllers. However, these measurements are often unreliable due to the complex flow created as the wind passes through the rotor, preventing accurate inputs into the individual turbine yaw controller. One way to address noisy wind direction information is to mount a forward-facing lidar on the nacelle to detect the wind in front of the turbine (Fleming et al. (2014b); Schlipf et al. (2013); Scholbrock et al. (2013); Simley et al. (2014)). In addition, meteorological (met) towers can  
10 be used to characterize the inflow; however, most turbines do not have dedicated met towers and the wind direction can vary across a wind farm due to variable meteorological conditions and topography.

This paper describes a distributed optimization-based method to robustly estimate the wind direction across a wind farm. Distributed optimization and control theory provide a framework for efficient computation of large systems, especially systems with network topologies. These types of optimization strategies have been used for multiagent systems, such as unmanned  
15 aerial vehicles and robots, and can be used to coordinate subsystems to interact with their larger environment (Zhu and Martínez (2015); Ferrari et al. (2016); Movric and Lewis (2014); Shamma (2008)). Distributed optimization has also been considered in the wind farm controls literature (Marden et al. (2012); Spudić et al. (2015)). However, complex aerodynamic interactions and large timescales make this a challenging problem. For example, a distributed optimization framework for wind farm controls has been presented in (Soleimanzadeh et al. (2013)). However, solving this problem becomes computationally complex  
20 as the system grows because of the number of turbines and larger flow domains. A limited-communication distributed model predictive controller designed to track a power reference signal is described in Bay et al. (2018); this algorithm uses a simplified linearized wake model to describe turbine interactions, allowing for scalability. Because this method requires a linear model, this method is difficult to extend to power maximization or load minimizations where the objective functions are highly nonlinear.

25     Consensus-based algorithms are a specific class of optimization algorithms that have the potential to accommodate sensor errors caused by failure, miscalibration, and noise by assuming that turbines experience wind inflow directions that share similar characteristics with that of their neighbors. Although consensus optimization is an active area of research for many applications, especially in multiagent systems, its use in wind farm applications is relatively new. A few studies have used consensus-based approaches for various problems in wind energy. In one such study, a dynamic average consensus estimator is  
30 used in Ebegbulem and Guay (2017) to estimate an overall cost function for turbines communicating via an undirected network where the goal is to maximize total wind farm power production. Similar power maximization approaches using consensus-based approaches for an undirected graph can be found in Wang et al. (2017); Gionfra et al. (2017). Finally, the research of Baros and Ilic (2017) allows turbines to self-organize using torque control and storage to regulate total wind farm power output.

This article presents a consensus-based distributed optimization algorithm for robustly calculating wind direction at a wind  
35 turbine using only supervisory control and data acquisition (SCADA) data from the turbines in the wind farm. This robust wind



direction signal can be used as an input to a turbine yaw controller or can be used to facilitate wake steering wind farm control (Fleming et al. (2014a)). It is important to note that this approach requires no additional sensing information. This algorithm is based on the work presented in Hallac et al. (2015) and is solved using alternating direction method of multipliers (Boyd et al. (2011)). Details of this algorithm are found in Section 2.3. A key contribution of this article is demonstrating this algorithm on wind farm SCADA data where the wind direction varies across the wind farm. For proprietary reasons, all the data has been normalized and only a subsection of the wind farm is shown (see Section 4). Results of this approach are compared with met towers and sodar on-site and are shown in Section 5. The results indicate that this approach provides a robust measurement of the wind direction at each turbine. Finally, Section 6 provides some conclusions and suggestions for future work.

## 2 Autonomous Wind Farm

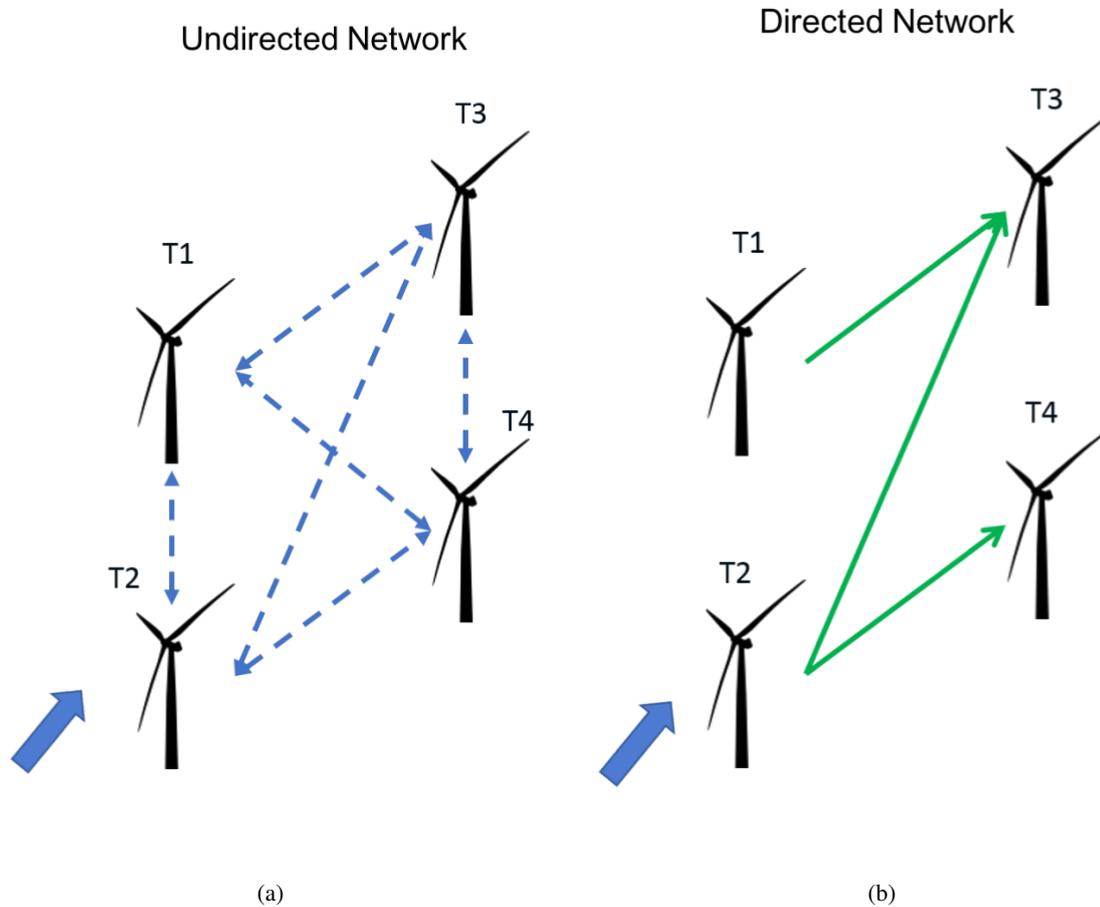
This section introduces the autonomous wind farm that self-organizes into groups, monitors, and controls its performance in real time based on existing SCADA data. In this framework, turbines can take advantage of data from nearby turbines to make more informed decisions that benefit the wind farm as a whole. This framework is scalable and reconfigurable and can be extended to include additional sensors incorporating nearby, relevant measurements from other turbines, meteorological instruments, and mobile sensors. Identifying a graph or network topology is important for incorporating local information and taking advantage of the structure of the wind farm to perform real-time optimization. The network topology described in this section and used to model the autonomous wind farm determines which turbines communicate with each other and can be used to implement an autonomous wind farm. The network connections can be based on proximity, aerodynamic interactions (wakes), or other metrics and this grouping is based on the objective of the developer/user. This study uses proximity to define the connections between turbines. Finally, this approach can solve local optimization problems and allows for local variations experienced in a wind farm.

### 2.1 Wind Farm as a Network

In an autonomous wind farm, the wind farm can be modeled as an undirected or a directed network where turbines communicate with nearby turbines. Turbines in a wind farm can be considered the nodes and the edges are established communication between nearby turbines. Information is communicated across these edges to determine local atmospheric conditions—such as wind direction or wind speed—at a particular turbine.

#### 2.1.1 Undirected Network

An undirected network is a network in which information is exchanged in both directions along an edge. A wind farm can be modeled as an undirected graph where turbines are communicating with connected turbines and information flows both ways, rather than from one turbine to the next turbine. An undirected graph is shown in Figure 1a. This plot shows turbines connected, that is communicating, with the other turbines, which in general might be determined to be all other turbines in the



**Figure 1.** (a) Example of an undirected network for a four-turbine wind farm. (b) Example of a directed network for a four-turbine wind farm.

farm or determined by some other feature such as distance. Modeling a wind farm as an undirected network allows for relevant spatial information to be used to determine the local atmospheric conditions.

It is important to note that although turbines typically communicate with a central computer to record SCADA data, the undirected network is used to determine which turbines to include while computing a local optimization at a particular turbine.

- This topology is designed to take advantage of temporal and spatial structures in a wind farm. For example, a turbine on the western most edge of the wind farm can be experiencing a different wind speed/direction than a turbine on the eastern edge of the wind farm that is several kilometers away.



## 2.1.2 Directed Network

A directed network is a network in which each edge has a direction and information flows in one direction from one node to another. A wind farm can also be modeled as a directed graph with information flowing from upstream turbines to downstream turbines as shown in Figure 1b. The turbine interactions, that is wakes, determine the network topology. Under this paradigm,  
5 the network topology is determined by current atmospheric conditions such as wind speed, and wind direction.

This article focuses on undirected network topologies. Autonomous wind farms consisting of directed networks will be a subject of future work.

## 2.2 Network Topology

As mentioned in Section 2.1, the network topology, i.e., graph structure, is important for incorporating local information and  
10 taking advantage of the structure of the wind farm to perform real-time optimizations. There are various ways to define the graph structure in a wind farm. A model-based approach can be used to determine the strength of aerodynamic interactions (Annoni et al. (2018)). A data-driven approach can be used to learn the interactions between turbines (Friedman et al. (2008)). Finally, turbine communications can be defined by the nearest  $N$  turbines. This article defines the graph structure based on the nearest  
15 10 turbines. Alternative approaches can be used to cluster turbines to optimally exchange information such as connectivity, hierarchical, or k-means algorithms (Xu and Wunsch (2005)). Additional ways to characterize the network topologies will be explored in future work.

There are a few important things to note when determining the network topology in a wind farm. First, the local conditions can vary across a wind farm. The number of connections between each turbine can determine the variability obtained across the wind farm. For example, if every turbine is connected to every other turbine, the variability across the wind farm will  
20 be small and the turbines will all agree on one set of atmospheric conditions based on a consensus approach. In addition, the computation time will be high due to the number of communication exchanges per iteration of the consensus algorithm. However, if the turbine has no connections, the variation of the output of the consensus algorithm across the wind farm will be high and possibly unreliable due to sensor noise and miscalibration. Smaller groups of turbines can agree on local conditions and provide a robust measurement that more accurately captures the variations across the wind farm.

## 25 2.3 Distributed Optimization for Real-Time Controls

Using an undirected or directed network topology described in Section 2.1, a distributed approach can be used to solve an optimization problem that takes advantage of the corresponding network topology. The problem can be decomposed such that each turbine can solve its own optimization problem, which incorporates information from connected turbines in the network topology. In other words, a few measurements from nearby turbines are used to solve the algorithm's optimization problem  
30 rather than solving a centralized optimization problem that includes all measurements from all turbines. Trying to incorporate all measurements from all turbines can be problematic in terms of communication limits and computational complexity, potentially taking hours to solve. A centralized solution can take hours to compute. However, grouping the wind farm provides a



computationally efficient algorithm for optimizing a particular objective function. The approach used in this article is based on Hallac et al. (2015) where the objective function can be written as:

$$\underset{x_i}{\text{minimize}} \quad \underbrace{\sum_{i \in \mathcal{V}} f_i(x_i)}_{\text{node objective}} + \underbrace{\sum_{(j,k) \in \mathcal{E}} g_{jk}(x_j, x_k)}_{\text{edge objective}} \quad i = 1, \dots, N_{\text{turbs}} \quad j \in \mathcal{N}(i) \quad (1)$$

where  $f_i(x_i)$  is the objective function at turbine  $i$ , i.e., the node objective,  $\mathcal{N}_i$ , indicates the turbines connected to turbine  $i$ ,  $x_i$  is the wind direction estimate at turbine  $i$ , and  $g_{jk}(x_j, x_k)$  compares wind direction measurements between turbines in  $\mathcal{N}$  in the wind farm network, i.e., the edge objective. The objective function,  $f_i(x_i)$ , is specified to improve the performance of a wind farm, whether that is to maximize power, minimize loads, or track a power reference. (Boersma et al. (2017); Annoni et al. (2018)). Each turbine is a node in  $\mathcal{V}$  and the nearest turbines are connected by edges in  $\mathcal{E}$ . The next section describes a wind direction consensus example to illustrate the utility of this framework.

### 10 3 Wind Direction Consensus

This study uses a consensus-based approach that uses the undirected network framework, introduced in Section 2.1, to robustly determine the wind direction at every turbine considering both the turbine's measurements and those of its nearest 10 neighbors. The SCADA data measurements recorded at each turbine are used to determine a robust measurement of wind direction at every turbine. This approach allows the wind direction and wind speed to vary across a wind farm. It is assumed that the wind directions recorded at the turbines are with reference to true north and that the wind direction varies smoothly across the wind farm.

#### 3.1 Node and Edge Objective Functions

For this problem, each turbine uses its own wind direction measurement as well as the wind direction measurement from the connected turbines to determine the local wind direction. First, the objective of the individual turbine  $i$ , i.e., node objective  $f_i(x_i)$ , is to minimize the error between the wind direction measurement measured at turbine  $i$  and the estimated wind direction,  $x_i$ .

$$f_i(x_i) = (\tilde{x}_{i, \text{measure}} - x_i)^2 \quad (2)$$

where  $x_{i, \text{measure}}$  is the wind direction measurement recorded at the turbine  $i$ . In this case, the objective function is convex and can be updated with a closed form solution (Boyd and Vandenberghe (2004)). In addition to the node objective, the edge objective incorporates information from nearby turbines to ensure a robust measurement of the wind direction at an individual turbine. The edge objective can be written as:

$$g_{jk}(x_j, x_k) = w_{jk} |x_j - x_k| \quad (3)$$



where  $w_{jk}$  is a weight placed on the connection between turbines,  $x_j$  is the estimated wind direction at turbine  $j$ , and  $x_k$  is the estimated wind direction at turbine  $k$ . The edge objective,  $g_{jk}(x_j, x_k)$ , minimizes the differences in estimated wind direction between neighboring turbines. In this paper, the weights,  $w_{jk}$ , are set to 1. However, different weights can be used to indicate the “trustworthiness” of a particular measurement. Future work will optimize the weighting between turbine communications to better integrate the data. Equations 2 and 3 are used by the optimization problem (1) for this wind direction consensus problem.

In addition to the communication and computational benefits of clustering a smaller number of turbines as opposed to optimizing across an entire wind farm, using clustering allows each subset optimization to be performed in parallel, further reducing computational time. An iterative approach is needed to solve this optimization problem and is detailed in the next section.

### 3.2 Alternating Direction Method of Multipliers

Alternating direction method of multipliers (ADMM) is a technique used to solve distributed optimization problems (Boyd et al. (2011)) such as (1). This algorithm is particularly useful in this case, as each individual turbine solves its own optimization in parallel, communicates the solution to neighboring subsets, and iterates this process until the wind farm has converged and each node has reached a single value. In this study, each turbine determines the local wind direction at each individual turbine by only talking to its nearest neighbors, as indicated in Section 2. ADMM is used to solve a network optimization with connecting nodes to determine a consensus between shared nodes such that:

$$\text{minimize} \quad \sum_i^{N_{turbines}} f_i(x_i) + \lambda \sum_{(j,k) \in \mathcal{E}} w_{jk} \|z_{jk} - z_{kj}\|_2 \quad (4)$$

$$\text{subject to} \quad x_i = z_{ij}, \quad j \in N(i) \quad (5)$$

where  $z_{jk}$  is a copy of  $x_j$  at turbine  $k$  such that the wind farm reaches consensus of the wind direction across the wind farm.

The distributed optimization problem is solved using ADMM by minimizing the augmented Lagrangian

$$\mathcal{L}_\rho(x, z, u) = \sum_{i \in \mathcal{V}} f_i(x_i) + \sum_{(j,k) \in \mathcal{E}} \lambda w_{jk} \|z_{jk} - z_{kj}\|_2 - (\rho/2) (\|u_{jk}\|_2^2 + \|u_{kj}\|_2^2) + (\rho/2) (\|x_j - z_{jk} + u_{jk}\|_2^2 + \|x_k - z_{kj} + u_{kj}\|_2^2) \quad (6)$$

where  $u$  is the scaled dual variable and  $\rho > 0$  is the penalty parameter. The following steps are used in an iterative way to solve (4):

$$x^{m+1} = \arg \min_x \mathcal{L}_\rho(x, z^m, u^m) \quad (7)$$

$$z^{m+1} = \arg \min_x \mathcal{L}_\rho(x^{m+1}, z, u^m) \quad (8)$$

$$u^{m+1} = u^m + (x^{m+1} - z^{m+1}) \quad (9)$$



where  $m$  indicates the iteration within the optimization problem. This problem is completely decomposable and can be separated based on turbines, i.e., one turbine optimization can be run on one node and run in parallel. This can significantly reduce computation time. The three updates,  $x$ ,  $z$ , and  $u$ , are performed according to (6) - (9):

First, the  $x$ -update is computed as:

$$5 \quad x_i^{m+1} = \arg \min \left( f_i(x_i) + \sum_{j \in \mathcal{N}(i)} (\rho/2) \|x_i - z_{ij}^m + u_{ij}^m\|_2^2 \right) \quad (10)$$

In this example, the  $x$  update, (10) is a convex problem and can be solved in closed form

$$x_i^{m+1} = \frac{2\tilde{x}_{measure} - \rho \sum_{j \in \mathcal{N}(i)} (-z_{ij}^m + u_{ij}^m)}{\rho N_{turb} + 2}. \quad (11)$$

If  $x$  is non-convex, then gradient-based optimization algorithms can be used to solve (10). However, there are no guarantees on convergence for a general gradient-based optimization concept. Hallac et al. (2015) presents an extension to address the nonconvexity of the problem to improve convergence metrics.

Next, the  $z$ -update:

$$z_{ij}^{m+1}, z_{ji}^{m+1} = \arg \min_{z_{ij}, z_{ji}} \left( \lambda w_{ij} \|z_{ij} - z_{ji}\|_2 + (\rho/2) \left( \|x_i^{m+1} - z_{ij} + u_{ij}^m\|_2^2 + \|x_j^{m+1} - z_{ji} + u_{ji}^m\|_2^2 \right) \right) \quad (12)$$

As with  $x$ , there is a closed-form analytical solution to the  $z$ -update for the almost consensus problem where  $z_{ij} \neq z_{ji}$ , shown in Hallac et al. (2015)

$$15 \quad z_{ij}^{m+1} = \theta (x_i^{m+1} + u_{ij}^m) + (1 - \theta) (x_j^{m+1} + u_{ji}^m) \quad (13)$$

$$z_{ji}^{m+1} = (1 - \theta) (x_i^{m+1} + u_{ij}^m) + \theta (x_j^{m+1} + u_{ji}^m) \quad (14)$$

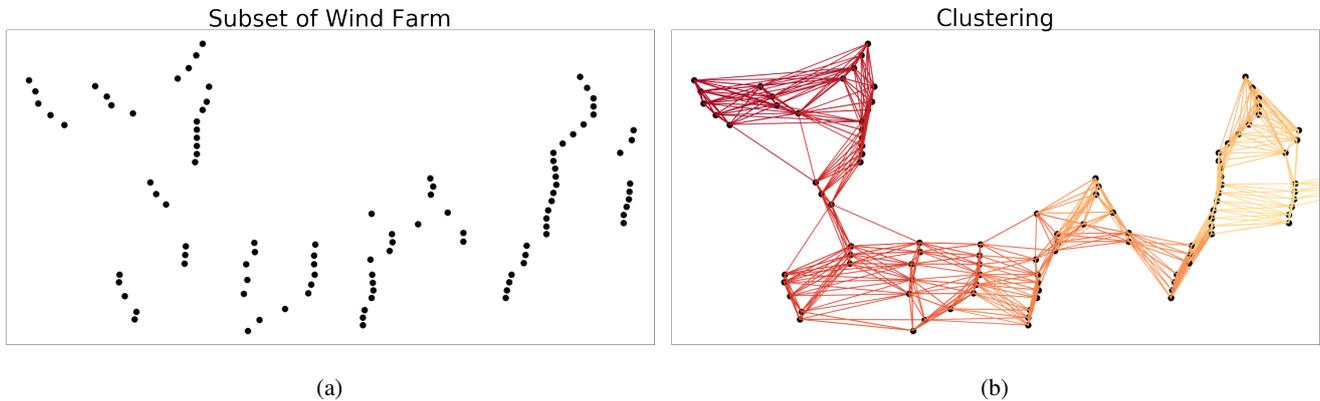
where

$$\theta = \max \left( 1 - \frac{\lambda w_{ij}}{\rho \|x_i + u_{ij} - (x_j + u_{ji})\|_2}, 0.5 \right)$$

Finally, the  $u$ -update is computed as:

$$20 \quad u_{ij}^{m+1} = u_{ij}^m + (x_i^{m+1} - z_{ij}^{m+1}) \quad (15)$$

In addition, the user can specify a stopping criteria that is defined based on the residuals of the primal and dual problems such that  $\|r^m\|_2 \leq \epsilon^{primal}$  and  $\|s^m\|_2 \leq \epsilon^{dual}$ . This setup provides an incentive for the difference between the connected nodes to be zero. For the wind farm example, this means that turbines near each other should have similar wind direction measurements. There are two penalty parameters,  $\lambda$  and  $\rho$ , that can be used to weight an individual turbine's measurement against the measurements of turbines nearby.

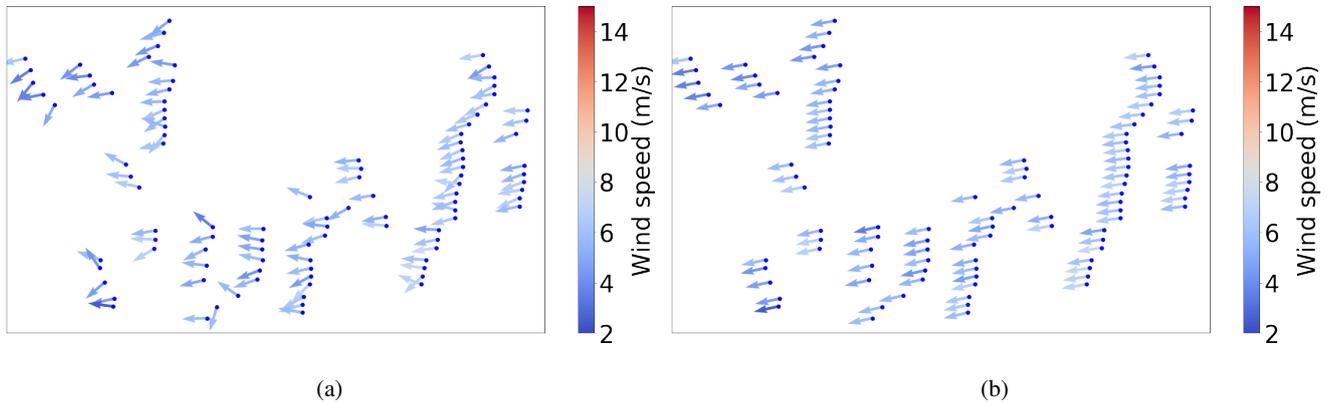


**Figure 2.** (a) This figure shows a subset of the turbine locations used in this study. (b) Connecting undirected edges indicating all groupings within the wind farm based on distance from each wind turbine to its neighbors.

#### 4 Wind Farm Configuration

The wind direction consensus approach was demonstrated on a subset of wind turbines used in the Wind Forecasting Improvement Project, also known as WFIP2 (Pearse et al. (2017)). For this study, we only consider a subset of the wind farm. A subset of those turbines are shown in Figure 2a, resulting in more than 100 turbines. Note, the scale has been removed for proprietary reasons. The wind farm includes two met towers with sensors at 50 m and 80 m and a sodar, shown in Figure 2a. In addition, SCADA data was used at 1-minute time intervals from individual turbines over approximately 8 months. The data channels of interest were the perceived wind direction, wind speed, and measured power at each turbine. The latitude and longitude values of each turbine were transformed into Universal Transverse Mercator (UTM) coordinates to provide approximate distances in meters between turbines. In addition, data was available for the same time period for the met towers and the sodar. The met towers had data at 1-minute time intervals and the sodar had data available at 10-minute intervals.

First, the network topology was determined by connecting each turbine to the nearest 10 turbines as shown in Figure 2b. Next, the penalty parameters  $\lambda$  and  $\rho$  in (6) were tuned using 20 minutes of SCADA and met tower data. In particular, the SCADA data was used to interpolate the wind direction at the met tower locations and  $\lambda$  and  $\rho$  were used to determine the amount of agreement in the wind direction across the wind farm. For example, if the penalty parameters are high, then the turbines will agree on one wind direction across the wind farm. These parameters indicate how much to trust connected turbine measurements with respect to the measurement from the turbine in determining the wind direction at each turbine. After tuning these parameters, 500 hours of SCADA data was evaluated by the consensus algorithm to calculate each turbine's wind direction and the sodar on site was used for validation. The results of this analysis are shown in Section 5.



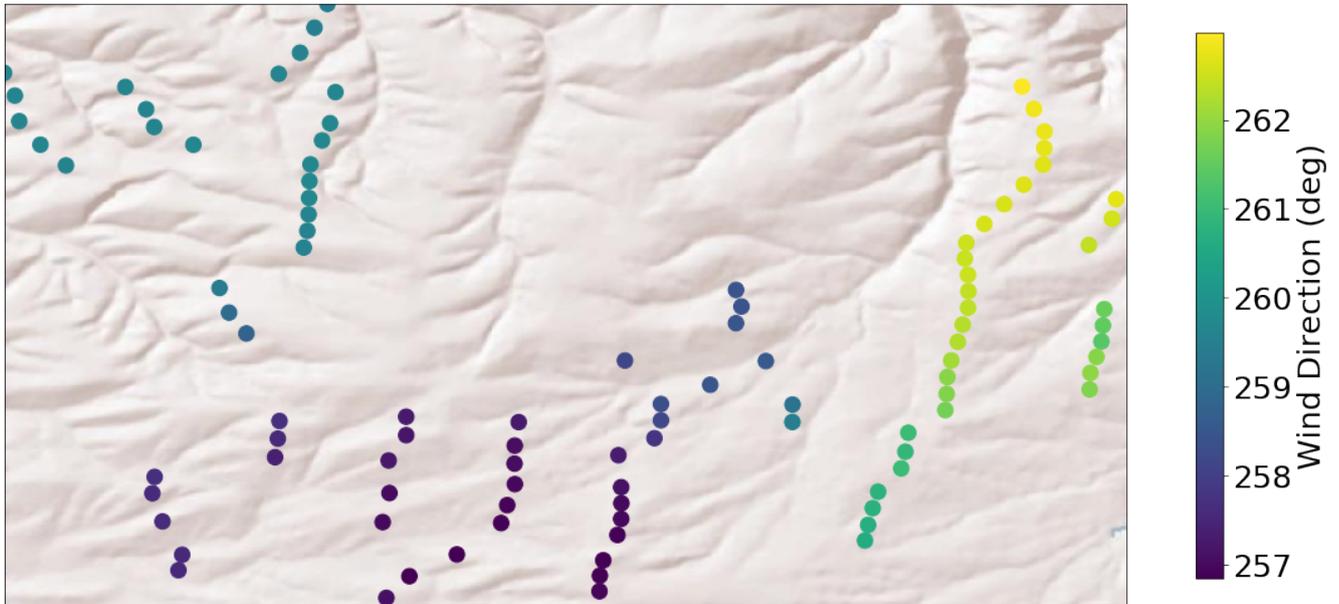
**Figure 3.** (a) The wind direction recorded at each turbine across the wind farm at one time step. (b) The wind direction estimated by the consensus algorithm using the SCADA data recorded by each turbine.

## 5 Results

First, the wind direction was examined at one time step with and without the consensus algorithm. Figure 3a shows the wind direction recorded at each wind turbine for one time step. This shows the variability across the wind farm and the disagreement among turbines. Figure 3b shows the wind direction determined from the consensus algorithm at one time step. Each time step takes 0.5 s to compute, in serial on a desktop computer, using the set-up described in Section 3. The output of the consensus algorithm shows a smoothly varying wind direction across the wind farm and the algorithm allows for the wind direction to vary smoothly across the wind farm. Figure 4 shows the terrain and the corresponding color-coded wind direction and indicates that the wind direction varies with terrain, an effect that this algorithm is able to capture even in complex terrain. In particular, a strong change in wind direction is detected near the canyon in the north-central part of the wind farm.

Next, the results of the consensus algorithm were used to determine the wind direction at the location of the sodar on the outside of the wind farm by interpolation from the individual turbines. The results were compared with the time series data recorded by the sodar as shown in Figure 5. The top plot shows the time-series of the sodar in blue and the estimate based on the consensus algorithm in red. This figure shows good agreement between the estimated wind direction and the wind direction recorded at the sodar. The 100 hours shown in Figure 5 were chosen to demonstrate the performance of the algorithm under large wind direction changes due to meteorological drivers. Figure 5 (bottom) shows the error between the estimated and actual signal recorded by the sodar. The points are color-coded with respect to wind speed. The largest errors are experienced at low wind speeds, typically at or below cut-in.

Next, to demonstrate the benefits of the estimated wind direction at each turbine, the estimated wind direction was used to determine the dynamic yaw error experienced by each turbine across the wind farm. The error was calculated between the estimated wind direction and the measured wind direction at the turbine. Figure 6a shows the power curve, computed with 95% confidence intervals, of one turbine with small yaw errors (less than  $1^\circ$ ) in green and large yaw errors (greater than  $10^\circ$ ) in

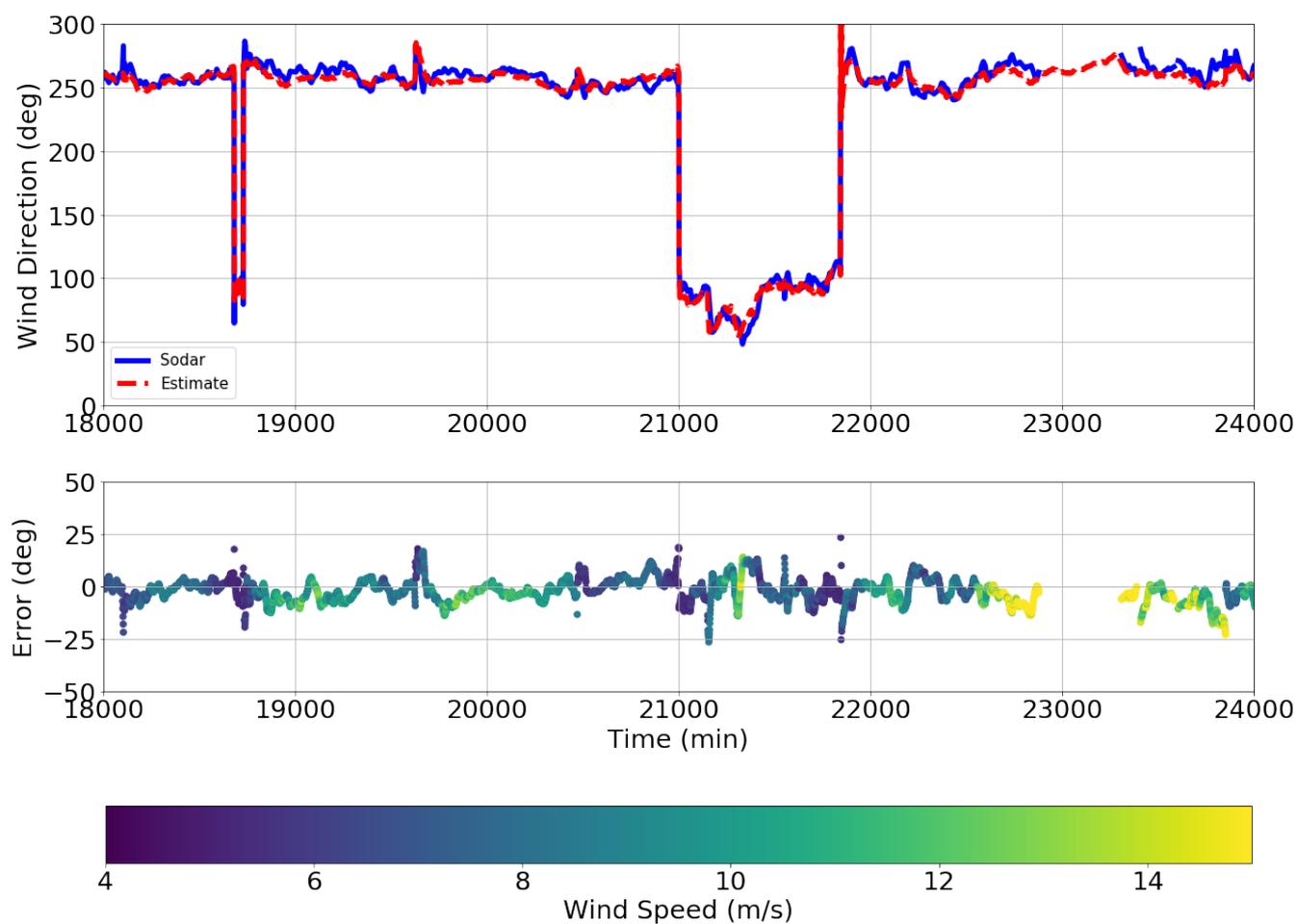


**Figure 4.** The wind direction is shown across part of the wind farm based on the color of each turbine with the terrain plotted in the background. This shows the effects of terrain on the wind direction and indicates that the wind direction can vary across the wind farm.

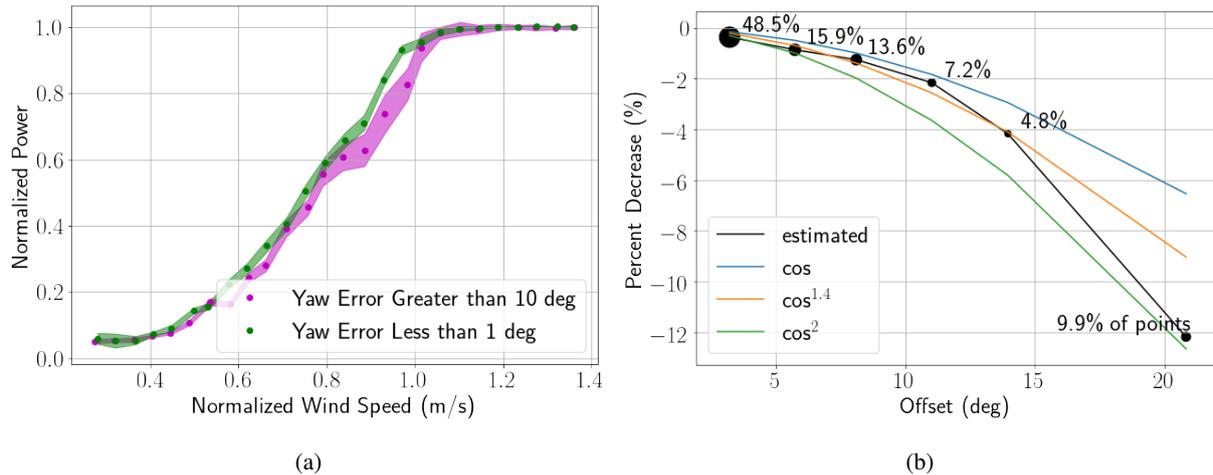
magenta. Using the estimated wind direction, we can identify when turbines are operating in misaligned conditions, possibly caused by noise at the turbine's individual sensor.

The average power loss of a turbine was computed for different amounts of yaw error using a yaw error of less than  $1^\circ$  as the baseline. Figure 6b shows the results of the average data, across all turbines across 500 hours, in black. The percent of the 500 hours of data used to compute the power loss at each offset is shown. This plot indicates that the turbines across the wind farm spend a significant amount of time misaligned. In literature, turbines operating in yaw misalignment have a loss of power that is proportional to  $\cos(\theta)^{p_P}$  (Gebraad et al. (2016)), where  $\theta$  is the misalignment angle and  $p_P$  is determined empirically (Fleming et al. (2017)). The value of  $p_P$  has been shown to be between 1.0 and 2.0 (Gebraad et al. (2016); Fleming et al. (2017)). In Figure 6b, the data most closely follows a  $\cos(\theta)^{1.4}$  relationship. The loss in power is consistent with literature, which again indicates that the consensus-algorithm-estimated wind direction is a reasonable estimate of the wind direction at each turbine. Having a better wind direction measurement for the yaw controller of a turbine could improve yaw misalignments and reduce the amount of yawing a turbine performs and might improve the effectiveness of wake steering (Fleming et al. (2014a)).

Finally, one additional metric was used for assessing the value of the consensus algorithm. We calculated the relative power performance of the turbines as they experienced large ( $>20^\circ$ ) and small ( $<10^\circ$ ) yaw errors compared to two baselines: (1) the sodar in the wind farm, and (2) the estimated wind direction from the consensus algorithm. In this analysis, we first eliminated erroneous data such as power values more than 13% over rated power and less than 0. We used 10-min sodar data and 1-min turbine power and nacelle position data and calculated the statistics including the mean, median, and standard deviation



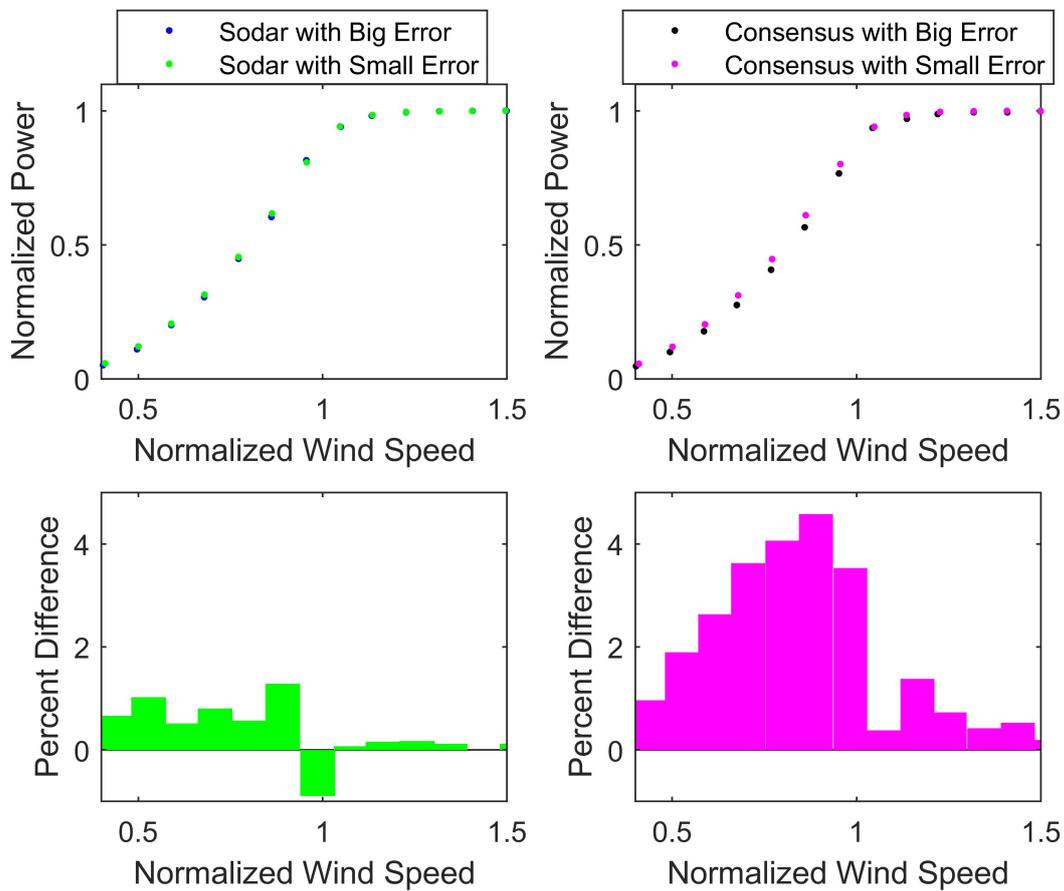
**Figure 5.** (Top) Comparison between the estimated wind direction at the location of the sodar (red) and the actual wind direction recorded by the sodar (blue). (Bottom) Error between the estimate and the actual wind direction recorded at the sodar. The dots are color-coded based on wind speed.



**Figure 6.** (a) The power curve of a single turbine was computed using 0.5 m/s bins over 500 hours of data. The power curve is plotted for small yaw errors (green) and large yaw errors (magenta). (b) The power loss is computed for different yaw offset angles. The trend is consistent with cosine power laws seen in literature.

of the power at each individual turbine based on wind speed bins of 1 m/s. An average power curve was computed for the wind turbines in this wind farm by removing data points that were outside of two standard deviations above or below the median in each 1 m/s wind speed bin. Based on a separate analysis using a power curve analysis for each individual turbine, we determined that some turbines likely experienced drift in their yaw-position sensors, causing it to appear that they had regularly large yaw errors despite being oriented correctly into the wind. We therefore removed data from turbines with consistently high, inexplicable yaw errors compared to the consensus algorithm or the sodar.

Figure 7 shows this power curve analysis that attempts to determine the effects of yaw error on the power of a turbine. The yaw error was computed using a sodar on-site and the wind direction estimated with the proposed consensus algorithm. The top-left plot shows the difference in the power-curve analysis when a large versus small error is detected using the sodar. The bottom-left plot shows the percent difference between these two power curves at each bin, where most percent differences are less than 1%. The top-right plot shows the difference between the two power curves when using the wind direction estimate as a baseline. The bottom-right plot shows the percent difference between the small and large yaw errors detected with the wind direction estimate, which shows much larger differences than when using the sodar as yaw direction “truth.” The consensus algorithm, therefore, appears better able to detect true yaw errors than the sodar. With a modified wind direction signal, it might be possible to account for these errors in real time and improve the performance of an individual turbine by taking advantage of consensus-derived wind direction estimates from additional data available within the wind farm. Many data-analysis factors impact the quantitative performance, so we do not assert any specific quantitative gains for the consensus algorithm compared to turbine wind vanes or sodar, but instead point to some qualitative differences to motivate future research. Given the data-processing decisions explained in the preceding paragraphs, it appears that the consensus algorithm more closely predicts the



**Figure 7.** (top row) Power curves based on binned data for 1 m/s wind speed bins across all turbines, and (bottom row) percent difference between binned power from small yaw errors to large yaw errors. Positive percent difference indicates more power is generated when yaw errors are small. The left column is based on comparisons between turbine wind vanes and the sodar and the right column is based on comparisons between the turbine wind vanes and the consensus algorithm.



actual yaw error across turbines than does the sodar in that small errors measured with consensus result in higher mean powers for wind speed bins below rated. This result is expected because there is only one sodar in the wind farm and a significant amount of spatial variation in the wind direction, which the consensus algorithm is able to capture. This analysis indicates that by using only SCADA data in the way outlined in this article, it is possible to detect dynamic yaw misalignment. A  
5 corrected wind direction input based on this algorithm could be used with the yaw controller, which might be able to minimize yaw misaligned conditions. Lidars have been used to date to correct for yaw misalignment. However, due to the limitations in scanning distances, lidars have only been able to correct static misalignment. This approach allows for more robust wind direction measurements that correspond to large time and space scales, which can ride through local wind variations with small timescales and might avoid yawing prematurely.

## 10 6 Conclusions and Future Work

This article presents a framework for an autonomous wind farm where turbines benefit from increased communication, i.e., sharing of data. The specific example that has been presented here: is a wind direction consensus algorithm that uses information from nearby turbines to determine the wind direction. The results indicate that this strategy is able to detect dynamic yaw errors and was shown using a power curve analysis. By increasing communication between turbines, it is possible to improve  
15 the performance of turbines in a wind farm.

Future work will determine the optimal number of connections between turbines given the layout and terrain features as well as allowable computation time. The network topology chosen could facilitate short-term forecasting in a wind farm. For example, it takes minutes for wind to propagate downstream. Turbines that exist upstream could communicate to connected downstream turbines the near-term conditions including wind direction changes that could mitigate extreme loading events.

20 Finally, future work will include incorporating wind speed data to have a better estimate of the atmospheric conditions at each turbine. In addition, future work can use this framework for fault detection in wind direction sensors. Sensors that stray far from the consensus can be flagged for maintenance and if wind direction sensors are offline, information from nearby turbines can be used to continue operating the turbine until maintenance can be scheduled and completed.

### *Acknowledgements.*

25 This work was authored by the National Renewable Energy Laboratory, operated by the Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy, Wind Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the U.S. Government. The U.S. Government retains, and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclu-  
30 sive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.



## References

- Annoni, J., Bay, C., Taylor, T., Pao, L., Fleming, P., and Johnson, K.: Active Power Control for Wind Farms Using Distributed Model Predictive Control and Nearest Neighbor Communication, in: American Control Conference (ACC), IEEE, 2018.
- Baros, S. and Ilic, M.: Distributed Torque Control of Deloaded Wind DFIGs for Wind Farm Power Output Regulation, 32, 4590–4599, 2017.
- 5 Bay, C., Annoni, J., Taylor, T., Pao, L., and Johnson, K.: Active Power Control for Wind Farms Using Distributed Model Predictive Control and Nearest Neighbor Communication, in: American Control Conference (ACC), IEEE, 2018.
- Boersma, S., Doekemeijer, B., Gebraad, P., Fleming, P., Annoni, J., Scholbrock, A., Frederik, J., and van Wingerden, J.: A tutorial on control-oriented modeling and control of wind farms, in: Proceedings of the American Control Conference (ACC), Seattle, USA, 2017.
- Boyd, S. and Vandenberghe, L.: Convex optimization, Cambridge university press, 2004.
- 10 Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine Learning, 3, 1–122, 2011.
- Ebegbulem, J. and Guay, M.: Distributed Extremum Seeking Control for Wind Farm Power Maximization, in: International Federation of Automatic Control, pp. 147–152, 2017.
- Ferrari, S., Foderaro, G., Zhu, P., and Wettergren, T. A.: Distributed optimal control of multiscale dynamical systems: a tutorial, IEEE Control Systems, 36, 102–116, 2016.
- 15 Fleming, P., Annoni, J., Shah, J. J., Wang, L., Ananthan, S., Zhang, Z., Hutchings, K., Wang, P., Chen, W., and Chen, L.: Field test of wake steering at an offshore wind farm, Wind Energy Science, 2, 229, 2017.
- Fleming, P. A., Gebraad, P. M., Lee, S., van Wingerden, J.-W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P.: Evaluating techniques for redirecting turbine wakes using SOWFA, Renewable Energy, 70, 211–218, 2014a.
- 20 Fleming, P. A., Scholbrock, A. K., Jehu, A., Davoust, S., Osler, E., Wright, A. D., and Clifton, A.: Field-test results using a nacelle-mounted lidar for improving wind turbine power capture by reducing yaw misalignment, Journal of Physics: Conference Series, 524, 012002, <http://stacks.iop.org/1742-6596/524/i=1/a=012002>, 2014b.
- Friedman, J., Hastie, T., and Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso, Biostatistics, 9, 432–441, 2008.
- Gebraad, P., Teeuwisse, F., Wingerden, J., Fleming, P., Ruben, S., Marden, J., and Pao, L.: Wind plant power optimization through yaw control using a parametric model for wake effects—a CFD simulation study, Wind Energy, 19, 95–114, 2016.
- 25 Gionfra, N., Sandou, G., Siguerdidjane, H., Faille, D., and Loevenbruck, P.: A Distributed Consensus Control Under Disturbances for Wind Farm Power Maximization, in: IEEE Conference on Decision and Control, Melbourne, Australia, 2017.
- Hallac, D., Leskovec, J., and Boyd, S.: Network lasso: Clustering and optimization in large graphs, in: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 387–396, ACM, 2015.
- 30 Kroposki, B. D., Dall-Anese, E., Bernstein, A., Zhang, Y., and Hodge, B. S.: Autonomous Energy Grids: Preprint, Tech. rep., National Renewable Energy Laboratory.(NREL), Golden, CO (United States), 2017.
- Lindenberg, S.: 20% Wind Energy By 2030: Increasing Wind Energy’s Contribution to US Electricity Supply, Diane Publishing, 2009.
- Marden, J., Ruben, S., and Pao, L.: Surveying game theoretic approaches for wind farm optimization, in: 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, p. 1154, 2012.
- 35 Movric, K. H. and Lewis, F. L.: Cooperative optimal control for multi-agent systems on directed graph topologies, IEEE Transactions on Automatic Control, 59, 769–774, 2014.



- Pearse, S., Munoz, P. J., and Kosovic, B.: Visualization of NCAR's Wind Forecast Improvement Project 2, in: Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact, p. 77, ACM, 2017.
- Schlipf, D., Schlipf, D. J., and Kühn, M.: Nonlinear model predictive control of wind turbines using LIDAR, *Wind Energy*, 16, 1107–1129, 2013.
- 5 Scholbrock, A., Fleming, P., Fingersh, L., Wright, A., Schlipf, D., Haizmann, F., and Belen, F.: Field testing lidar-based feed-forward controls on the NREL controls advanced research turbine, in: 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, p. 818, 2013.
- Shamma, J.: Cooperative control of distributed multi-agent systems, John Wiley & Sons, 2008.
- Simley, E., Pao, L. Y., Frehlich, R., Jonkman, B., and Kelley, N.: Analysis of light detection and ranging wind speed measurements for wind turbine control, *Wind Energy*, 17, 413–433, 2014.
- 10 Soleimanzadeh, M., Wisniewski, R., and Johnson, K.: A distributed optimization framework for wind farms, *Journal of Wind Engineering and Industrial Aerodynamics*, 123, 88–98, 2013.
- Spudić, V., Conte, C., Baotić, M., and Morari, M.: Cooperative distributed model predictive control for wind farms, *Optimal Control Applications and Methods*, 36, 333–352, 2015.
- 15 Wang, L., Wen, J., Cai, M., and Zhang, Y.: Distributed Optimization Control Schemes Applied On Offshore Wind Farm Active Power Regulation, in: The 8th International Conference on Applied Energy – ICAE2016, pp. 1192–1198, 2017.
- Xu, R. and Wunsch, D.: Survey of clustering algorithms, *IEEE Transactions on neural networks*, 16, 645–678, 2005.
- Zhu, M. and Martínez, S.: Distributed optimization-based control of multi-agent networks in complex environments, Springer, 2015.