

Mobile Sensing for Wind Field Estimation in Wind Farms

Preprint

David J. Pasley,¹ Marco M. Nicotra,¹ Lucy Pao,¹ Jennifer King,² Christopher Bay²

1 University of Colorado Boulder 2 National Renewable Energy Laboratory

Presented at the 2020 American Control Conference (ACC) July 1–3, 2020

NREL is a national laboratory of the U.S. Department of Energy Office of Energy Efficiency & Renewable Energy Operated by the Alliance for Sustainable Energy, LLC Conference Paper NREL/CP-5000-76133 July 2020

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308



Mobile Sensing for Wind Field Estimation in Wind Farms

Preprint

David J. Pasley,¹ Marco M. Nicotra,¹ Lucy Pao,¹ Jennifer King,² Christopher Bay²

1 University of Colorado Boulder 2 National Renewable Energy Laboratory

Suggested Citation

Pasley, David J., Marco M. Nicotra, Lucy Pao, Jennifer King, Christopher Bay. 2020. *Mobile Sensing for Wind Field Estimation in Wind Farms: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5000-76133. <u>https://www.nrel.gov/docs/fy20osti/76133.pdf</u>.

NREL is a national laboratory of the U.S. Department of Energy Office of Energy Efficiency & Renewable Energy Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy

Conference Paper NREL/CP-5000-76133 July 2020

National Renewable Energy Laboratory 15013 Denver West Parkway Golden, CO 80401 303-275-3000 • www.nrel.gov

Contract No. DE-AC36-08GO28308

Laboratory (NREL) at www.nrel.gov/publications.

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) 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 herein do not necessarily represent the views of the DOE 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 nonexclusive, 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.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at <u>www.nrel.gov/publications</u>.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via <u>www.OSTI.gov</u>.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Mobile Sensing for Wind Field Estimation in Wind Farms

David J. Pasley, Marco M. Nicotra, Lucy Pao, Jennifer King, Christopher Bay

Abstract—This paper introduces a novel approach for estimating the wind field over an entire wind farm using a mobile sensor to collect limited amounts of data. The proposed method estimates the boundary conditions of a simplified turbine wake model by computing the model sensitivity matrix and using a recursive least-squares algorithm to recover the model parameters from the wind field measurements. To address the fact that it is not practical to take measurements across the entire wind farm, the proposed method classifies each area on the map based on its sensitivity to parameter variations. This classification is then used to generate a suitable path for a mobile sensor, which is charged with collecting data for the recursive least-squares algorithm. The proposed framework can successfully estimate the model boundary conditions using just the measurements collected along the path of the mobile sensor. This preliminary result paves the way for using real-time wind field estimates for the coordinated control of all the turbines within a wind farm.

I. INTRODUCTION

Wind energy continually gains momentum as a clean and competitive source of power for many communities worldwide [1]. Efforts to reduce the levelized cost of energy from wind farms have led to an active research community that aims to increase energy output and reduce maintenance costs [2], [3]. While approaches for improving power production have focused on controlling single turbines, it has been shown in high-fidelity simulation that coordinated control of multiple wind turbines can yield over a 5% increase in power production [3], [4]. A key obstacle for the implementation of

J. King and C. Bay are researchers at the National Renewable Energy Laboratory. e-mail: jennifer.king@nrel.gov; christopher.bay@nrel.gov

The authors would like to thank Eric Frew, director, Autonomous Systems Interdisciplinary Research Theme (ASIRT) at University of Colorado Boulder (CU Boulder) for supporting this research through the ASIRT Seed Grants; and Cory Dixon, chief technologist, Integrated Remote and In Situ Sensing (IRISS) at CU Boulder for his insights and suggestions. The third author also acknowledges support from a Palmer Endowed chair professorship and a fellowship from the Hanse-Wissenschaftskolleg Institute for Advanced Study (Delmenhorst, Germany).

Copyright Satement: This work was authored [in part] by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) 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 DOE 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 nonexclusive, 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

coordinated control for wind farms is the absence of accurate real-time measurements and estimates of the entire wind field. Indeed, the current practice for wind field estimation relies on a combination of: i) regional weather data provided by radars and/or satellites, and ii) local measurements from lidar sensors and anemometers deployed on the actual wind turbines and/or fixed meteorological towers [3].

One possible way to improve accuracy is to incorporate measured data into a wind field model to extrapolate the behavior in unmeasured locations, and existing research has shown that model-based estimation can provide accurate flow-field estimates using a few local sensors [5]. However, this result is impractical when using fixed sensors because the optimal placement strategy varies based on the wind conditions. This paper addresses the issue by developing an accurate and flexible framework for flow field estimation in wind farms using mobile sensors. Recent research has shown the utility of mobile sensing in air quality monitoring [6], gas tomography [7], the measurement of greenhouse gases [8], and even the mapping of atmospheric phenomena [9].

This paper introduces a method for estimating wind conditions in a wind farm using a mobile sensor. The method employs a recursive least-squares algorithm to estimate the boundary conditions of a reduced-order model using sparse data. The data used in the estimator is selected on the basis of an information density map (IDM), which classifies each location based on its sensitivity to parameter variation. The IDM is initially validated using a threshold strategy that ignores locations with an IDM score below a certain value. Then, the IDM is used to generate a path for the mobile sensor. This work does not focus on optimization of mobile sensor trajectories or sensing locations, but rather simply seeks to demonstrate that mobile sensing is a viable method for real-time estimation of wind conditions in wind farms.

II. METHODS

Existing literature provides numerous computational fluid dynamics (CFD) models which specialize the Navier-Stokes equations to wind turbine applications [10]. One well-known method is the Simulator fOr Wind Farm Applications, which uses CFD to generate high-fidelity simulations [11]. Such models are too computationally expensive for real-time implementation and are more suited for off-line validations. To enable real-time implementation, this paper uses the FLOw Reduction In Steady-State (FLORIS) model, which relies on a simplified turbine wake approximation to provide estimates of the wind field [12]. Recent studies validated the FLORIS model for wind farm control and turbine layout optimization

D.J. Pasley is a Ph.D. Student in the Department of Electrical, Computer, and Energy Engineering (ECEE), University of Colorado Boulder. e-mail: david.pasley@colorado.edu

M.M. Nicotra and L. Pao are professors in the ECEE department of the University of Colorado Boulder. email: marco.nicotra@colorado.edu; pao@colorado.edu

[13]–[16], and its potential as a real-time wind field estimator is being investigated.

For the reader's convenience, this paper is structured as a series of incremental steps validating each underlying concept. The first step demonstrates the ability to estimate the boundary conditions of the wind field model, given access to the complete set of measurements. This is achieved by quantifying the sensitivity of the FLORIS model to parameter variations, and using a recursive least-squares algorithm to update the model parameters, given the current estimation error. The second step quantifies the amount of useful information contained in each gridpoint of the estimated wind field. This is achieved by combining the coefficients in the sensitivity matrix to generate an information density map, and validating the IDM by modifying the recursive leastsquares algorithm to only use measurements with an IDM value above a threshold. The final step formulates a data collection strategy, which is updated based on the current model estimate. This is achieved by using the IDM as a score map for a greedy path planner, and using the resulting data for wind field estimation.

A. Sensitivity Matrix

The FLORIS model, available in [17], is implemented in Python and requires a JavaScript Object Notation (JSON) input to define the wind farm. The JSON input allows users to define the layout of the field, the type of turbines in use, the yaw of the turbines, the incoming wind speed and direction, and other parameters. For this research, two National Renewable Energy Laboratory (NREL) 5-MW reference turbines [18] were placed 800 meters apart along the dominant downwind direction of the wind farm. The yaw angles of the turbines are kept at 0° . The output in use from the FLORIS model is the flow field, or u field, a discretized 3D map of wind speeds throughout the wind farm. For this work, only the 2D data at hub height was used. An example output is shown in Fig. 1. The $n \times m$ flow field generated by the FLORIS model is vectorized to $u \in \mathbb{R}^{\ell}$, with $\ell = nm$, and is modeled as the output of a function

$$u = f(v, \theta), \tag{1}$$

where $v \in \mathbb{R}^+$ is the average speed and $\theta \in (-\pi, \pi]$ is the average direction of the incoming wind. The sensitivity of the estimated *u* field with respect to parameter variation is defined using the Jacobian

$$\nabla f(v,\theta) = \left[\frac{\partial f(v,\theta)}{\partial v}, \frac{\partial f(v,\theta)}{\partial \theta}\right].$$
 (2)

Because the analytic expression for the partial derivatives is not available, the Jacobian is computed numerically as

$$\frac{\partial f(v,\theta)}{\partial v} \approx \frac{f(v,\theta) - f(v + \epsilon_v,\theta)}{\epsilon_v} \tag{3}$$

$$\frac{\partial f(v,\theta)}{\partial \theta} \approx \frac{f(v,\theta) - f(v,\theta + \epsilon_{\theta})}{\epsilon_{\theta}}, \tag{4}$$

where ϵ_v and ϵ_{θ} represent small perturbations in the estimates. Here, the Jacobian is used rather than other methods,



such as the Bayesian approach, to reduce computational complexity to meet the real-time requirements of the system.

B. Recursive Least Squares

Given estimates \hat{v} and $\hat{\theta}$, consider the linear approximation

$$u = \hat{u} + \nabla f(\hat{v}, \hat{\theta}) \begin{bmatrix} v - \hat{v} \\ \theta - \hat{\theta} \end{bmatrix}.$$
 (5)

By measuring u and computing $\hat{u} = f(\hat{v}, \hat{\theta})$, it is possible to compute a local least-squares estimate of v and θ using the Moore-Penrose pseudoinverse, $\nabla f(v, \theta)^{\dagger}$. This operation can be repeated iteratively to account for the model nonlinearities, leading to the recursive least-squares algorithm

$$\begin{bmatrix} \hat{v}_{k+1} \\ \hat{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{v}_k \\ \hat{\theta}_k \end{bmatrix} + \nabla f(\hat{v}_k, \hat{\theta}_k)^{\dagger} [u - \hat{u}_k]$$
(6)

where \hat{v}_k and $\hat{\theta}_k$ are the current wind speed and direction estimates and \hat{u}_k is the current u field based on those estimates. Convergence properties for the recursive leastsquares method (given in [19]) require the assumption that the function f(x) is Lipschitz continuous, locally bounded, and strongly convex. As shown later in Section III-A, the proposed method rapidly converges to the correct values, v and θ as $k \to \infty$, and thus these assumptions can be made. However, equation (6) assumes that the entire u field is available for measurement. The following subsections will illustrate how to drop this assumption.

C. Information Density Map (IDM)

The objective of the IDM is to quantify the amount of "useful information" contained in each node of the estimated flow field and provide a metric for prioritizing data collection points. In the presence of a single parameter, the most straightforward choice is to prioritize the nodes with the highest sensitivity. Given the parameters v and θ , the individual sensitivities must be combined into a single value; The IDM is generated by normalizing each column of the Jacobian (2) to its highest value and performing a point-wise multiplication. Figures 2 and 3 portray normalized sensitivities with respect to the average wind speed, v, and direction, θ . The resulting IDM is illustrated in Fig. 4.

This approach assigns a value in the range [0,1] to each location, where higher values imply an overall higher sensitivity to parametric variations. The IDM serves as the basis for performing parametric estimation with sparse data.



D. Mask Matrix

To account for partial knowledge of the u field, we define M_k as a diagonal $\ell \times \ell$ matrix, where each value on the diagonal is either 1, if the corresponding measurement is to be used, or 0. By modifying equation (6) into

$$\begin{bmatrix} \hat{v}_{k+1} \\ \hat{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{v}_k \\ \hat{\theta}_k \end{bmatrix} + \nabla f_{Mk} (\hat{v}_k, \hat{\theta}_k)^{\dagger} M_k [u - \hat{u}_k], \quad (7)$$

with $f_{Mk}(v,\theta) = M_k u$, it is possible to study the behavior of the method, given partial knowledge of the vector u.

Unlike static sensor placement strategies [5], [20], [21], the mask matrix, M_k , can be updated at each iteration, depending on the current estimates. Fig. 5 illustrates a masking strategy where M_k is assigned by only taking into account the nodes with an IDM score above a threshold, γ . As discussed in Section III-B, this approach can be used to validate the proposed IDM. The following subsection will illustrate how the IDM can be used to generate a mobile sensing trajectory.

E. Path Planner

To take into account that a single mobile sensor cannot take measurements across a wind farm instantaneously, a path planning algorithm is used to generate a suitable trajectory for the mobile sensor. The path traveled by the sensor is then used to generate M_k . Since the purpose of this work is to validate the proposed principle rather than obtain an optimal path, we propose a greedy path planner.

The path planner assumes movement can only occur in cardinal and ordinal directions, and a virtual boundary is created around each turbine. The planner decides each step based on immediate rewards and penalties.

Rewards

- *I_i*, the IDM score of the node being transitioned into (miopic update);
- dI_i/dT , the first derivative of the IDM score with respect to the transition (gradient-based update);
- d^2I_i/dT^2 , the second derivative of the IDM score with respect to the transition (Newton-based update);



Fig. 5: IDM threshold mask M_k at v=13 m/s, $\theta=10^\circ,\,\gamma=0.5$

• D_i , normalized distance with respect to the node with the highest IDM score with $D_i = 1 - \frac{dist_i}{distMax_i}$, where dist is the Euclidean distance to the unvisited and unplanned node with the highest IDM score, and distMax is the maximum distance possible from that node to any other point on the map (attraction-based update).

Penalties

- β, the direction of transition with respect to the estimated wind direction (promotes flying downwind);
- φ, the direction of transition with respect to the current heading (promotes straight paths);
- λ, a depreciation applied to nodes which have been visited within a defined number of steps (promotes greater coverage).

Each reward and penalty has a separate weight, K, applied to it, except for λ , which is itself applied as a weight raised to the power of N, where N is the number of times a node has been visited within Γ steps. The eight possible transitions are thus scored as in (8). The greedy path planner chooses transitions with the maximum S at each step. An example of a path generated by the planner is shown in Fig. 6.

At the start of the algorithm, a planning horizon P_0 is used, and at least $R_0 \leq P_0$ moves are made by the mobile sensor. The first iteration of the recursive least-squares algorithm is then performed to obtain the first iteration, \hat{v}_1 , $\hat{\theta}_1$, \hat{u}_1 , and the corresponding IDM. After that, the path planner generates a plan horizon of P steps, and the mobile sensor moves $R \leq P$ steps before recalculation. Up to Γ steps of the mobile sensor's actual path are retained in memory, and the nodes in memory are used to define the mask matrix, M_k .

F. The Complete Algorithm

The algorithm is as shown in Algorithm 1.

III. RESULTS

The algorithm converges, even given a large initial error and sparse knowledge of the "actual" u field. The results



Fig. 6: Path planning algorithm executing over the score map at v = 13 m/s and $\theta = 10^{\circ}$. The green path is planned, and the red path has been taken by the mobile sensor.

Algorithm 1

1: $\Gamma, P_0, R_0, R, P \leftarrow$ initialize planning parameters 2: $\hat{v}_k, \hat{\theta}_k \leftarrow \text{initialize estimates } \hat{v}_0, \hat{\theta}_0;$ for desired number of iterations do 3: $\hat{u}_k \leftarrow f(\hat{v}_k, \theta_k);$ 4: Calculate IDM using (3) and (4); 5: if 1^{st} iteration then 6: $PLAN \leftarrow P_0, MOVES \leftarrow R_0$ 7: 8: else $PLAN \leftarrow P, MOVES \leftarrow R$ 9: for $j = 1 \rightarrow PLAN$ do 10: for i in set of possible transitions, T do 11: calculate S_i for T_i using (8) 12: choose T_i with max (S_i) and add to planned path 13: for MOVES do 14: Follow planned path 15: Calculate M_k using path memory as mask 16: Calculate \hat{v}_{k+1} , $\hat{\theta}_{k+1}$ using (7) 17:

are presented incrementally in the following subsections. First, we investigate the behavior of the algorithm with full knowledge of u. Then, we introduce a strategy that generates M_k by only using the nodes with an IDM score above a given value, $\gamma \in [0, 1]$. Finally, we study the behavior of the proposed estimator when M_k is generated using the path planner described in Section II-E. These results were obtained using the set values given in Table I.

A. Full knowledge

With full knowledge of u, the wind speed and direction estimation errors (e_v and e_{θ} , respectively) converge to below E_v and E_{θ} after five iterations of the algorithm. A plot of the estimation error with no mask is given in Fig. 7.

TABLE I: Wind speed and direction values used in simulation

(8)

Description	Symbol	Value
Initial speed estimate	\hat{v}_0	8.0 m/s
Actual speed	v	13.0 m/s
Target speed accuracy	E_v	0.1 m/s
Initial direction estimate	$\hat{ heta}_0$	0.0°
Actual direction	θ	10.0°
Target direction accuracy	E_{θ}	0.1°

B. Thresholding Method

The algorithm was tested with various values of the threshold, γ . As expected, the rate of convergence decreases as γ increases due to the fact that fewer nodes are available to compute the estimate, and the algorithm fails as $\gamma \rightarrow 1$. Plots of the estimate errors are shown in Fig. 7. The map coverage for each iteration is recorded as a percentages of the total number of nodes on the map. The average, minimum, and maximum percentage of nodes covered by the mask are shown in Table II along with the associated time to convergence.

C. Greedy Path Planner

To test the algorithm with the greedy path planner, we use the memory of the sensor, Γ , as a way of measuring the maximum coverage possible. To this end, Γ is expressed in terms of a percentage of the total number of nodes on the map. Note that setting Γ to 100% does not guarantee 100% coverage of the map because the mobile sensor may revisit nodes along its path. Planning parameters are shown in Table III, and transition score weights are presented in Table IV. The convergence of e_v and e_{θ} as Γ is varied is shown in Fig. 8 and the number of iterations to convergence below E_v and E_{θ} are plotted against Γ in Table V.

Various values of P and R were tested with $\Gamma = 100\%$ before settling on the values in Table III. At values of Rless than around 0.15Γ , the rate of convergence decreases, though the processing time of the simulation remains fairly



Fig. 7: Estimation error plots using unmasked ($\gamma = 0$) and IDM thresholding ($\gamma \in [0, 1]$) in the recursive least-squares algorithm.



Fig. 8: Estimation error plots using a mobile sensor path as a mask for u.

constant. As R increases beyond 0.15Γ , the number of iterations remains nearly constant, thereby only increasing the processing time of the simulation. P = 2R is chosen in order to mimic a real-world planning horizon, but P makes no difference in the algorithm's performance.

Keeping relatively high weights for K_I and K_D appears to make the best gains in performance. A high value of K_D ensures the mobile sensor will travel to the unvisited area with the highest information density, and a high K_I results in heavier coverage in areas with high IDM scores. Increases in K_{dI} and K_{d^2I} increase the probability that the sensor will



¹Values of $\gamma \ge 0.7$ do not lead to convergence upon correct values

TABLE III: Planning Parameters

P_0	R_0	R	P
Г	0.9Γ	0.15Γ	2R

TABLE IV: Transition Score Weights

K_I	K_{dI}	$K_{d^2}I$	K_D	K_{β}	K_{ϕ}	λ
10	2	1	200	1	4	0.2

revisit a node to follow a trend of increasing IDM scores. Lower weights often allow the mobile sensor to travel within a few nodes of a high-value node without approaching it.

Lowering λ increases planner coverage, but changing λ does not significantly affect convergence times. The chosen value $\lambda = 0.2$ demonstrated the highest amount of coverage.

Increasing K_{β} causes the mobile sensor to loiter downwind. The weight chosen was the highest integer value that does not cause this behavior. This allows penalizing for energy usage while allowing maneuvers not aligned in the downwind direction. Increasing K_{ϕ} creates long, straight trajectories. Increasing it much beyond the weight chosen results in what appears to be less optimal coverage. Decreasing it further results in highly jagged paths.

Both e_v and e_θ converge to within acceptable margins for $\Gamma \geq 2\%$. Similar to the masked algorithm in Section III-B the performance varies slightly by coverage, but simulation data show that the mask generated by the mobile sensing path converges with a lower amount of actual coverage, and also converges faster. For instance, the average coverage of the masked algorithm at $\gamma = 0.3$ is approximately 17% and it converges in six steps. The path planning algorithm



TABLE V: Percent Coverage and Convergence vs. Γ

converges in five steps at $\Gamma = 20\%$, with slightly over 16% average coverage.

IV. CONCLUSION AND FUTURE WORK

The results of this proof-of-concept study indicate that mobile sensing is a viable solution for wind estimation in wind farms, and are being verified with the high-fidelity Simulator fOr Wind Farm Applications [11].

The initial estimate errors used for this study are larger than what would likely be encountered in practice, but this study also assumes steady-state wind conditions. Future research will evaluate the ability of the algorithm to track changing wind conditions. Additionally, this study includes only a single mobile sensor that utilizes a greedy path planner based on the IDM described in Section II-C. Ongoing research is investigating the use of multiple mobile sensors, as well as optimization of mobile sensing trajectories and the inclusion of vehicle dynamics. Existing methods for optimization of sensing locations will also be explored.

The two-turbine simulations of this estimation algorithm utilizing FLORIS can be run for more than 100 iterations in under a minute on a typical personal computer running Python. Simulations on larger wind farm layouts execute similarly as fast. Though the algorithm may not be suitable for tracking high-frequency changes in local wind conditions such as gusts, when used in conjunction with current sensing methods, it could help to better predict when those gusts will reach downwind turbines, and could also be useful for coordinated control techniques such as wake steering, which are more dependent on lower frequency changes in the wind field. More research is needed to fully understand how mobile sensors could be used to reduce the levelized cost of energy for wind farms.

REFERENCES

- Global Wind Energy Council. Global Wind Statistics 2017, Feb. 2018, https://gwec.net/wp-content/uploads/vip/GWEC_PRstats2017_EN-003_FINAL.pdf
- [2] K. Dykes, M. Hand, T. Stehly, P. Veers, M. Robinson, E. Lantz. Enabling the SMART Wind Power Plant of the Future Through Science-Based Innovation, Aug. 2017, NREL/TP-5000-68123https://www.nrel.gov/docs/fy17osti/68123.pdf
- [3] D.J. Willis, C. Niezrecki, D. Kuchma, E. Hines, S.R. Arwade, R.J. Barthelmie, M. DiPaola, P.J. Drane, C.J. Hansen, M. Inalpolat, J.H. Mack, A.T. Myers, M. Rotea. *Wind Energy Research: State-of-the-Art and Future Research Directions*, Renewable Energy, Volume 125, Sep. 2018, pp. 133-154, ISSN 0960-1481
- [4] S. Boersma, B.M. Doekemeijer, P.M.O. Gebraad, P.A. Fleming, J. Annoni, A.K. Scholbrock, J.A. Frederik, J.W. van Wingerden. A Tutorial on Control-Oriented Modeling and control of Wind Farms, Proc. American Control Conf., May 2017.
- [5] J. Annoni, T. Taylor, C. Bay, K. Johnson, L. Pao, P. Fleming, K. Dykes. *Sparse-Sensor Placement for Wind Farm Control*, 2018, Journal of Physics: Conf. Ser. 1037 032019.
- [6] F. Yang, J. Zhang, Y. Xing, J. He. Deployment of Mobile Air Sensing Network for Urban Air Pollution Monitoring in Hong Kong, International Symposium on Sensor Science, Dec. 2017.
- [7] M.A. Arain, H. Fan, V.H. Bennetts, E. Schaffernicht, A.J. Lilienthal. Improving Gas Tomography with Mobile Robots: An Evaluation of Sensing Geometries in Complex Environments, ISOCS/IEEE International Symposium on Olfaction and Electronic Nose, Montreal, QC, 2017
- [8] L. Tao, K. Sun, D.J. Miller, D. Pan, L.M. Golston, M.A. Zondlo. Low-Power, Open-path Mobile Sensing Platform for High-Resolution Measurements of Greenhouse Gases and Air Pollutants, Appl. Phys. B, Volume 119, Apr. 2015, Pages 153-164
- [9] S. Ravela, I. Sleder, J. Salas. Mapping Coherent Atmospheric Structures with Small Unmanned Aircraft Systems, 2013, AIAA Infotech@Aerospace Conference, Guidance, Navigation, and Control and Co-located Conferences, (AIAA 2013-4667)
- [10] S.K. Kanev, F.J. Savenije, M. Soleimanzadeh, E. Wiggelinkhuizen. Wind Farm Modeling and Control: An Inventory, ECN Wind Energy, Jan. 2014, ECN-E–13-058
- [11] P.A. Fleming, P. Gebraad, J.W. van Wingerden, S. Lee, M. Churchfield, A. Scholbrock, J. Michalakes, K. Johnson, P. Moriarty. *The SOWFA Super-Controller: a High-Fidelity Tool for Evaluating Wind Plant Control Approaches*, Proceedings of the EWEA Annual Meeting, Vienna, Austria, 2013.
- [12] P.M.O. Gebraad and J.W. van Wingerden. A Control-Oriented Dynamic Model for Wakes in Wind Plants, 2014, Journal of Physics: Conf. Ser. 524 012186.
- [13] P.M.O. Gebraad, M.J. Churchfield, P.A. Fleming. Incorporating Atmospheric Stability Effects into the FLORIS Engineering Model of Wakes in Wind Farms, 2016, Journal of Physics: Conf. Ser. 753 052004.
- [14] J.J. Thomas, P.M.O. Gebraad, A. Ning. Improving the FLORIS Wind Plant Model for Compatibility with Gradient-Based Optimization, Wind Engineering, Vol 41, Issue 5, Oct. 2017.
- [15] P.A. Fleming, J. Annoni, M.J. Churchfield, L.A. Martinez-Tossas, K. Gruchalla, M. Lawson, P. Moriarty. A Simulation Study Demonstrating the Importance of Large-Scale Trailing Vortices in Wake Steering. Wind Energy Science, 2018.
- [16] J.J. Thomas, J. Annoni, P.A. Fleming, A. Ning. Comparison of Wind Farm Layout Optimization Results Using a Simple Wake Model and Gradient-Based Optimization to Large Eddy Simulations, AIAA Scitech 2019 Forum, (AIAA 2019-0538)
- [17] NREL, FLORIS Wake Modeling Utility, https://github.com/WISDEM/FLORIS, Accessed Sep. 2018
- [18] J. Jonkman, S. Butterfield, W. Musial, G. Scott. Definition of a 5-MW Reference Wind Turbine for Offshore System Development, Technical Report NREL/TP-500-38060, Feb. 2009.
- [19] A. Beck. Introduction to Nonlinear Optimization. Theory, Algorithms, and Applications with MATLAB, MOS-SIAM Series on Optimization, Ch. 5, pp. 83. Oct. 2014, ISBN-13: 978-1611973648
- [20] S. Powell, M. Howland, J. Crane. Optimization of Sparse Sensor Network in Wind Farms using Reinforcement Learning, Power, 75(100), p.125
- [21] S. Sun, S. Liu, J. Liu, H. Inaki Schlaberg. Wind Field Reconstruction Using Inverse Process With Optimal Sensor Placement, IEEE Transactions on Sustainable Energy 10, no. 3 (2018): 1290-1299