

Date July 31, 2018
Our reference WES-2018-33
Contact person B M Doekemeijer
Telephone +31 15 278 5623
E-mail b.m.doekemeijer@tudelft.nl
Subject Response to reviewers

Delft University of Technology

Delft Center for Systems and Control

Address
Mekelweg 2 (3ME building)
2628 CD Delft
The Netherlands

www.dcsc.tudelft.nl

Reviewers
Wind Energy Science Discussions

Dear Reviewers,

The authors express their gratitude to the reviewers for the time and effort spent to provide positive and constructive feedback to the submitted WES manuscript. Their comments have played a crucial role in further improving the scientific quality and relevance of this work. In accordance to the provided feedback, the article has been revised appropriately. The objective of the attached document is to respond to all the concerns raised by each reviewer, and explain how the authors have addressed each issue in the revised article.

Yours sincerely,

Bart Doekemeijer

Enclosure(s): Response to comments of Reviewer 1
Response to comments of Reviewer 2
Marked-up manuscript highlighting the changes made

Response to comments of Reviewer 1

General comments

- Good and relevant work.
We thank and appreciate the positive feedback of the reviewer.

Specific comments

- **Feedback:** I have concerns about the validity of neglecting the vertical dimension, given significant effects such as vertical meandering, wind shear and veer. There should be more comment about this.

Response: The reviewer raises an important issue: the validity of the control-oriented model. This has also been a concern of the authors in previous work, and has been tackled in the original article of Boersma et al. [2017]. That paper presents and discusses the control-oriented model “WFSim” in greater detail, including the complete mathematical derivation and comparisons against two different high-fidelity large-eddy simulation codes. In the current WES submission, the aim is to provide a concise yet informative overview of the mathematical model used in the remainder of the article.

Furthermore, from an estimation point of view, indeed the model should be as accurate as possible. However, from a controls perspective, the computational cost limits us to models that are fast enough for real-time application, and therefore the accuracy is limited. The presented estimation algorithm has the potential to account for modeling errors (e.g., vertical meandering, wind shear, and wind veer, as the reviewer rightfully mentioned) by assimilating real-time measurements into the simplified mathematical model. In addition to the simulation results presented in the work at hand, the simulation results presented in the work of Boersma et al. [2017] show that the model matches well in terms of the hub-height flow and the turbine power production compared to a high-fidelity large-eddy simulation code, which *does* include the aforementioned atmospheric effects. While these simulations are not conclusive, it at least provides a safe ground to state that the control-oriented model is valid for the demonstrated cases.

Revised changes: The authors have clearly defined the scope of this paper, and more explicitly the scope of Section 2: “The scope of this section is ... assumptions made.” Furthermore, the assumptions in the mathematical model are stated more explicitly, e.g., “ Other vertical flow ... are neglected.”. And the original article is referenced: “The reader is referred to Boersma et al. (2017b) for more information.”

- **Feedback:** In the introduction, some re-wording is necessary: "all commercial turbines ... power electronics" is not right. They are certainly connected, it's only that the rotational speed is decoupled from the grid frequency. Also, in the next line, "grid-disconnected renewable energy plants": maybe "non-synchronous" or some other expression, but not "grid-disconnected"!

Response: The reviewer is correct, and the proposed changes will be made to the introductory text.

Revised changes: The text has been revised in accordance to the suggestions.

- Page 1. *"While there are ... the rotational speed of almost all commercial turbines is currently decoupled from the electricity grid frequency via each turbine's power electronics (Aho et al., 2012)."*
- Page 1. *"As the current grid-connected ... replaced by non-synchronous renewable energy plants, the inertia of the electricity grid will decrease."*

Technical corrections

- **Feedback:** Equation (5): Actually there is now plenty of evidence that the cosine-cubed law for the effect of yaw on power coefficient is not at all correct. There should at least be some comment about this.

Response: The authors thank the reviewer for raising this concern. This issue is in line with the comment concerning model validity. The reviewer is correct that other control-oriented models such as FLORIS [Gebraad et al., 2016] use a different correction term to incorporate a yaw misalignment in the turbine power expression. Unfortunately, to the best of the authors' knowledge, no systematic way is described in the literature on how to best choose this exponential term, besides through a comparison with high-fidelity data, as also discussed in the original paper by Boersma et al. [2017]. In the presented "WFSim" model, the cosine-cubed law originates from a physical derivation of the Navier-Stokes equations where the turbines are modeled using actuator disk theory. It might indeed turn out that, for a certain topology and turbine type, the cubed-law will not suffice, and a different exponent should be assumed. The effect of this discrepancy does not further impinge on the results presented in this article, as all simulations are with non-yawed turbines.

Revised changes: In accordance with the comment of the author, the assumptions and limitations underlying the cosine-cubed law in Equation (5) are clearly mentioned, including an explicit reference to the original paper on WFSim by Boersma et al. [2017]. The original paper gives a more thorough motivation for this power law.

Response to comments of Reviewer 2

General comments

- **Feedback:** The paper is interesting, but way too long compared to its actual contribution. I suggest it is reconsidered for publication after major review.

Response: We thank and appreciate the constructive feedback of the reviewer. The authors understand the concern of the reviewer, and have also had internal discussions before submission on whether or not to include certain sections. The purpose of this paper is to present a complete and unifying framework for joint state-parameter estimation for a control-oriented wind farm model, and therefore it was decided to include certain levels of detail. A more explicit motivation is found in the following responses.

Major comments

1. **Feedback:** My main comment is: reduce the length substantially to increase the readability, but most importantly, to clarify the actual contribution of the paper. Remove details that are previously published. More specifically,

- in the introduction, focus on the actual innovation of the work and its relation to other work.

Response: Currently, the complete closed-loop wind farm control framework is presented in Fig. 1, including the place of this work's contribution in relation to the other components. Furthermore, this (closed-loop) framework is compared against other (open-loop & model-free) frameworks on page 2. For each component of the presented framework, the state of the art is presented and compared to the work of the authors. The actual innovation of this work is described on page 3, where the current contributions compared to previous work are detailed.

- The detailed explanation of the surrogate model (Section 2) can be removed, providing the proper reference to the original work. Of course, a shorter summary of the model will be welcome.

Response: The authors understand the concern of the reviewer. The authors had considered the removal of sections 2 and 3 before submission, but it was decided not to, because the purpose of this article is to present a unifying framework, including enough detail for unfamiliar readers (without the necessary aerodynamics and/or state estimation background) to understand the work to a sufficient degree. The model is presented in such detail to motivate the choice and understanding of the states and parameters that are to be estimated, and to help interpret the results found in section 4. For example, the mixing length parameter ℓ_s is estimated in certain simulations in section 4, and therefore the turbulence model is presented beforehand in section 2. Furthermore, the mathematical notation of section 2 is in agreement with that of sections 3 and 4, making it easier for unfamiliar readers to follow.

Revised changes: The authors have revised the entire document and reduced the paper's length by 10 %.

- Details about the different forms of (standard) Kalman filters are also not necessary, a reference is sufficient, like linear and extended KF.

Response: We appreciate the reviewer's constructive feedback. This concern is in line with the previous comment concerning section 2. It is apparent that the reviewer is an expert in the field of control and state estimation. However, the target audience for this article is not exclusively control engineers, but to the wider range of the wind energy community. The purpose of this article is to present a unifying framework. The various Kalman filters are mathematically introduced to ensure reproducibility and to promote clarity and understanding when interpreting the results presented in section 4. The mathematical model formulation, the Kalman filtering section, and the results all maintain a homogeneous symbol notation and follow one another logically, providing clarity in the remainder of this article.

Revised changes: The authors have revised the entire document and reduced the paper's length by 10 %.

2. **Feedback:** Section 4 but wel written and clear, ...

Response: We appreciate the positive feedback on section 4. The authors also believe that this clarity is largely due to the unifying framework presented, which led to an article that is longer than average.

...but I recommend the following changes:

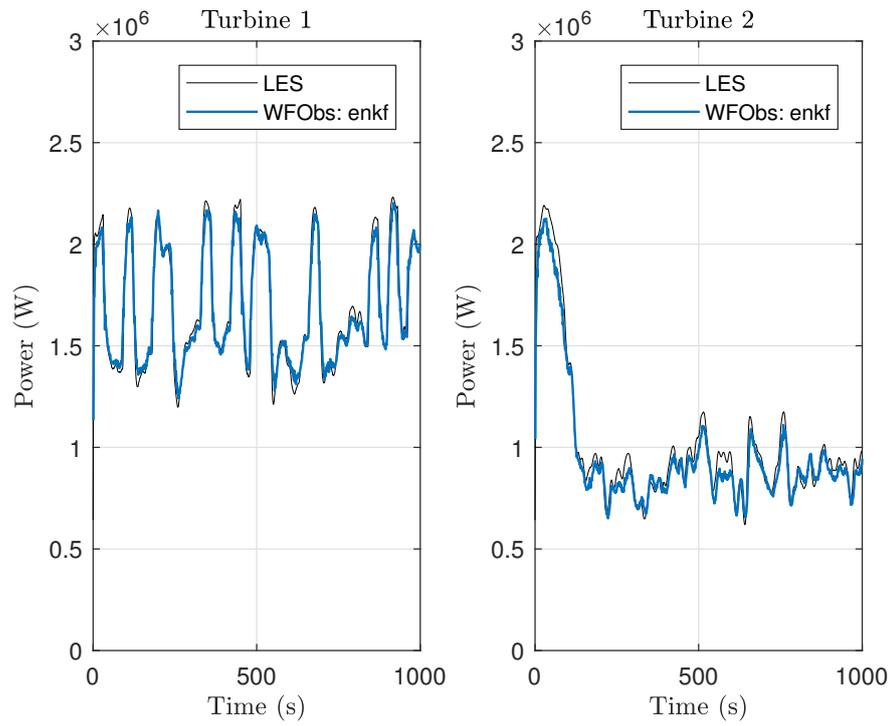
- **Feedback:** Sec 4.2.1 The improvement with respect to the estimation-free case does not seem to be very pronounced. Can you comment more on that. The state estimation adds significant computational complexity, which requires more significant benefits in terms of accuracy. Maybe a different simulation scenario, including changes in the wind conditions during the simulation, can be used to better demonstrate the performance.

Response: The reviewer addresses a very important topic: the usefulness of the estimator compared to the addition in computational cost. In Figure 5, an open-loop simulation (“WFSim”) is compared to closed-loop simulations in which the states of WFSim are estimated using a variety of Kalman filters. For the 2-turbine simulation, the WFSim model is accurate for the first wake, and there is a negligible increase in accuracy. For the second wake, the improvements are more noticeable, and the presented simulation case suffices for the comparison of the different Kalman filter algorithms. The reviewer is correct to state that the case does not necessarily motivate the use of state estimation in the 2-turbine case. However, the 2-turbine simulation study showcases the differences in Kalman filters and sensor locations, building up towards the more realistic 9-turbine wind farm simulation. In the 9-turbine simulation study, the need for state estimation becomes more apparent.

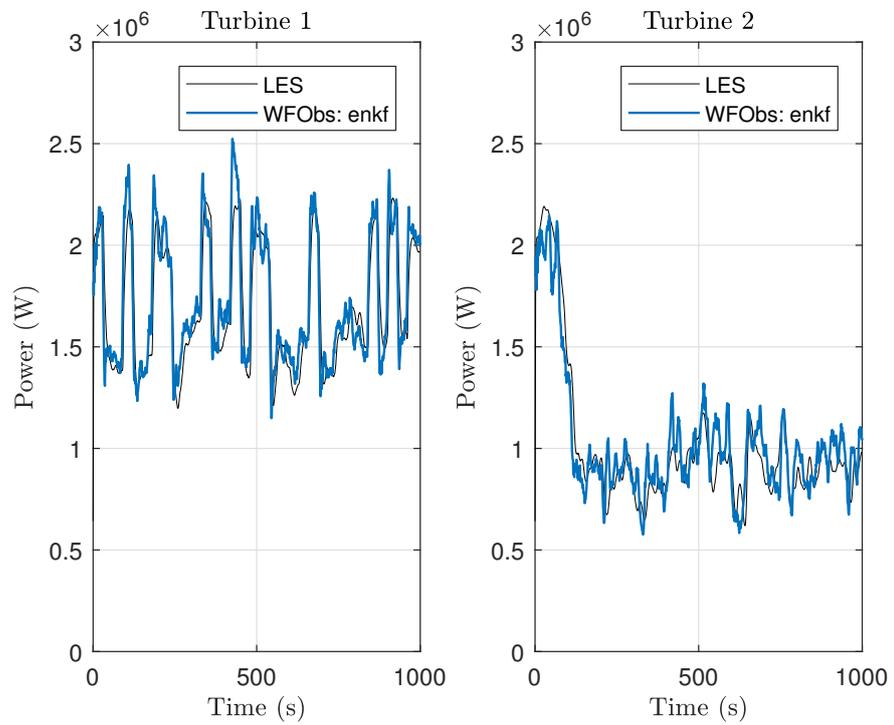
Revised changes: The purpose of the 2-turbine case study (“a two-turbine wind farm is simulated to analyze the effect of different measurement sources, KF algorithms, and the difference between state-only and state-parameter estimation”) and the 9-turbine case study (“the purpose of this case study is to highlight the need for state-parameter estimation for accurate wind farm modeling”) have been highlighted.

- **Feedback:** Sec 4.2.2, Fig 6: please show the model output (WT power) estimation and compare to the true output. At both turbines the wind speed is estimated clearly higher than the simulated one, implying that the estimated power will also be (much!) higher.

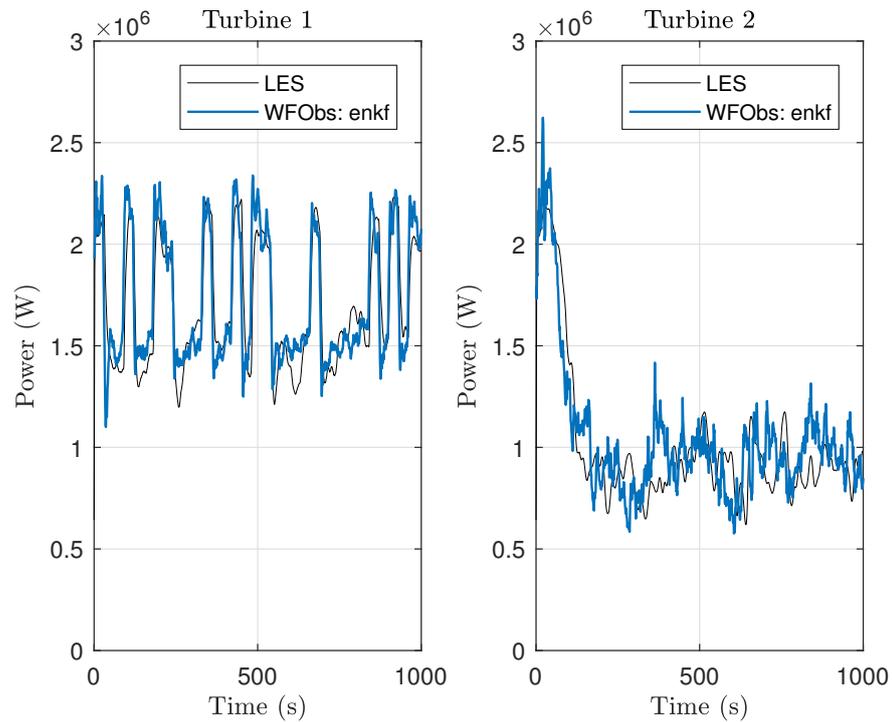
Response: The authors appreciate the reviewer’s idea of showing the turbine power signals, be it that it will elongate the article even further. However, the authors are confused about the statement that the estimated turbine power is higher than the simulated power. Figure 6 shows the absolute value of the *error* in the flow velocities. For clarification, the power signals are plotted next of 1) the simulation where the EnKF uses power measurements, 2) the simulation in which upstream flow measurements are used, and 3) the simulation in which downstream flow measurements are used:



Power measurements



Upwind lidar measurements



Downwind lidar measurements

From these figures, the difference between true and estimated power is small. Note that the red dots and red turbines in Fig. 6 of the paper are indicating that they use flow and turbine power measurements, respectively, rather than their error value. This may have caused confusion. The inflow at each turbine shows a close-to-blue color, indicating that the estimation error in wind speed is very small, which fits with the small error in power estimation from the figures above.

Revised changes: The authors have more carefully emphasized that what is being plotted in Figures 5, 6 and 9 are absolute values of the errors. This was done by additional information in the caption and main body text.

- **Feedback:** Fig 7: it is essential here to simulate using a realistic low frequency variations in the wind condition. Obviously, the prediction will be good when the incoming wind speed and direction does not change, since the underlying model does not change. The different KFs outperform the OL simulation in the prediction ONLY because the initial condition at the beginning of the prediction part is worse for the OL case (the KF based models have been adapted during the estimation phase, and the OL model has not). I suggest you focus in Figure 7 on the estimation part and remove the prediction part. Same holds for Figure 10.

Response: The authors thank the reviewer for the valuable comment and suggestions on further simulation studies. The authors think that it is not obvious that the predictions in Figure 7 will be accurate. Note that only two measurements are available at each timestep, while 1 model parameter and 3200 states are estimated simultaneously. As observability is not a given, the model is nonlinear and can go unstable, the turbines are following a quickly varying excitation signal, and we are feeding in noisy measurements of a significantly different, high-fidelity model, the authors believe it is not a given that the prediction will be accurate.

Furthermore, the purpose of showing the prediction is to highlight the difference in state-only against joint state-parameter estimation. In state-only estimation, indeed only the initial condition is better than the OL case. The state-only estimation forecast will converge to the OL forecast after a certain amount of time (around $t = 800$ s and 1100 s, respectively). However, for the state-parameter estimation case, the estimate should outperform the state-only estimation for larger prediction horizons, showing the importance of parameter estimation in addition to state estimation.

Further, the authors think it is a great suggestion to investigate the performance under a realistic, time-varying and spatially-varying inflow. Currently, the turbulent inflow for SOWFA is generated following a precursor simulation, in which a realistic turbulent flow field is developed. As the reviewer rightfully mentions, while there are turbulent fluctuations in the inflow, this is with a single *mean* wind speed and wind direction. However, to generate a realistic inflow with low-frequency changes in the ambient conditions for a high-fidelity wind farm simulation, one would need to couple a mesoscale model with a large-eddy simulation model. This is a scientific study by itself (e.g., Rodrigo et al. [2016], Santoni et al. [2018]), and considered outside of the scope of this work.

The authors still believe there is value in the work presented in this article. Currently, the solution has been tested in a high-fidelity simulation environment with a realistic, turbulent inflow. Furthermore, convergence of the wind speed is shown, by purposely initializing the model internal to the estimation algorithm with a wrong freestream wind speed, with success. This suggests that in the comparable situation that when the actual freestream wind speed changes, the solution will succeed too.

An alternative to high-fidelity simulation would be to use experimental data. However, such data is currently difficult to obtain for the authors. Furthermore, it is questionable whether a wind tunnel provides an inflow with realistic low-frequency variations in ambient conditions, and flow scaling may become a problem too. Experimental full-scale data would be more sensible, but this will yield other issues such as confidentiality, the scientific relevance of the turbines in the farm, signal synchronization, sensor uncertainties and inconsistencies, and the lack of (reliable) flow and turbine measurements.

Revised changes: The need for high-fidelity data with realistic inflow conditions, including low-frequency changes in the wind direction, turbulence intensity, and wind speed has been presented as an important next step for future work, and the limitations of the current simulations have been described more explicitly. Important remarks and notes have been added at the start of the results section (“Also, note that the simulations... algorithm validation”), at the end of the results section (“A crucial remark... practical wind farm implementation”), and in the conclusion (“In future work... practical wind farm implementation”).

- **Feedback:** Sec 4.3: the performance is compared to the open-loop simulation with the correct wind velocity U_{∞} , while the EnKF is initialized with a different U_{∞} . For better comparison I recommend you compare to the OL simulation with the same initial condition. That will not only be fair, but actually also in favor of your approach.

Response: The reviewer has a good point: for a fair comparison, the authors should “compare apples to apples”: use the same initial conditions in the OL simulation as in the KF simulation.

Revised changes: The authors have added (rather than replaced) the OL simulation results with $U_{\infty} = 9$ m/s, including the necessary discussion and comparisons. Namely, the EnKF will easily outperform the OL simulation with $U_{\infty} = 9$ m/s even if it would only estimate the inflow conditions. It is interesting to also compare how well the EnKF performs compared to a pre-tuned OL simulation. The results are shown in Figures 9, 10, 11, and Table 5.

- **Feedback:** Fig 11. It would make more sense to focus the comparison only on the wind velocity in the wakes, rather than the whole wind field.

Response: The reviewer makes a good point concerning the region of interest when comparing flow fields, and this has been a concern of the authors too in the past. However, the authors believe that it may also be of interest to accurately estimate the non-waked flow surrounding the wind turbines, as this may influence the optimal control strategy in the closed-loop framework later on. Namely, an optimal yaw steering strategy may depend on the non-waked flow conditions on either side of the farm. Furthermore, the definition of a waked region and the extraction of the flow velocities therein requires additional work and explanation, while the dominant trends are identical when considering the complete flow field. Note that the inclusion of the unwaked flow in the surrogate model is important for stability and to reduce boundary effects of the model.

3. **Feedback:** It is claimed on several occasions in the paper that the presented estimator is useful for performing long-term forecasting. I disagree with this statement, and if you want to convince me then you would have to demonstrate the ability of the estimator to make correct predictions in the future when the input conditions (eg ambient wind velocity) vary in time. For instance, to predict upcoming wind speed or direction changes. Obviously, this will not be possible, so I suggest that with respect to the application of the estimator you stick to feedback control. When used for MPC, I suggest using the term short-term prediction.

Response: The reviewer makes a very good point, and there is not sufficient evidence to claim that our proposed solution can consistently and reliably provide long-term forecasting. Furthermore, after revision, the authors agree that the definition “long-term” has not been defined appropriately in the article. While the idea should not be discarded, there is not enough evidence supporting the claim as of right now. Further, the high-fidelity simulation with realistic low-frequency changes in the ambient conditions is outside of the scope of this article.

Revised changes: As suggested, the claims concerning long-term forecasting have been removed. The claims for forecasting, including short-term forecasting, have been rephrased to address the limitations. Namely, time-varying wind directions and wind speeds have not been considered. In addition, the part on wind direction estimation in Section 3.7 has been removed, as it was neither tested in high-fidelity simulation nor with realistic low-frequency changes in the ambient conditions.

Minor comments

- **Feedback:** p.2, line 9 - remove bracket

Response: The authors express gratitude for the time and effort invested by the reviewer to read the document so carefully.

Revised changes: The suggested changes have been made.

- **Feedback:** p.6, line 2 - abbrev ADM seems not used in the sequel, check and if so - remove. Same holds for the abbrev. UT on page 11, line 15.

Revised changes: The suggested changes have been made.

- **Feedback:** p.6, eq. 4: define ϕ , D and C'_{T_i} (what do you mean by variation?). Also, does the term $H[\bullet]$ not imply that the thrust force is exerted within a circle around the turbine, rather than on the rotor plane/line?

Response: With *variation*, a parametrization is meant. There is a one-to-one mapping between the traditional thrust coefficient and C'_{T_i} . The latter has been used more popularly in the work by Goit and Meyers [2015] and Munters and Meyers [2017], and this is the way it has been defined in the original paper by Boersma et al. [2017]. Furthermore, the reviewer is correct that $H[\bullet]$ implies a circle around the turbine center. However, the additional term $\delta[\bullet]$ projects this circle onto the rotor plane, leading to the actuator disk implementation.

Revised changes: The symbols have been defined as suggested.

- **Feedback:** p. 8, bottom: notation already defined in Sec 2.4

Response: The authors appreciate the reviewer's eye for detail.

Revised changes: The repeated definition of symbols have been omitted.

- **Feedback:** p. 9 bottom: why is $P^z_{k|k-1}$ is not necessarily invertible, then L_k will not be full rank (or possibly even $L_k = 0$), implying that some (or all) measurements will not contribute to the state estimation

Response: The reviewer makes a justified comment concerning the invertibility of the covariance matrix, and the state updates that it results in. The issue of invertibility is closely related to how the covariance matrices are calculated. Specifically, for the sample-based algorithms, it may occur that singular covariance matrices arise (see Equation (28)). This is especially the case when the number of samples is smaller than the number of system states. As the reviewer rightfully mentions, this may result into a Kalman gain L_k which is not full rank. In that situation, certain measurements may not be used to update the state vector. Since the samples in the Ensemble KF and thus the rank properties of the covariance matrix change in each timestep due to the random Gaussian noise, this is not expected to be an issue.

- **Feedback:** p.11, eq. (21): define all used notation

Revised changes: The definition of $\bar{\psi}_{k-1|k-1}$ in Equation (21) has been introduced in the text, and the definition of N has been repeated for clarity.

- **Feedback:** p.13, eq (27) : how are the estimates $\hat{\mathbf{w}}_{k-1}^i$ and $\hat{\mathbf{w}}_k^i$ obtained and updated?

Response: The variables $\hat{\mathbf{w}}_{k-1}^i$ and $\hat{\mathbf{v}}_k^i$ are realizations of zero-mean Gaussian white noise, where the covariance is defined through Equation (8), which reads:

$$\mathbb{E} \left[\begin{bmatrix} \mathbf{v}_k \\ \mathbf{w}_k \end{bmatrix} \begin{bmatrix} \mathbf{v}_\ell^T & \mathbf{w}_\ell^T \end{bmatrix} \right] = \begin{bmatrix} \mathbf{R}_k & \mathbf{S}_k^T \\ \mathbf{S}_k & \mathbf{Q}_k \end{bmatrix} \Delta_{k-\ell}, \quad \text{where} \quad \Delta_{k-\ell} = \begin{cases} 1, & \text{if } k = \ell, \\ 0, & \text{otherwise.} \end{cases}$$

In practice, these noise terms are generated using MATLAB's `randn()` command, employing a constant preloaded random seed between simulations for one-to-one comparisons between different Kalman filtering algorithms.

Revised changes: A more explicit explanation has been introduced near Equation (27), detailing how $\hat{\mathbf{w}}_{k-1}^i$ and $\hat{\mathbf{v}}_k^i$ are calculated.

- **Feedback:** p 16, line 1: are you suggesting to use the wind vane measurements of only the upstream turbines to estimate ϕ , or all turbines? Please be clear, because if you use all turbines you will be neglecting the dynamics of the propagation of the wind direction through the wind farm.

Response: The reviewer raises an important question concerning the determination of the wind direction, ϕ . Currently, the wind direction is calculated as the average of the wind vane measurements of *all* turbines inside the farm, both up- and downstream. The issue that the reviewer mentions has not explicitly been considered in this work. Based on the available data from high-fidelity large-eddy simulations, no significant differences were found between the wind vane measurements for the various wind turbines. The main motivation for the use of all the turbine vane measurements was to reduce the variance by using all measurements as they are assumed to measure the same thing with (more or less) independent errors, and to account for changes in the wind direction elsewhere than at the upstream turbines. However, the reviewer rightfully hints that this may not be the case in the situation with realistic, low-frequency changes in the inflow, as also mentioned in earlier comments. It is still uncertain what methodology works best for realistic inflow conditions.

Revised changes: Since the wind direction estimation algorithm has not been tested under the relevant conditions, this wind direction estimation algorithm has been removed from the revised article.

- **Feedback:** p.18, line 14: estimation - ζ you mean “parameter estimation”?

Response: The sentence “*First, the performance of the ExKF, UKF, EnKF, and the case without estimation are compared for the two-turbine simulation case of Table 1.*” is expected to have caused confusion. Actually, in this section, four simulation cases are compared:

- State estimation with Extended Kalman filter (ExKF)
- State estimation with Unscented Kalman filter (UKF)
- State estimation with Ensemble Kalman filter (EnKF)
- Open-loop, no estimation (OL)

None of the cases include parameter estimation. The authors will rephrase this sentence to make it clear.

Revised changes: This sentence has been rephrased for clarity.

- **Feedback:** p.21 Fig 6 caption: “The freestream wind is coming in from the top of the page, and flows towards the bottom.” Isn’t it the other way around?

Response: The authors appreciate the reviewer’s comment, and it now also becomes more apparent that this figure had not been presented clearly enough. The freestream wind flow is coming in from the top of the page and flowing towards the bottom, as correctly stated in the initial paper. The regions in red, i.e., the regions with a higher estimation error, are the waked regions, which are typically harder to predict than the freestream flow. This also corresponds to the way the measurements are defined: turbine power measurements are used in column 2, flow measurements *upstream* of each turbine are used in column 3, and flow measurements *downstream* of each turbine are used in column 4.

Revised changes: There appear to be some artifacts with the images, in which the orientation of the flow fields depend on the document viewer used. The authors have attempted to repair this. In the author’s document viewers, the orientation is as described in the caption.

- **Feedback:** p.22 line 3: “Dual estimation using flow measurements downstream...”
- please explain the used measurements in more detail

Response: The authors appreciate the reviewer’s detailed comments on the clarity of certain sections in the paper. Here, the downstream flow measurements refer back to the same downstream flow measurements presented in Section 4.2.2, as indicated in columns 2-4 in Figure 5 and column 4 of Figure 6.

Revised changes: A clear reference to Section 4.2.2 and Figure 6 has been made on page 22, line 3.

- **Feedback:** p22, line 4: explain figure 7 clearly. What's on the y axis?

Response: The variable on the y -axis refers back to the equation presented on page 19, line 2. Basically, this is the L^2 norm of the estimation error of all the longitudinal velocity states, \mathbf{u}_k , as defined in the equation on page 7, line 13. Note that Figures 5, 6, and 9 are graphical representations of the absolute value of the estimation errors of the same variable \mathbf{u}_k . These states are the most relevant for control, since the dominant wind direction is parallel to the x -axis (thus, aligned with \mathbf{u}_k).

Revised changes: The authors mention that the y -axis shows the 2-norm. Furthermore, the authors have rephrased the captions of Figures 5, 6, and 9, stating that they show representations of the absolute value of the estimation errors of variable \mathbf{u}_k .

- **Feedback:** page 25, equation on bottom: bullet notation not clear

Response: The variable $(\Delta P)^i$ represents the error between the true ("SOWFA") and the estimated ("OL" or "EnKF") turbine power signal timeseries of turbine i . Fundamentally, $(\Delta P)^i$ is the vector with errors between the forecasted and the true power signal over a certain time horizon. This forecast is from the current time instant, T_f , to the final time instant $T_k = 1000$ s. The L^2 norm of the timeseries of this variable for all turbines is shown in Figure 10. This could cause confusion, because it is not straight-forward how these values are derived from the L^2 norms for the single turbines. This is to be clarified in the revised document. Furthermore, the \bullet notation is a placeholder for the respective method used for forecasting. This may be the open-loop forecast ("OL"), but can also be the forecast of the Ensemble Kalman filter ("EnKF").

Revised changes: The authors have decided to remove the definition of $\|(\Delta P)^\bullet\|_2$, and rather put all the results in a separate table (Table 5), with a more insightful description and definition of the variable represented.

References

- S. Boersma, B M Doekemeijer, M Vali, J Meyers, and J W van Wingerden. A control-oriented dynamic wind farm model: WFSim. *Wind Energy Science*, pages 1–34, 2017. doi: 10.5194/wes-2017-44.
- P M O Gebraad, F W Teeuwisse, J W van Wingerden, P A Fleming, S D Ruben, J R Marden, and L Y Pao. Wind plant power optimization through yaw control using a parametric model for wake effects - a CFD simulation study. *Wind Energy*, 19(1): 95–114, 2016. ISSN 1099-1824. doi: 10.1002/we.1822.
- J P Goit and J Meyers. Optimal control of energy extraction in wind-farm boundary layers. *Journal of Fluid Mechanics*, 768:5–50, Apr 2015. doi: 10.1017/jfm.2015.70.
- W Munters and J Meyers. An optimal control framework for dynamic induction control of wind farms and their interaction with the atmospheric boundary layer. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 375(2091), 2017. ISSN 1364-503X. doi: 10.1098/rsta.2016.0100.
- J Sanz Rodrigo, R Aurelio Chávez Arroyo, P Moriarty, M Churchfield, B Kosović, P-E Réthoré, K S Hansen, A Hahmann, J D Mirocha, and D Rife. Mesoscale to microscale wind farm flow modeling and evaluation. *Wiley Interdisciplinary Reviews: Energy and Environment*, 6(2):e214, 2016. doi: 10.1002/wene.214.
- C Santoni, E J Garca-Cartagena, U Ciri, G V Iungo, and S Leonardi. Coupling of mesoscale weather research and forecasting model to a high fidelity large eddy simulation. *Journal of Physics: Conference Series*, 1037(6), 2018.

Online model calibration for a simplified LES model in pursuit of real-time closed-loop wind farm control

Bart Doekemeijer¹, Sjoerd Boersma¹, Lucy Pao², Torben Knudsen³, and Jan-Willem van Wingerden¹

¹Delft Center for Systems and Control, Delft University of Technology, The Netherlands

²Electrical, Computer & Energy Engineering, University of Colorado Boulder, United States of America

³Department of Electronic Systems, Aalborg University, Denmark

Correspondence: B.M. Doekemeijer (B.M.Doekemeijer@tudelft.nl)

Abstract. Wind farm control often relies on computationally inexpensive surrogate models to predict the dynamics inside a farm. However, the reliability of these models over the spectrum of wind farm operation remains questionable due to the many uncertainties in the atmospheric conditions and tough-to-model dynamics at a range of spatial and temporal scales relevant for control. A closed-loop control framework is proposed in which a simplified model is calibrated and used for optimization in real time. This paper presents a joint state-parameter estimation solution with an Ensemble Kalman filter at its core, which calibrates the surrogate model to the actual atmospheric conditions. The estimator is tested in high-fidelity simulations of a nine-turbine wind farm. Using exclusively SCADA measurements, the adaptability to modeling errors and mismatches in atmospheric conditions is shown. Convergence is reached within 400 seconds of operation, after which the estimation error in flow fields is negligible. At a low computational cost of 1.2 s on an 8-core CPU, this algorithm shows comparable accuracy to the state of the art from the literature while being approximately two orders of magnitude faster.

1 Introduction

Over the past decades, global awakening on climate change and the environmental, political and financial issues concerning fossil fuels have been catalysts for the growth of the renewable energy industry. As the primary energy demand in Europe is projected to decrease by 200 million tonnes of oil equivalent from 2016 to 2040, there is an additional shift in the energy source used to meet this demand (International Energy Agency, 2017). Shortly after 2030, onshore and offshore wind energy are projected to become the main source of electricity for the European Union. By then, about 80% of all new capacity added is projected to come from renewable energy sources, enabled by a favorable political climate.

While there are clear benefits in the growth of the wind energy industry, an important problem with wind energy is that the rotational speed of almost all commercial turbines ~~are currently disconnected~~ is currently decoupled from the electricity grid ~~by their frequency via each turbine's~~ power electronics (Aho et al., 2012). As the current grid-connected fossil fuel plants are replaced by ~~grid-disconnected non-synchronous~~ renewable energy plants, the inertia of the electricity grid will decrease. Thus, the grid will become less stable, making it more prone to machine damage and blackouts (Ela et al., 2014). Therefore, there is a strong need for wind farms and other renewables to provide ancillary grid services. Wind farm control aimed at increasing

the grid stability is more commonly defined as active power control (APC). In APC, the power production of a wind farm is regulated to meet the power demand of the electricity grid, which may change from second to second.

Existing literature on wind farm control has ~~focused mainly~~ mainly focused on maximizing the power capture (e.g., Rotea, 2014; Gebraad (e.g., Rotea, 2014; Gebraad and van Wingerden, 2015; Gebraad et al., 2016; Munters and Meyers, 2017). Though, literature on APC has been receiving an increasing amount of attention (e.g., Fleming et al., 2016; Van Wingerden et al., 2017; Boersma et al., 2017a). The main challenges in wind farm control are the large time delays caused by the formation of wakes, the many uncertainties in the atmospheric conditions, and the questionable reliability of surrogate models over the wide spectrum of wind farm operation (see, See Boersma et al. (2017a) and Knudsen et al. (2015) for state-of-the-art overviews of control and control-oriented modeling for wind farms). While there has been success with model-free methods for power maximization (e.g., Rotea, 2014), it is unclear to what degree such methods can be used for power forecasting. Furthermore, model-free methods typically have long settling times, making ~~these methods~~ them intractable for APC. On the other hand, for model-based approaches, the aforementioned challenges make it impossible for any model to reliably provide power predictions in an open-loop setting. Hence, a model-based approach in which a surrogate wind farm model is actively adjusted to the present conditions is a necessity for reliable and computationally tractable APC algorithms. This closed-loop wind farm control framework ~~is displayed~~, consisting of three components, is shown in Fig. 1. ~~The control framework of Fig. 1 requires three components.~~

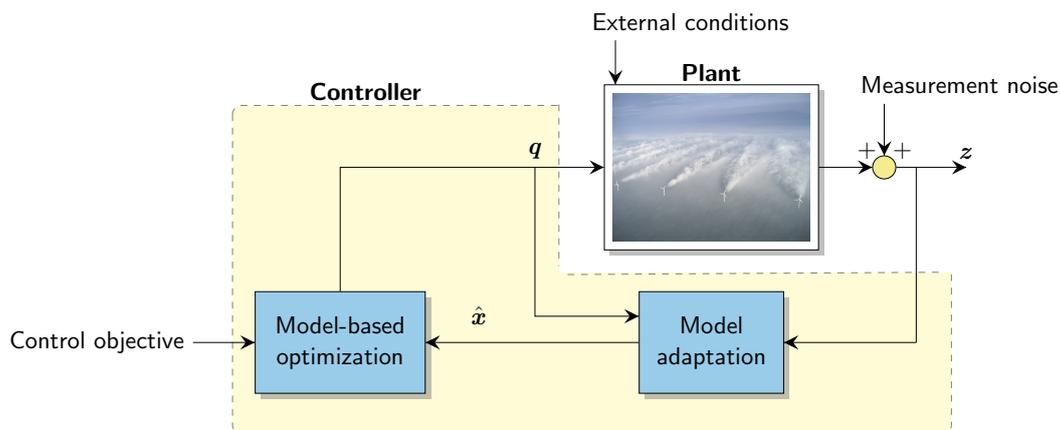


Figure 1. Closed-loop wind farm control framework. Measurements z (e.g., SCADA, met mast, LiDAR data) are fed into the controller ~~block~~. First, the state of the surrogate wind farm model \hat{x} is estimated to represent the actual atmospheric and turbine conditions inside the wind farm. Secondly, using the calibrated model, an optimization algorithm determines the control policy (e.g., yaw angles, blade pitch angles) for all turbines q . This control policy may be a set of constant operating points, but can also be time-varying, depending on whether the surrogate model is time-varying and the employed optimization algorithm. The photograph of the wind farm is from Christian Steiness.

The first component of the closed-loop framework is a computationally inexpensive ~~control-oriented~~ surrogate model that accurately predicts the power production of the wind farm ahead in time, on a time-scale relevant for control. The most

commonly used surrogate models in wind farm control are steady-state models, which are heuristic and neglect all temporal dynamics (Boersma et al., 2017a). ~~Thus, wind farm control algorithms synthesized using such models neglect any transient dynamics in the wind farm, thereby potentially limiting performance.~~ While some of these models have shown success in wind tunnel tests (e.g., Schreiber et al., 2017) and field tests (e.g., Fleming et al., 2017a, b) for power maximization, the actuation frequency is limited to the minutes-scale, since the flow and turbine dynamics are predicted on the minute-scale. Furthermore, time-ahead predictions with these models are limited to the ~~time-invariant steady-state~~ steady state, limiting their use for APC.

¹ There is a smaller yet significant number of dynamic surrogate wind farm models (e.g., Munters and Meyers, 2017; Boersma et al., 2017b; Shapiro et al., 2017a), which attempt to model-include the dominant temporal dynamics inside the farm. These models can be used for control on the seconds-scale, and furthermore allow time-ahead predictions, some even under changing atmospheric conditions. Specifically, the dynamic surrogate model employed in Shapiro et al. (2017a) is computationally feasible, but only models the flow in one dimension, and furthermore allows no turbine yaw or changes in the wind direction, limiting its applicability. Furthermore, the dynamical model in Munters and Meyers (2017) has shown success for closed-loop control applications, but ~~as it is a 3D LES code, it is much~~ it is too computationally costly for any kind of real-time control, and the authors present their results solely as a benchmark case. In the work presented here, the model described in Boersma et al. (2017b) is used, which is a light-weight two-dimensional LES code with wind farm control as its main objective. This dynamic surrogate model, named “WindFarmSimulator” (WFSim), includes yaw and axial induction actuation, turbine-induced turbulence effects, and spatially and temporally varying inflow profiles, with a moderate computational cost (~~10^1 — 10^2 ms per timestep~~).¹

The second component of the closed-loop framework is an algorithm that adjusts the surrogate model’s parameters to improve its accuracy online using flow and/or turbine measurements (e.g., SCADA data, LiDAR measurements, met masts). In terms of control, this turns into a dual-joint estimation problem, in which both the model state and a subset of model parameters are estimated online. Currently, the optimization algorithms presented in Munters and Meyers (2017) and Vali et al. (2017) have assumed full state knowledge, conveniently ignoring the step of model adaptation. Literature on state reconstruction and model calibration for dynamical wind farm models is sparse, limited to linear low-order models and/or common estimation algorithms. Gebraad et al. (2015) designed a traditional Kalman filter (KF) for their low-fidelity “FLORIDyn” model, showing marginal improvements compared to optimization using a static model. Shapiro et al. (2017a) present a one-dimensional dynamic wake model used with receding horizon control for secondary frequency regulation, using an estimation algorithm following Doekemeijer et al. (2016). Furthermore, Jungo et al. (2015) used dynamic mode decomposition to obtain a reduced-order model of the wind farm dynamics, which was then combined with a traditional KF for state estimation. To the best of the authors’ knowledge, none of these methods have explored more sophisticated models such as WFSim, and often only use simple state estimation algorithms that are lacking in terms of accuracy and ~~in terms of~~ computational tractability.

¹ ~~Control using steady-state models is typically limited to an actuation frequency of every 5-10 minutes, depending on the wind speed and size of the farm. Namely, after each change in control settings, it takes minutes before the flow propagates through the farm and a steady-state is formed. In steady-state, SCADA data is then temporally averaged, upon which a new set of control parameters is calculated for the current atmospheric conditions.~~

¹ ~~For a more detailed analysis on the computational cost involved in WFSim, the reader is referred to Boersma et al. (2017b).~~

The third component of the closed-loop framework is an optimization algorithm, which typically is a gradient-based or nonlinear optimization algorithm (e.g., Gebraad et al., 2016; Thomas et al., 2017) (e.g., Gebraad et al., 2016) for steady-state models, and a model-based predictive optimization method for dynamical models (e.g., Goit and Meyers, 2015; Vali et al., 2017; Siniscalchi-Minna et al., 2018). A more in-depth discussion on optimization algorithms for the framework of Fig. 1 is out of the scope of this article, ~~and therefore not further continued here.~~

5 The focus of this work is on a model adaptation algorithm for WFSim, which trades off estimation accuracy with computational complexity ~~for online model calibration~~. In previous work (Doekemeijer et al., 2016, 2017), ~~recursive~~ state estimation using flow measurements downstream of each turbine has shown success using an Ensemble KF (EnKF), with a computational cost several orders of magnitude lower than traditional KF methods. The main contributions of this article specifically are 1) the addition of adaptation to time-varying a mismatch in atmospheric conditions (specifically, the freestream wind speed and turbulence intensity), ~~which is of crucial importance for accurate longer-term forecasting;~~ 2) each the option to use turbine's power ~~signal can now be used~~ signals in addition to, or instead of, flow measurements, ~~as power measurements are readily available in existing farms, in contrast to flow measurements (other than the hub anemometer, which yields low-quality measurements);~~ 3) a further reduction in the computational complexity ~~is further reduced compared to previous work~~, and 4) a comparison of the EnKF with the ~~EnKF algorithm will be compared to the~~ state of the art algorithms in the literature.

15 The structure of this article is as follows. In Section 2, the surrogate model will be ~~described in more detail~~ introduced. In Section 3, a time-efficient, online model calibration algorithm for ~~low- and medium-fidelity~~ dynamical wind farm models is detailed. This ~~online~~ calibration algorithm is ~~tested against validated and compared with~~ standard algorithms in the literature ~~using in~~ high-fidelity ~~simulation data~~ simulations in Section 4. The article is concluded in Section 5.

2 The surrogate model

20 ~~To motivate the choice of model parameters that are to be estimated in real time, the surrogate model used in the work at hand is outlined in this section. The chosen surrogate model is~~ The framework of Fig. 1 requires a surrogate model of the wind farm. In this work, that is the WindFarmSimulator (WFSim) model presented by Boersma et al. (2017b). ~~In short, WFSim solves a modified set of unsteady two-dimensional (2D) Navier-Stokes equations in a horizontal plane at the turbine hub height. This surrogate model is a medium-fidelity nonlinear dynamic wind farm model with a total of 5 tuning parameters. WFSim~~ WFSim has shown success in reconstructing the flow field and turbine power signals of high-fidelity ~~LES data~~ simulation data (Boersma et al., 2017b; Doekemeijer et al., 2017). This model is particularly suited for the framework presented in Fig. 1 as it ~~is dynamic,~~ includes both yaw and axial induction ~~control, handles temporally and spatially varying inflows, and~~ actuation and yields a relatively high accuracy with a ~~manageable~~ relatively low computational cost.¹

30 The scope of this section is to give a summary of the surrogate model, rather than a full derivation and motivation of the assumptions made. The reader is referred to Boersma et al. (2017b) for more information. In Section 2.1, the governing equations of the model are presented. The turbulence and turbine model are described in Sections 2.2 and 2.3, respectively. The

¹Note that it is still uncertain what accuracy is necessary and what computational cost can be permitted for real-time closed-loop wind farm control.

spatial and temporal discretization process is described in Section 2.4; ~~including some remarks about the boundary conditions and the computational tractability of the model.~~

2.1 Governing equations

The WFSim ~~wind farm~~ model is based on the two-dimensional unsteady incompressible Navier-Stokes (NS) equations to maintain computational tractability compared to a three-dimensional model. ~~Furthermore, in WFSim the continuity equation is modified to accommodate for flow dissipation in the neglected vertical dimension.~~ The surrogate model can be described completely by the flow and rotor dynamics in a horizontal plane at hub height, derived from the following set of partial differential equations:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla_H) \mathbf{u} + \nabla_H \cdot \boldsymbol{\tau}_H + \nabla_H \cdot p = \mathbf{f}, \quad \frac{\partial u}{\partial x} + 2 \frac{\partial v}{\partial y} = 0, \quad \text{where } \mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^T, \quad \nabla_H = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}^T, \quad (1)$$

where u and v are the longitudinal and lateral flow velocity respectively, x and y are the spatial coordinates in longitudinal and lateral direction respectively, ~~$\boldsymbol{\tau}_H$ is a 2D tensor containing the horizontal~~ are the subgrid stresses (turbulence model), p is pressure, and \mathbf{f} contains the forcing terms (turbine model) acting on the flow. Equation (1) deviates from the traditional 2D NS equations in two ways. Firstly, the diffusion term is neglected, as it plays a negligible role in the dominant flow dynamics due to the low viscosity of air. Secondly, the term $\frac{\partial v}{\partial y}$ in the continuity equation is multiplied by a factor 2 to approximate flow dissipating in the vertical flow dimension. ~~See Boersma et al. (2017b) for a detailed derivation of.~~ Other vertical flow contributions such as vertical meandering and shear are neglected.

2.2 Subgrid-scale (turbulence) model

~~Boersma et al. (2017b) introduced a new model for the subgrid-scale term $\boldsymbol{\tau}_H$.~~ The subgrid-scale model is formulated using an eddy-viscosity assumption in combination with Prandtl's mixing length model,

$$\boldsymbol{\tau}_H = -\ell_u(x, y)^2 \left| \frac{\partial \mathbf{u}}{\partial y} \right| \cdot \frac{1}{2} (\nabla_H \mathbf{u} + (\nabla_H \mathbf{u})^T), \quad \text{with } \ell_u(x, y) = \begin{cases} G(x'_i, y'_i) * \ell_u^i(x'_i, y'_i), & \text{if } x \in \mathcal{X} \text{ and } y \in \mathcal{Y}, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $\ell_u(x, y) \in \mathbb{R}^+$ is a local spatially varying parametrization of the mixing length, inspired by the high-fidelity simulation results presented in Iungo et al. (2017). $G(x'_i, y'_i)$ is a smoothing pillbox filter with radius 3, $*$ is the 2D spatial convolution operator, and \mathcal{X} and \mathcal{Y} define a rectangular region behind the turbine rotor to which the turbulence model applies, given by

$$\mathcal{X} = \{x : x'_i \leq x \leq x'_i + \cos(\phi) \cdot d\}, \quad \mathcal{Y} = \{y : y'_i - \frac{D}{2} + \sin(\phi) \cdot x'_i \leq y \leq y'_i + \frac{D}{2} + \sin(\phi) \cdot x'_i\},$$

with (x'_i, y'_i) the wind-aligned axis system centered at the turbine rotor, D the ~~turbine~~ rotor diameter, ϕ the mean wind direction in the original (x, y) -axis system, and d a length parameter for the turbulence model. See Fig. 2 for a schematic drawing. Then, $\ell_u^i(x, y)$ is defined as

$$\ell_u^i(x'_i, y'_i) = \begin{cases} (x'_i - d') \ell_s, & \text{if } d' \leq x'_i \leq d \text{ and } -\frac{D}{2} \leq y'_i \leq \frac{D}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where ℓ_s defines the slope of $\ell_u^i(x'_i, y'_i)$, and d' is a second length parameter for the turbulence model. Thus, the entire turbulence model has three tuning parameters: the length parameters d and d' are the upper and lower spatial bounds, respectively, and ℓ_s is a gradient parameter for the mixing length.

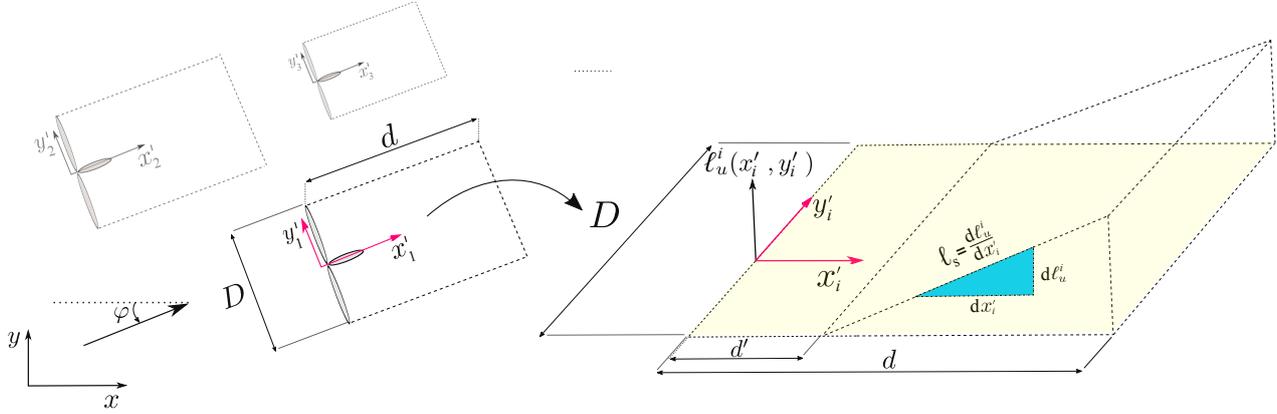


Figure 2. The subgrid-scale model implemented in WFSim employs a spatially varying mixing length parameter that increases with distance behind the rotor. This can be explained by the turbine-induced turbulent structures in the wake. Image courtesy of Boersma et al. (2017b).

2.3 Turbine model

- 5 Turbine forces in WFSim are modeled using the classical non-rotating (static) actuator disk model (ADM), projected onto the 2D plane at hub height. The turbine is assumed to be a rigid object applying a two-dimensional force vector on the flow. The turbine forcing term in (1), \mathbf{f} , at spatial location $\mathbf{s} = \begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^2$ is expressed as

$$\mathbf{f} = \sum_{i=1}^{N_T} \mathbf{f}_i, \quad \text{with} \quad \mathbf{f}_i = \frac{c_f}{2} C'_{T_i} [U_i \cos(\gamma_i)]^2 \begin{bmatrix} \cos(\gamma_i + \phi) \\ \sin(\gamma_i + \phi) \end{bmatrix} \cdot \mathbf{H} \left[\frac{D}{2} - \|\mathbf{s} - \mathbf{t}_i\|_2 \right] \cdot \delta[(\mathbf{s} - \mathbf{t}_i) \cdot \mathbf{e}_{\perp, i}], \quad (4)$$

- with $\mathbf{H}[\bullet]$ the heaviside function, D the rotor diameter, ϕ the mean wind direction on the rotor plane, $\delta[\bullet]$ the Dirac delta function, and $\mathbf{e}_{\perp, i} \in \mathbb{R}^2$ the unit vector perpendicular to the i^{th} rotor disk with position $\mathbf{t}_i \in \mathbb{R}^2$. The scalar C'_{T_i} is a variation of the non-dimensional the thrust coefficient of turbine i which defined according to Meyers and Meneveau (2010). can be related to physical turbine parameters such as the generator torque and blade pitch angles (Goit and Meyers, 2015). The scalar γ_i is the yaw misalignment angle of turbine i with respect to the incoming wind, and U_i is the average flow velocity over the rotor of turbine i . The scalar c_f is a static tuning variable to account for the time-invariant rotor dimensions and containing the rotor swept area and a correction for numerical grid effects, making it the fourth tuning variable in WFSim. The control variables for optimization in the framework of Fig. 1 are γ_i and C'_{T_i} for $i = 1, \dots, N_T$, with N_T the number of turbines. Furthermore, the instantaneous power capture of the wind farm P_{farm} is calculated in a similar approach by

$$P_{\text{farm}} = \sum_{i=1}^{N_T} P_{\text{turb}, i}, \quad \text{with} \quad P_{\text{turb}, i} = \frac{c_p}{2} \rho A C'_{T_i} [U_i \cos(\gamma_i)]^3, \quad (5)$$

with scalar c_p the fifth tuning factor used to account for numerical grid effects and time-invariant turbine losses, and A the rotor swept surface area. Note that C'_T has a direct mapping to the turbine power $P_{\text{turb},i}$, and thus replaces the usual non-dimensional power coefficient, following [the example of Goit and Meyers \(2015\)](#). [Goit and Meyers \(2015\)](#). Note that a cosine-cubed term is used to model the effects of a yaw misalignment. The validity of this correction term is still under investigation (Boersma et al., 2017b).

2.4 Discretization, boundary conditions, and computational cost

Equation (1) is spatially discretized on a quadrilateral grid employing the finite volume method and the hybrid differencing scheme (Boersma et al., 2017b). Temporal discretization is performed using the implicit method, ~~which guarantees stability of the solution.~~ Dirichlet boundary conditions for u and v are applied on one side of the grid for inflow, while Neumann boundary conditions are applied on the remaining sides for the outflow. After discretization, the surrogate wind farm model described in this section reduces to a nonlinear discrete-time deterministic state-space model, described by

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{q}_k), \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{q}_k), \end{aligned}$$

where $\mathbf{x}_k \in \mathbb{R}^N$ is the system state at discrete time instant $k \in \mathbb{Z}$, which is a column vector containing the collocated longitudinal flow velocity at each cell in the domain $\mathbf{u}_k \in \mathbb{R}^{N_u}$, the lateral flow velocity at each cell in the domain $\mathbf{v}_k \in \mathbb{R}^{N_v}$, and the pressure term at each cell in the domain $\mathbf{p}_k \in \mathbb{R}^{N_p}$, with $N = N_u + N_v + N_p$ and $N_u \approx N_v \approx N_p \approx \frac{1}{3}N$. The state \mathbf{x}_k is formulated as

$$\mathbf{x}_k^T = \begin{bmatrix} \mathbf{u}_k^T & \mathbf{v}_k^T & \mathbf{p}_k^T \end{bmatrix}.$$

Empirically, good results have been achieved with cell dimensions of about 30–50 m in width and length, resulting in N with a typical value on the order of 10^3 – 10^4 for [medium-sized wind farms](#) (e.g., [Vali et al., 2016, 2017](#); [Doekemeijer et al., 2016, 2017](#); [Boersma et al., 2017](#)), [six- to nine-turbine wind farms](#) (e.g., [Vali et al., 2017](#); [Doekemeijer et al., 2016, 2017](#); [Boersma et al., 2017b](#)). Such a number of states may seem very small for LES simulations, yet is very high for control purposes. Furthermore, $\mathbf{q}_k \in \mathbb{R}^O$ includes the system inputs, i.e., the turbine control settings γ_i and C'_{T_i} for $i = 1, \dots, N_T$. The system outputs $\mathbf{z}_k \in \mathbb{R}^M$ are defined by sensors. It can include, among others, flow field measurements ($\mathbf{z}_k \subset \mathbf{x}_k$) and power measurements. We define the integer $M_{u,v} \in \mathbb{Z}$ with $M_{u,v} \leq M$ as the total number of flow field measurements. The nonlinear functions f and h are the state forward propagation and output equation, respectively.

The computational cost may vary from 0.02 s for [a small two-turbine wind farm](#) with $N = 3 \cdot 10^3$ states (e.g., [a 2-by-1 wind farm](#) in [Doekemeijer et al. \(2017\)](#)), to 1.2 s for $N = 1 \cdot 10^5$ states for medium-sized wind farms (e.g., [a 3-by-3 wind farm](#) in [Boersma et al. \(2017b\)](#)), for a single time-step forward simulation on a single desktop CPU core. [This computational complexity](#) [The computational complexity of the model](#) is what motivates the use of time-efficient estimation algorithms in [the work at hand](#) [this work](#), and time-efficient predictive control methods for optimization in related work (Vali et al., 2017). [In this](#)

~~work~~Here, the limits of computational cost are explored to maximize model accuracy while still allowing real-time control.
5 Note that research on the computational feasibility of optimization algorithms using WFSim is ongoing.

3 Online model calibration

Due to the limited accuracy of surrogate wind farm models, and due to the many uncertainties in the environment, surrogate models often yield predictions with significant uncertainty of the wind flow and power capture inside a wind farm. Since control algorithms largely rely on such predictions, this may suppress gains or even lead to losses inside a wind farm. Unfortunately, higher-fidelity models are computationally prohibitively expensive for control applications. Hence, rather, lower-fidelity surrogate models are calibrated online using readily available measurement equipment.

In this section, first the challenges for real-time model calibration for the surrogate “WFSim” model described in Section 2 will be highlighted in Section 3.1. Secondly, a mathematical framework for recursive model state estimation will be presented in Section 3.2. Thirdly, a number of state estimation algorithms are presented in Sections 3.3 to 3.6, building up from the industry standard to the state of the art in the literature. Finally, a robust, computationally efficient model calibration solution is synthesized in Section 3.7, which allows the simultaneous estimation of the boundary conditions, model parameters, and the model states of WFSim in real time using readily available measurements from the wind farm.

Note that we will henceforth refer to the estimation of x as *state(-only) estimation*. The estimation of ~~tuning parameters,~~ both model states and model parameters such as ℓ_s and e_f (Section 2), ~~which are included in the expressions for f and h , are~~ considered as is referred to as *parameter (joint) state-parameter estimation*.

3.1 Challenges

Online model calibration for WFSim is challenging for a number of reasons. First of all, the model is nonlinear, and thus the common linear estimation algorithms cannot be used without linearization. ~~While analytical expressions for the linearized surrogate model are available (Boersma et al., 2017b), the absence of linear expressions for the subgrid-scale model and the multiple \max , \min and abs operators in the nonlinear model limit its accuracy, which limits accuracy (Boersma et al., 2017b)~~ Secondly, the surrogate model an estimation solution relying on WFSim is sensitive to instability when the estimated state sufficiently deviates from the continuity equation in (1). ~~In addition, while state estimation (i.e., estimation of the instantaneous flow field) may prove helpful in short-term forecasting, calibration of additional model parameters is necessary (e.g., the inflow/boundary conditions and the turbulence model) for reliable longer-term forecasting, as will be shown in Section 4.~~ Finally, the surrogate model typically has on the order of $N \sim 10^3 - 10^4$ states, which is extraordinarily high for control applications. Though, real-time estimation is a necessity for real-time model-based ~~closed-loop~~ control, and thus one needs to find a trade-off between accuracy on the one hand, while guaranteeing updates at a low computational cost on the ~~seconds-scale on~~ the other hand.

3.2 General formulation

This section details the basics of ~~Kalman filtering~~the Kalman Filter (KF), which is the literature standard for state estimation in control. The goal of a ~~Kalman filter (KF)~~ is to ~~KF is to recursively~~ estimate the unmeasured states of a dynamical system through noisy measurements. Assumed here is a system (the wind farm) represented mathematically by a discrete-time stochastic state-space model with additive noise,

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{q}_k) + \mathbf{w}_k, \quad (6)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{q}_k) + \mathbf{v}_k, \quad (7)$$

10 where k is the time index, $\mathbf{x} \in \mathbb{R}^N$ is the unobserved system state (in this case: the flow and pressure fields inside the wind farm), $\mathbf{z} \in \mathbb{R}^M$ are the measured outputs of the system (e.g., flow measurements, SCADA data), f describes the forward-in-time state propagation mapping, h describes the output equation from state to measurement, $\mathbf{q} \in \mathbb{R}^O$ and $\mathbf{w} \in \mathbb{R}^N$ are the controllable inputs and process noise respectively that drive the system dynamics, and $\mathbf{v} \in \mathbb{R}^M$ is measurement noise. Furthermore, we assume \mathbf{w} and \mathbf{v} to be zero-mean white Gaussian noise with covariance matrices

$$15 \quad \mathbb{E} \left[\begin{bmatrix} \mathbf{v}_k \\ \mathbf{w}_k \end{bmatrix} \begin{bmatrix} \mathbf{v}_\ell^T & \mathbf{w}_\ell^T \end{bmatrix} \right] = \begin{bmatrix} \mathbf{R}_k & \mathbf{S}_k^T \\ \mathbf{S}_k & \mathbf{Q}_k \end{bmatrix} \Delta_{k-\ell}, \quad \text{where } \Delta_{k-\ell} = \begin{cases} 1, & \text{if } k = \ell, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

with \mathbb{E} the expectation operator. Estimates of the state \mathbf{x}_k , denoted by $\hat{\mathbf{x}}_{k|k}$, are computed based on measurements from the real system. Here, $\hat{\mathbf{x}}_{k|\ell}$ means an estimate of the model's state vector \mathbf{x} at time k , using all past measurements and inputs \mathcal{Z}_ℓ , as

$$\hat{\mathbf{x}}_{k|\ell} = \mathbb{E}[\mathbf{x}_k | \mathcal{Z}_\ell], \quad \text{with } \mathcal{Z}_\ell = \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2 \dots \mathbf{z}_\ell, \quad \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2 \dots \mathbf{q}_\ell. \quad (9)$$

20 State estimates are based on the internal model dynamics and the measurements, weighted according to their respective probability distributions. We aim to find an optimal state estimate, in which optimality is defined as unbiasedness, $\mathbb{E}[\mathbf{x}_k - \hat{\mathbf{x}}_k] = 0$, and when the variance of any linear combination of state estimation errors (e.g., the trace of $\mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$) is minimized (Verhaegen and Verdult, 2007).

In reality, the assumed model described by f and h always has mismatches with the true system, and the assumptions in (8) often do not hold. Further, the matrices \mathbf{Q}_k , \mathbf{R}_k , and \mathbf{S}_k are usually not known and rather considered tuning parameters. In practice, the values of \mathbf{R} and \mathbf{Q} are used to shift the confidence levels between the internal model and the measured values. For $\mathbf{R} \ll \mathbf{Q}$, estimations will heavily rely on the measurements, while for $\mathbf{Q} \ll \mathbf{R}$, estimations will mostly rely on the internal model. Kalman filtering remains one of the most common methods of recursive state estimation, as it has proven successful in many applications. KF algorithms typically consist of two steps, namely:

1. A state and output forecast, including their uncertainties (covariances):

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}[f(\mathbf{x}_{k-1}, \mathbf{q}_{k-1}) + \mathbf{w}_{k-1} | \mathcal{Z}_{k-1}], \quad (10)$$

$$\hat{\mathbf{z}}_{k|k-1} = \mathbb{E}[h(\mathbf{x}_k, \mathbf{q}_k) + \mathbf{v}_k | \mathcal{Z}_{k-1}], \quad (11)$$

$$\mathbf{P}_{k|k-1}^x = \text{Cov}(\mathbf{x}_k, \mathbf{x}_k | \mathcal{Z}_{k-1}) = \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T], \quad (12)$$

$$\mathbf{P}_{k|k-1}^z = \text{Cov}(\mathbf{z}_k, \mathbf{z}_k | \mathcal{Z}_{k-1}) = \mathbb{E}[(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T], \quad (13)$$

$$\mathbf{P}_{k|k-1}^{xz} = \text{Cov}(\mathbf{x}_k, \mathbf{z}_k | \mathcal{Z}_{k-1}) = \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T]. \quad (14)$$

In (10) and (11), $\hat{\mathbf{x}}_{k|\ell}$ and $\hat{\mathbf{z}}_{k|\ell}$ are the forecasted system state vector and measurement vector, respectively.

2. An analysis update of the state vector, where the measurements are fused with the internal model:

$$\mathbf{L}_k = \mathbf{P}_{k|k-1}^{xz} \cdot \left(\mathbf{P}_{k|k-1}^z \right)^{-1} \quad (15)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \quad (16)$$

$$\mathbf{P}_{k|k}^x = \text{Cov}(\mathbf{x}_k, \mathbf{x}_k | \mathcal{Z}_k) = \mathbf{P}_{k|k-1}^x - \mathbf{L}_k \mathbf{P}_{k|k-1}^z \mathbf{L}_k^T. \quad (17)$$

Here, $\left(\mathbf{P}_{k|k-1}^z \right)^{-1}$ in (15) is the pseudo-inverse of $\mathbf{P}_{k|k-1}^z$, since this matrix is not necessarily invertible.

~~As can be seen in, a trade-off is made between the measured quantities and the surrogate model using the covariance terms as weights.~~

3.3 Linear Kalman filter

Traditionally, state estimation for linear dynamic models is done using the linear ~~Kalman filter (KF)~~ KF (Kalman, 1960). In the idealized situation where the following three criteria hold: 1) the assumptions on noise in (8) ~~hold are correct~~, 2) the surrogate model f and h perfectly match reality, and 3) f and h are linear in \mathbf{x} and \mathbf{q} , with

$$f(\mathbf{x}_k, \mathbf{q}_k) = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{q}_k,$$

$$h(\mathbf{x}_k, \mathbf{q}_k) = \mathbf{C}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{q}_k,$$

where \mathbf{A}_k , \mathbf{B}_k , \mathbf{C}_k , \mathbf{D}_k are the (possibly time-varying) matrices of the state-space system, then the linear KF is optimal in the sense that it provides unbiased estimates, $\mathbb{E}(\mathbf{x}_k) = \hat{\mathbf{x}}_k$, with minimal mean-square error (the trace of $\mathbf{P}_{k|k}^x$ is minimized).

For (10) to (17), one can derive that in the linear case,

$$\mathbf{P}_{k|k-1}^x = \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1}^x \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}, \quad (18)$$

$$\mathbf{P}_{k|k-1}^z = \mathbf{C}_k \mathbf{P}_{k|k-1}^x \mathbf{C}_k^T + \mathbf{R}_k, \quad (19)$$

$$\mathbf{P}_{k|k-1}^{xz} = \mathbf{P}_{k|k-1}^x \mathbf{C}_k^T + \mathbf{S}_k. \quad (20)$$

If any of the three criteria is not met, optimality of the KF is lost. While points criteria 1 and 2 are practically never met, good results are often still achieved. The crux lies with point criterion 3. Namely, the traditional KF cannot deal with nonlinearity in the surrogate model (\hat{f} and/or \hat{h}).

3.4 Extended Kalman filter (ExKF)

Linearization of the surrogate model is the most straight-forward solution to the issue of **nonlinearity in $f(\mathbf{x}, \mathbf{q})$ and $h(\mathbf{x}, \mathbf{q})$** . **This is what model nonlinearity. In** the Extended KF (ExKF) **does. Here**, the surrogate model is linearized around some point

10 $(\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}})$ w.r.t. \mathbf{x} and \mathbf{q} at every timestep k :

$$f(\mathbf{x}_k, \mathbf{q}_k) \approx \underbrace{\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{x}} \Big|_{\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}}}}_{\mathbf{A}_k} (\mathbf{x}_k - \mathbf{x}^{\text{lin}}) + \underbrace{\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \Big|_{\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}}}}_{\mathbf{B}_k} (\mathbf{q}_k - \mathbf{q}^{\text{lin}}) + f(\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}}),$$

$$h(\mathbf{x}_k, \mathbf{q}_k) \approx \underbrace{\frac{\partial h(\mathbf{x}, \mathbf{q})}{\partial \mathbf{x}} \Big|_{\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}}}}_{\mathbf{C}_k} (\mathbf{x}_k - \mathbf{x}^{\text{lin}}) + \underbrace{\frac{\partial h(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \Big|_{\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}}}}_{\mathbf{D}_k} (\mathbf{q}_k - \mathbf{q}^{\text{lin}}) + h(\mathbf{x}^{\text{lin}}, \mathbf{q}^{\text{lin}}).$$

Using the linearized system matrices \mathbf{A}_k , \mathbf{B}_k , \mathbf{C}_k , \mathbf{D}_k , one can directly apply (10) to (17) for state estimation, where (18) to (20) become approximations instead of equalities. Fundamentally, in the ExKF, the state is assumed to have a Gaussian probability distribution. This variable is propagated through the linearized system dynamics, yielding a posterior distribution which is also Gaussian. Hence, the ExKF can be considered a first-order approximation of the true state probability distribution. Optimality is not guaranteed, and this lower-order approximation can even lead to divergence for some models. Though, the ExKF has shown success in academia and industry (Wan and Van Der Merwe, 2000).

As described in Section 3.1, model linearization is troublesome. Furthermore, for surrogate models with many states such as WFSim, the ExKF has an additional challenge: computational complexity. The operation in (15) includes a matrix inversion with a computational complexity of $\mathcal{O}(M^3)$, and (18) includes two matrix multiplications each with a computational complexity of $\mathcal{O}(N^3)$. As there are significantly fewer measurements than states ($M \ll N$) for the problem at hand, (18) dominates the computational cost. **For example, a mesh in WFSim with 50×25 cells yields a state vector size of $N = 3 \cdot 10^3$, and contributes to about 80–90% of the computational cost for the entire KF cycle, with a CPU time on** **The ExKF has a CPU time in** the order of **$1 \cdot 10^1$ s, about one order of magnitude 10^1 s for a two-turbine wind farm, which may be** too large for online model calibration **if APC is the objective**. To reduce computational cost in the ExKF, the surrogate model and/or the covariance matrix P have to be simplified. This is not further explored here. Instead, two KF approaches will be explored that use the nonlinear system directly for forecasting and analysis updates. Doing so, we circumvent the problems with linearization, and additionally better maintain the true covariance of the system state.

3.5 The Unscented Kalman filter (UKF)

The Unscented Kalman filter (UKF) relies on the so-called “unscented transformation” (UT) to estimate the means and covariance matrices described by (10) to (14). The conditional state probability distribution of \mathbf{x}_k knowing \mathcal{Z}_k is again assumed to be Gaussian. In the UKF, firstly a number of sigma points (also referred to as “particles”) are generated such that their mean is equal to $\hat{\mathbf{x}}_{k|k}$ and their covariance is equal to $\text{Cov}(\mathbf{x}_k, \mathbf{x}_k)$. Secondly, each particle is propagated through the nonlinear system dynamics (f, h). Thirdly, the mean and covariance of the forecasted state probability distribution is again approximated by a weighted mean of these forecasted sigma points (Wan and Van Der Merwe, 2000).

Mathematically, we define the i^{th} particle as $\psi_{k|\ell}^i \in \mathbb{R}^N$, which is a realization of the ~~condition~~-conditional probability distribution of x_k given \mathcal{Z}_ℓ . The UKF follows a very similar forecast and analysis update approach as the traditional KF in (10) to (17), yet applied to a finite set of particles (Wan and Van Der Merwe, 2000).

- 10 1. For the forecast step, a particle-based approach is taken.
- (i) A total of $Y = 2N + 1$ particles, with N equal to the state dimension, are (re)sampled to capture the mean and covariance of the conditional state probability distribution $p[x_{k-1}|\mathcal{Z}_{k-1}]$, by

$$\psi_{k-1|k-1}^i = \begin{cases} \bar{\psi}_{k-1|k-1} & \text{for } i = 1, \\ \bar{\psi}_{k-1|k-1} + \left(\sqrt{(N+\lambda) \cdot \mathbf{P}_{k-1|k-1}^x} \right)_i & \text{for } i = 2, \dots, N+1, \\ \bar{\psi}_{k-1|k-1} - \left(\sqrt{(N+\lambda) \cdot \mathbf{P}_{k-1|k-1}^x} \right)_{i-N-1} & \text{for } i = N+2, \dots, Y, \end{cases} \quad (21)$$

15 where $\lambda = \alpha^2(N + \kappa) - N$ is a scaling parameter, ~~where~~- α determines the spread of the particles around the mean, and κ is a secondary scaling parameter typically set to 0 (Wan and Van Der Merwe, 2000). ~~The~~-The vector $\bar{\psi}_{k-1|k-1}$ is the estimated state vector calculated as $\bar{\psi}_{k-1|k-1} = \sum_{i=1}^Y (w_{\text{mean}}^i \cdot \psi_{k-1|k-1}^i)$, where the weight of each particle's mean w_{mean}^i and covariance $w_{\text{cov.}}^i$ is given by

$$w_{\text{mean}}^i = \begin{cases} \lambda(N+\lambda)^{-1} & \text{for } i = 1, \\ \frac{1}{2}(N+\lambda)^{-1} & \text{otherwise,} \end{cases} \quad w_{\text{cov.}}^i = \begin{cases} \lambda(N+\lambda)^{-1} + (1 - \alpha^2 + \beta) & \text{for } i = 1, \\ \frac{1}{2}(N+\lambda)^{-1} & \text{otherwise,} \end{cases}$$

20 ~~where and~~ β is used to incorporate prior knowledge on the probability distribution. In this work, $\beta = 2$ is assumed, which is stated to be optimal for Gaussian distributions (Wan and Van Der Merwe, 2000).

- (ii) Each particle is propagated forward in time using the expectation of the nonlinear model, as

$$\begin{aligned} \psi_{k|k-1}^i &= f(\psi_{k-1|k-1}^i, \mathbf{q}_{k-1}) & \text{for } i = 1, \dots, Y, \\ \zeta_{k|k-1}^i &= h(\psi_{k-1|k-1}^i, \mathbf{q}_k) & \text{for } i = 1, \dots, Y, \end{aligned} \quad (22)$$

where $\zeta_{k|\ell}^i$ is defined as the system output corresponding to the particle $\psi_{k|\ell}^i$.

- (iii) The expected state $\bar{\psi}$ and expected output $\bar{\zeta}$ are calculated as

$$\begin{aligned} \hat{x}_{k|k-1} &= \bar{\psi}_{k|k-1} = \sum_{i=1}^Y (w_{\text{mean}}^i \cdot \psi_{k|k-1}^i), \\ \hat{z}_{k|k-1} &= \bar{\zeta}_{k|k-1} = \sum_{i=1}^Y (w_{\text{mean}}^i \cdot \zeta_{k|k-1}^i), \end{aligned} \quad (23)$$

and the covariance matrices are (re-)estimated from the forecasted ensemble by

$$\mathbf{P}_{k|k-1}^x = \sum_{i=1}^Y \left(w_{\text{cov.}}^i \cdot \left(\psi_{k|k-1}^i - \bar{\psi}_{k|k-1} \right) \left(\psi_{k|k-1}^i - \bar{\psi}_{k|k-1} \right)^T \right) + \mathbf{Q}_{k-1}, \quad (24)$$

$$\mathbf{P}_{k|k-1}^z = \sum_{i=1}^Y \left(\mathbf{w}_{\text{cov.}}^i \left(\boldsymbol{\zeta}_{k|k-1}^i - \bar{\boldsymbol{\zeta}}_{k|k-1} \right) \left(\boldsymbol{\zeta}_{k|k-1}^i - \bar{\boldsymbol{\zeta}}_{k|k-1} \right)^T \right) + \mathbf{R}_k, \quad (25)$$

$$\mathbf{P}_{k|k-1}^{xz} = \sum_{i=1}^Y \left(\mathbf{w}_{\text{cov.}}^i \left(\boldsymbol{\psi}_{k|k-1}^i - \bar{\boldsymbol{\psi}}_{k|k-1} \right) \left(\boldsymbol{\zeta}_{k|k-1}^i - \bar{\boldsymbol{\zeta}}_{k|k-1} \right)^T \right) + \mathbf{S}_k. \quad (26)$$

10 2. For the analysis step, one can apply the same equations as in (15) to (17).

The UKF has been shown to consistently outperform the ExKF in terms of accuracy, since it uses the nonlinear model for ~~model~~ forecasting and covariance propagation. However, this does come at ~~a~~ an increased computational cost. Namely, $Y = 2N + 1$ particles are required to capture the mean and covariance of the conditional state probability distribution. This implies that $2N + 1$ function evaluations are required for each UKF update. Even for a small 2-turbine wind farm in WFSim, $N = 3 \cdot 10^3$, ~~one function evaluation takes approximately 0.02 s. This means that a lower limit on the a~~ computational cost of ~~the UKF algorithm is~~ $1 \cdot 10^2$ s ~~on a single core for a single timestep forward simulation per iteration~~ ($k \rightarrow k + 1$) ~~would not be surprising.~~ While (22) can easily be parallelized, computational complexity remains troublesome, especially for larger wind farms. ~~Rather, a more computationally efficient particle-based KF algorithm is investigated. This is the Ensemble Kalman filter described in Section 3.6. The issue of computational complexity is tackled by the Ensemble KF.~~

20 3.6 The Ensemble Kalman filter (EnKF)

The Ensemble Kalman filter (EnKF) (Evensen, 2003) is very similar to the UKF in that it relies on a finite number of realizations (the “sigma points” or “particles” in the UKF) to approximate the mean and covariance of the conditional probability distribution of \mathbf{x}_k knowing \mathcal{Z}_k . However, whereas the UKF relies on a systematic way of distributing the particles such that the mean and covariance of the conditional probability distribution $p[\mathbf{x}_k | \mathcal{Z}_k]$ are equal to that of the particles, the EnKF relies
25 on random realizations, without guarantees that the mean and covariance are captured accurately. Though, the EnKF has been shown to work well in a number of applications, with typically far fewer particles than states, i.e., $Y \ll N$ (e.g., Houtekamer and Mitchell, 2005; Gillijns et al., 2006). The forecast and update step are very similar to that of the UKF, namely:

1. In the UKF the particles are redistributed at every timestep, in contrast to the EnKF. Rather, the EnKF propagates the particles forward without redistribution. We define the i^{th} particle as $\boldsymbol{\psi}_{k|\ell}^i \in \mathbb{R}^N$, which is a realization of the conditional probability distribution $p[\mathbf{x}_k | \mathcal{Z}_\ell]$. The forecast step is:

(i) Each particle is propagated forward in time using the nonlinear system dynamics, and with the realizations of noise terms \mathbf{w} and \mathbf{v} denoted by $\hat{\mathbf{w}}_{k-1}^i \in \mathbb{R}^N$ and $\hat{\mathbf{v}}_k^i \in \mathbb{R}^M$, ~~respectively generated using MATLABs randn(...) function.~~

$$\begin{aligned} \boldsymbol{\psi}_{k|k-1}^i &= f(\boldsymbol{\psi}_{k-1|k-1}^i, \mathbf{q}_{k-1}) + \hat{\mathbf{w}}_{k-1}^i & \text{for } i = 1, \dots, Y, \\ \boldsymbol{\zeta}_{k|k-1}^i &= h(\boldsymbol{\psi}_{k|k-1}^i, \mathbf{q}_k) + \hat{\mathbf{v}}_k^i & \text{for } i = 1, \dots, Y. \end{aligned} \quad (27)$$

(ii) The expected state and output are calculated identically as in the UKF using (23) with $\mathbf{w}_{\text{mean}}^i = (Y - 1)^{-1}$. The covariance matrices are (re-)estimated from the forecasted ensemble, by

$$\mathbf{P}_{k|k-1}^z = \frac{1}{Y-1} \sum_{i=1}^Y \left(\left(\zeta_{k|k-1}^i - \bar{\zeta}_{k|k-1} \right) \left(\zeta_{k|k-1}^i - \bar{\zeta}_{k|k-1} \right)^T \right), \quad (28)$$

$$\mathbf{P}_{k|k-1}^{xz} = \frac{1}{Y-1} \sum_{i=1}^Y \left(\left(\zeta_{k|k-1}^i - \bar{\zeta}_{k|k-1} \right) \left(\psi_{k|k-1}^i - \bar{\psi}_{k|k-1} \right)^T \right). \quad (29)$$

10 2. For the analysis step, one applies (15) to determine the Kalman gain \mathbf{L}_k . Then, each particle is updated individually, as

$$\psi_{k|k}^i = \psi_{k|k-1}^i + \mathbf{L}_k \left(z_k - \zeta_{k|k-1}^i \right) \quad \text{for } i = 1, \dots, Y. \quad (30)$$

Note that, in contrast to the ExKF and the UKF, the state covariance matrix \mathbf{P}^x (see (12) and (17)) need not be calculated explicitly in the EnKF. This, in combination with the small number of particles $Y \ll N$, is what makes the EnKF computationally superior to the UKF (and often also computationally superior to the ExKF). However, this reduction in computational complexity comes at a price. The disadvantages of the EnKF are discussed in the next section.

3.6.1 Challenges in the EnKF for small number of particles

The caveat to representing the conditional state probability distribution with fewer particles than states, $Y \ll N$, is the formation of inbreeding and long-range spurious correlations (Petrie, 2008). The former, inbreeding, is defined as a situation where the state error covariance matrix \mathbf{P}^x is consistently underestimated, leading to state estimates that incorrectly rely more on the internal model. One straight-forward method to address this is called ‘‘covariance inflation’’, in which \mathbf{P}^x (or rather, the ensemble from which \mathbf{P}^x is calculated) is ~~sealed (the ensemble is ‘‘inflated’’)~~ to correct for the underestimated state uncertainty (Petrie, 2008). Mathematically, this is achieved by applying

$$\psi_{k|k-1}^i = \bar{\psi}_{k|k-1} + r \left(\psi_{k|k-1}^i - \bar{\psi}_{k|k-1} \right) \quad \text{for } i = 1, \dots, Y, \quad (31)$$

before the analysis step, with $r \in \mathbb{R}$ the inflation factor, typically with a value of 1.01 – 1.25.

The latter problem, long-range spurious correlations, can be better visualized in Fig. 3. In particle-based approaches, the covariance terms cannot be captured exactly. This may lead to the formation of small yet nonzero covariance terms between states and outputs which, in reality, are uncorrelated. ~~Then, these states will be adapted according to uncorrelated measurements, which This~~ can lead to the drift of unobservable states ~~(states for which no information is available). This state drift can build up and lead, and eventually~~ to instability of the ~~surrogate model~~ KF. Increasing the number of particles is the most straight-forward solution to this problem, but comes at a huge computational cost. A better alternative is ‘‘covariance localization’’, where physical knowledge of the states and measurements is used to steer the sample-based covariance matrices. Recall that in the surrogate model of Section 2, the model states are the velocity and pressure terms inside the wind farm at a physical location. Define that the i^{th} state entry $(\mathbf{x}_k)_i$ belongs to a physical location in the farm s_i . Then, looking at an arbitrary state

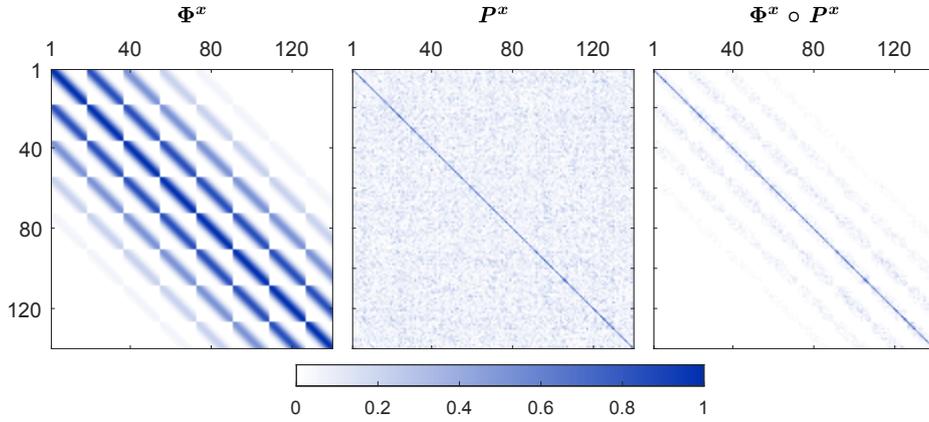


Figure 3. Long-range spurious correlations arise in the case where a covariance matrix is described by a small number of particles. Using physical knowledge of the system, these undesired correlations can be corrected. Φ^x is the localization matrix. Applying localization, the covariance of physically nearby states are multiplied with a value close to 1, and the covariance of physically distant states are multiplied with a value close to 0. In our example case, this results in the localized covariance matrix $\Phi^x \circ P^x$, where \circ is the element-wise product.

covariance term (i, j) ,

$$\left(P_{k|k-1}^x \right)_{i,j} = E \left[\left((\mathbf{x}_k)_i - (\hat{\mathbf{x}}_{k|k-1})_i \right) \left((\mathbf{x}_k)_j - (\hat{\mathbf{x}}_{k|k-1})_j \right)^T \right],$$

we define the physical distance between these two states as $\Delta s_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\|_2$. Now, we introduce a weighting factor into our covariance matrices by multiplying physically distant states with a value close to 0, and multiplying physically nearby states with a value close to 1. A popular choice for such a weighting function is Gaspari-Cohn's fifth-order discretization of a Gaussian distribution (Gaspari and Cohn, 1999), given by

$$\phi(c_{i,j}) = \begin{cases} -\frac{1}{4}c_{i,j}^5 + \frac{1}{2}c_{i,j}^4 + \frac{5}{8}c_{i,j}^3 - \frac{5}{3}c_{i,j}^2 + 1 & \text{if } 0 \leq c_{i,j} \leq 1, \\ \frac{1}{12}c_{i,j}^5 - \frac{1}{2}c_{i,j}^4 + \frac{5}{8}c_{i,j}^3 + \frac{5}{3}c_{i,j}^2 - 5c_{i,j} + 4 - \frac{2}{3}\frac{1}{c_{i,j}} & \text{if } 1 < c_{i,j} \leq 2, \\ 0 & \text{otherwise,} \end{cases} \quad (32)$$

with $c_{i,j} = \frac{\|\Delta s_{i,j}\|_2}{L}$ a normalized distance measure, with L the cut-off distance. Applying (32) for the covariance matrices $P_{k|k-1}^z$ and $P_{k|k-1}^{xz}$ (note that the state covariance matrix $P_{k|k-1}^x$ is not calculated explicitly in the EnKF, but could be calculated similarly), we can define the localization matrices

$$\Phi^z = \begin{bmatrix} \phi(c_{1,1}^z) & \cdots & \cdots & \phi(c_{1,M}^z) \\ \vdots & \ddots & & \\ \phi(c_{M,1}^z) & & & \phi(c_{M,M}^z) \end{bmatrix}, \quad \Phi^{xz} = \begin{bmatrix} \phi(c_{1,1}^{xz}) & \cdots & \cdots & \phi(c_{1,M}^{xz}) \\ \vdots & \ddots & & \\ \phi(c_{N,1}^{xz}) & & & \phi(c_{N,M}^{xz}) \end{bmatrix},$$

where $c_{i,j}^z$ is the normalized distance between two measurements i and j , and $c_{i,j}^{xz}$ is the normalized distance between state i and measurement j , respectively. Finally, localization and inflation can be incorporated into (28) and (29) by

$$5 \quad \mathbf{P}_{k|k-1}^z = \Phi^z \circ \frac{1}{Y-1} \sum_{i=1}^Y \left(\left(\zeta_{k|k-1}^i - \bar{\zeta}_{k|k-1} \right) \left(\zeta_{k|k-1}^i - \bar{\zeta}_{k|k-1} \right)^T \right), \quad (33)$$

$$\mathbf{P}_{k|k-1}^{xz} = \sqrt{rr} \cdot \Phi^{xz} \circ \frac{1}{Y-1} \sum_{i=1}^Y \left(\left(\zeta_{k|k-1}^i - \bar{\zeta}_{k|k-1} \right) \left(\psi_{k|k-1}^i - \bar{\psi}_{k|k-1} \right)^T \right), \quad (34)$$

where \circ is the element-wise product (Hadamard) of the two matrices. The improvement in terms of computational efficiency and estimation performance is displayed in Fig. 4. A ~~very~~ significant increase in performance is shown, especially for smaller

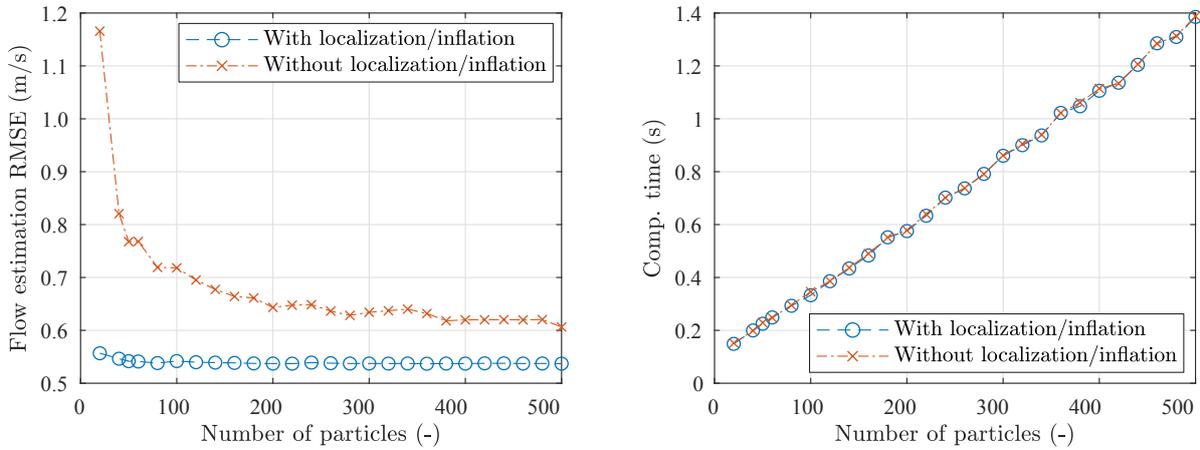


Figure 4. This figure shows the estimation performance and computational cost (parallelized, 8 cores) of the EnKF for a range of ensemble sizes, with and without inflation and localization. Great improvement is seen for estimation accuracy, at no additional computational cost. The simulation scenario is described in detail in Section 4.2, and the results presented here are rather meant as an indication.

numbers of particles. This is in agreement with what was seen in previous work (Doekemeijer et al., 2017). Furthermore, performance is ~~much~~ more consistent. Additionally, note that there is no increase in computational cost, as the covariance matrices are ~~sparsified~~ made sparse, leading to a cost reduction in the calculation of (15), which makes up for the extra operations of (33) and (34). Also, note that the localization matrices are time-invariant and can be calculated offline.

3.7 Synthesizing an online model calibration solution

~~As mentioned in Section 3.1, parameter estimation may be even of a higher importance than state estimation for longer-term forecasting. Parameter estimation~~ Certain model parameters such as ℓ_s are closely related to the turbulence intensity, which vary over time. Estimation of such parameters is achieved by extending the state vector with (a subset of) the model parameters. In this work, ~~the model parameter ℓ_s (turbulence mixing length factor)~~ the model parameter ℓ_s (turbulence mixing length factor) is concatenated to the state vector. Higher values of ℓ_s lead to more ~~mixing behind the turbines, yielding more~~ wake recovery, making the calibration solution adaptable to varying turbulence levels. This adds one scalar entry to ~~\mathbf{x}_k~~ \mathbf{x}_k , which is a negligible addition in terms of computational cost.

~~The Furthermore, a proposal is made for the estimation of the freestream wind speed U_∞ and direction ϕ in the wind farm are also estimated in this framework.~~ This is suggested to be done using the turbine's ~~wind vane and~~ power generation measurements, following the ideas of Gebraad et al. (2016) and Shapiro et al. (2017b). Using the wind vanes, ~~ϕ can be~~
5 ~~calculated as the average of the wind vane measurements. Knowing this,~~ and employing a simple steady-state wake model from the literature (Mittelmeier et al., 2017), the turbines operating in freestream flow can be distinguished from the ones operating in waked flow. Next, define $\text{III} \in \mathbb{Z}^{\mathcal{K}}$ as a vector specifying the upstream turbines, with ~~$\mathcal{K} \in \mathbb{Z}^{\mathcal{K}}$~~ the total number of turbines operating in freestream. Then, the instantaneous rotor-averaged flow speed at each turbine's hub can be estimated using the inverse relationship of (5). One wind-farm-wide freestream wind speed U_∞ is then calculated using actuator disk
10 theory. Smoothing results with a low-pass filter ~~action~~ on the average of U_{∞_i} for each upstream turbine i , we obtain

$$c_{u_\infty} \frac{\partial U_\infty}{\partial t} = \frac{1}{\mathcal{K}} \sum_{i \in \text{III}} \left(\sqrt[3]{\frac{P_{\text{turb},i}^{\text{meas.}}}{\frac{c_p}{2} \rho A C'_{T_i} \cos(\gamma_i)^3}} \cdot \left(1 + \frac{1}{4} C'_{T_i} \right) \right) - U_\infty, \quad (35)$$

where ~~we used actuator disk theory for the identity~~

$$U_{\infty_i} \approx U_{r_i} \left(1 + 0.25 \cdot C'_{T_i} \right), \text{ when } \gamma_i \approx 0.$$

Furthermore, ~~c_{u_∞} is the time constant of the first-order low-pass filter, and $P_{\text{turb},i}^{\text{meas.}}$ is the measured instantaneous power capture of turbine i . While the assumption $\gamma = 0$ is made here for the calculation of U_∞ , research is~~ it is assumed that
15 $U_{\infty_i} \approx U_{r_i} \left(1 + \frac{1}{4} \cdot C'_{T_i} \right)$, when $\gamma_i \approx 0$. Research is currently ongoing on how to best incorporate the effects of turbine yaw ($\gamma \neq 0$) into the definition of C'_{T_i} . ~~Furthermore, c_{u_∞} is the time constant of the first-order low-pass filter, and $P_{\text{turb},i}^{\text{meas.}}$ is the measured instantaneous power capture of turbine i .~~²

~~An important remark is that this methodology for the estimation of U_∞ relies solely on power measurements, and therefore~~
20 ~~only works for below-rated conditions. For estimation of U_∞ in above-rated conditions, one may, for example, require the implementation of a wind speed estimator on each individual turbine, from which the local wind speed in front of each turbine can be estimated, as demonstrated by Simley and Pao (2016).~~

Combining these elements yields an efficient, modular, and accurate model calibration solution for ~~low- and medium-fidelity dynamic wind farm models. The natural~~ WFSim. The model states are estimated using SCADA and/or LIDAR data ~~inside a~~
25 ~~wind farm~~, of which the former is readily available, and the latter becoming more popular. State estimation ~~greatly improves short-term forecasting, important for the small timescales involved in active power control for wind farms. Furthermore, model parameters can be estimated online in parallel with the states, and is required for accurate long-term forecasting and important for active power control at lower frequencies.~~ paired with parameter estimation improves the accuracy of the surrogate model, potentially leading to more accurate control. Additionally, the freestream ~~conditions (boundary conditions in our surrogate model, see Section 2.4) are~~ wind speed is estimated using readily available SCADA data.

5 This control solution is implemented in MATLAB, ~~but and~~ leverages the numerically efficient precompiled solvers ~~and parallelization~~ for model propagation. ~~Furthermore, the forecasting step of is parallelized, making the EnKF easily scalable up~~

² Note that this method for the estimation of U_∞ relies solely on power measurements, and therefore only works for below-rated conditions. For estimation of U_∞ in above-rated conditions, one may require the implementation of a wind speed estimator on each turbine (e.g., Simley and Pao (2016)).

~~to Y-cores. This, in combination with covariance localization and inflation, makes the EnKF~~ The EnKF is orders of magnitude faster than existing estimation algorithms due to covariance localization and inflation, while competing with the UKF in terms of accuracy.

10 4 Results

In this section, the ~~model~~ calibration solution detailed in Section 3 will be validated using high-fidelity ~~simulation data~~ simulations. First, the ~~simulation tool~~ model used to generate the ~~high-fidelity~~ validation data will be described in Section 4.1. Then, a two-turbine and a nine-turbine simulation case are presented in Sections 4.2 and 4.3, respectively.

Note that for the ~~all~~ presented results, pressure terms are ignored in the state vector, as they appeared unnecessary for ~~the~~ estimation of flow fields and powers state estimation in previous work (Doekemeijer et al., 2017). Furthermore, for simplicity and due to lack of information ~~about the true system~~, the process and measurement noise will be assumed to be uncorrelated, ~~i.e.,~~ $S_k = 0$, and Q_k and R_k are assumed to be time-invariant and diagonal. Also, note that the simulations presented are not conclusive on the feasibility of the solution under all relevant conditions experienced in an operational wind farm. Rather, this work presents a first step towards algorithm validation.

20 4.1 SOWFA

High-fidelity simulation data is generated using the Simulator fOr Wind Farm Applications (SOWFA), developed by the National Renewable Energy Laboratory (NREL). ~~This wind farm model provides highly-~~ SOWFA provides accurate flow data at a fraction of the cost of field tests. ~~SOWFA~~ It solves the filtered, three-dimensional, unsteady, incompressible Navier-Stokes equations over a finite temporal and spatial mesh, accounting for the Coriolis and geostrophic forcing terms. SOWFA is a ~~large-eddy simulation solver, meaning that larger scale dynamics are resolved directly, but and~~ turbulent structures smaller than the discretization are approximated using subgrid-scale models to suppress computational cost. ~~Coriolis and geostrophic forcing terms are included in SOWFA (Churchfield et al., 2016)(Churchfield et al., 2012).~~ The turbine rotor is modeled using an actuator line representation as derived from Sorensen and Shen (2002). ~~In the actuator line model (ALM), the rotor blades are discretized spatially along their radial lines, where lift and drag forces are determined based on the incoming flow angle, flow velocity, and blade (airfoil) geometry (Fleming et al., 2015).~~

SOWFA has previously been used for lower-fidelity model validation, controller testing, and to study the aerodynamics in wind farms (e.g., Fleming et al., 2015, 2016, 2017a; Gebraad et al., 2016, 2017) (e.g., Fleming et al., 2016, 2017a; Gebraad et al., 2017). The interested reader is referred to Churchfield et al. (2012) for a more in-depth description of SOWFA and LES solvers in general.

4.2 2-turbine ALM with turbulent inflow

5 In this section, a two-turbine wind farm is simulated to ~~highlight the need for state and model parameter estimation, analyze the effect of different measurement sources, KF algorithms, and the difference between state-only~~ and ~~to motivate the use for~~

the EnKF state-parameter estimation. This simple wind farm contains two NREL 5-MW baseline turbines with $D = 126.4$ m, separated ~~five turbine diameters apart~~ $5D$ in stream-wise direction. This LES simulation ~~has been used before in the literature and~~ was described in more detail in Annoni et al. (2016b). ~~Several important~~ Important simulation properties are listed in Table 1 for SOWFA and WFSim. The effect of the turbulence intensity on the wake dynamics in SOWFA is captured in WFSim through its mixing-length turbulence model. In these simulations, WFSim is purposely initialized with a too low value for ℓ_s in order to represent the realistic situation of a model mismatch. The remaining tuning parameters in WFSim were chosen such that a weighted-sum cost function of the power and flow errors was minimized.

Table 1. Overview of several settings for the SOWFA and the WFSim 2-turbine wind farm simulation.

Variable	Symbol	SOWFA	WFSim
Domain size	-	3.0km \times 3.0km \times 1.0km	1.9km \times 0.80km
Number of states $N \mathcal{O}(10^8) \rightarrow \mathcal{O}(10^9)$ Cell size near rotors	-	3m \times 3m \times 3m	38m \times 33m
Cell size outer regions	-	12m \times 12m \times 12m	38m \times 33m
Rotor model	-	ALM	ADM ($c_f = 1.4$, $c_p = 0.95$)
Inflow wind speed	U_∞	8.0 m/s	8.0 m/s
Atmospheric turbulence	-	Turbulent inflow, $TI_\infty = 5.0\%$	$d' = 1.8 \cdot 10^2$ m, $d = 6.1 \cdot 10^2$ m, $\ell_s = 1.8 \cdot 10^{-2}$

Firstly, the three KF variants will be compared ~~for state estimation~~ in Section 4.2.1. Secondly, in Section 4.2.2, estimation using different information (~~sensor~~) sources is compared. Thirdly, the ~~strength of simultaneous~~ potential of joint state-parameter estimation is displayed in Section 4.2.3.

4.2.1 A comparison of the KF variants for state estimation

~~First, the performance of the ExKF, UKF~~ In this simulation study, four estimation cases are compared: 1) the ExKF, 2) the UKF, 3) the EnKF, and ~~the case without estimation (denoted as 4) the open-loop , or "OL")~~ are compared for the two-turbine ~~simulation case of Table 1. This simulation only focuses on estimation of the model states, not the model parameter (OL) simulation, i.e., without estimation. The focus here is on state-only estimation, thus excluding ℓ_s .~~ Flow measurements downstream of each turbine are assumed (e.g., using LiDAR), their locations denoted as red dots in Fig. 5, which is about 2% of the full to-be-estimated state space. These measurements are artificially disturbed by zero-mean white noise with $\sigma = 0.10$ m/s. The KF settings are listed in Tables 2 and 3. The KF covariance matrices were obtained through an iterative tuning process in previous work (Doekemeijer et al., 2017) with minor adjustments, to simulate performance for untrained data. Figure 5 shows state (flow field) estimation of the three KF variants for two time instants, $t = 300$ s and $t = 700$ s. In this figure, $(\Delta \mathbf{u})_\bullet \in \mathbb{R}^{N_u}$ is defined as the ~~absolute~~ error between the estimated and true longitudinal flow velocities in the field, ~~given by~~

$$(\Delta \mathbf{u})_\bullet = |\mathbf{u}_\bullet - \mathbf{u}_{\text{SOWFA}}|.$$

Table 2. Covariance settings for the KF variants, with \mathbf{I}_\bullet the $\mathbb{R}^{\bullet \times \bullet}$ identity matrix. The full cov. matrices are diagonal concatenations of the entries. For example, P_0 is $\text{diag}(P_{0,u}, P_{0,v})$ and $\text{diag}(P_{0,u}, P_{0,v}, P_{0,\ell_s})$ for state-only and state-parameter estimation, respectively.

Variable	Symbol	Units	Value
Init. state error cov. of \mathbf{u}_k	$P_{0,u}$	(m/s) ²	$1.0 \cdot 10^{-1} \cdot \mathbf{I}_{N_u}$
Init. state error cov. of \mathbf{v}_k	$P_{0,v}$	(m/s) ²	$1.0 \cdot 10^{-1} \cdot \mathbf{I}_{N_v}$
Init. state error cov. of ℓ_{s_k}	P_{0,ℓ_s}	—	$5.0 \cdot 10^{-1}$
Model error cov. of \mathbf{u}_k	$Q_{0,u}$	(m/s) ²	$1.0 \cdot 10^{-2} \cdot \mathbf{I}_{N_u}$
Model error cov. of \mathbf{v}_k	$Q_{0,v}$	(m/s) ²	$1.0 \cdot 10^{-4} \cdot \mathbf{I}_{N_v}$
Model error cov. of ℓ_{s_k}	Q_{0,ℓ_s}	—	$1.0 \cdot 10^{-4}$
Meas. error cov. of flow	$R_{u,v}$	(m/s) ²	$1.0 \cdot 10^{-2} \cdot \mathbf{I}_{M_{u,v}}$
Meas. error cov. of P	R_P	(W) ²	$1.0 \cdot 10^8 \cdot \mathbf{I}_{N_T}$

Table 3. Choice of tuning parameters for the KF variants, for both the 2-turbine and 9-turbine simulation case. Note that the ExKF does not support power measurements nor parameter estimation due to the lack of linearization, and does not have any additional tuning parameters. In terms of computational cost: simulations were run on a single node using 8 cores in parallel.

	2-turb.		2-turb.		9-turb.		
Variable	ExKF	UKF	EnKF	EnKF	EnKF	EnKF	
Number of particles, Y	—	4275	50	50	50	50	
Tuning parameters	—	α	1.0	L	131 m	L	131 m
		β	2.0	r	1.025	r	1.025
		κ	0				
Comp. cost/it.	16.2 s	14.0 s	0.25 s	0.25 s	1.2 s	1.2 s	

Looking at Fig. 5, the open-loop estimations are accurate for the unwaked and single waked flow, yet are lacking in the situation of two overlapping wakes, for which the KFs correct. There is no significant difference in accuracy between the different KF variants, yet they differ by two orders of magnitude in computational cost (Table 3).

4.2.2 A comparison of sensor configurations

Previous results (Doekemeijer et al., 2016, 2017) have relied on flow measurements for state estimation. However, in existing wind farms, such measurements are typically not available. Rather, readily available SCADA data should be used for the purpose of model calibration. For this reason, state estimation with the EnKF leveraging instantaneous turbine power measurements, using an upstream-pointing LiDAR, and using a downstream-pointing LiDAR are compared in Fig. 6. Flow and power measurements are artificially disturbed by zero-mean white Gaussian noise with $\sigma = 0.10$ m/s and $\sigma = 10^4$ W, respectively.

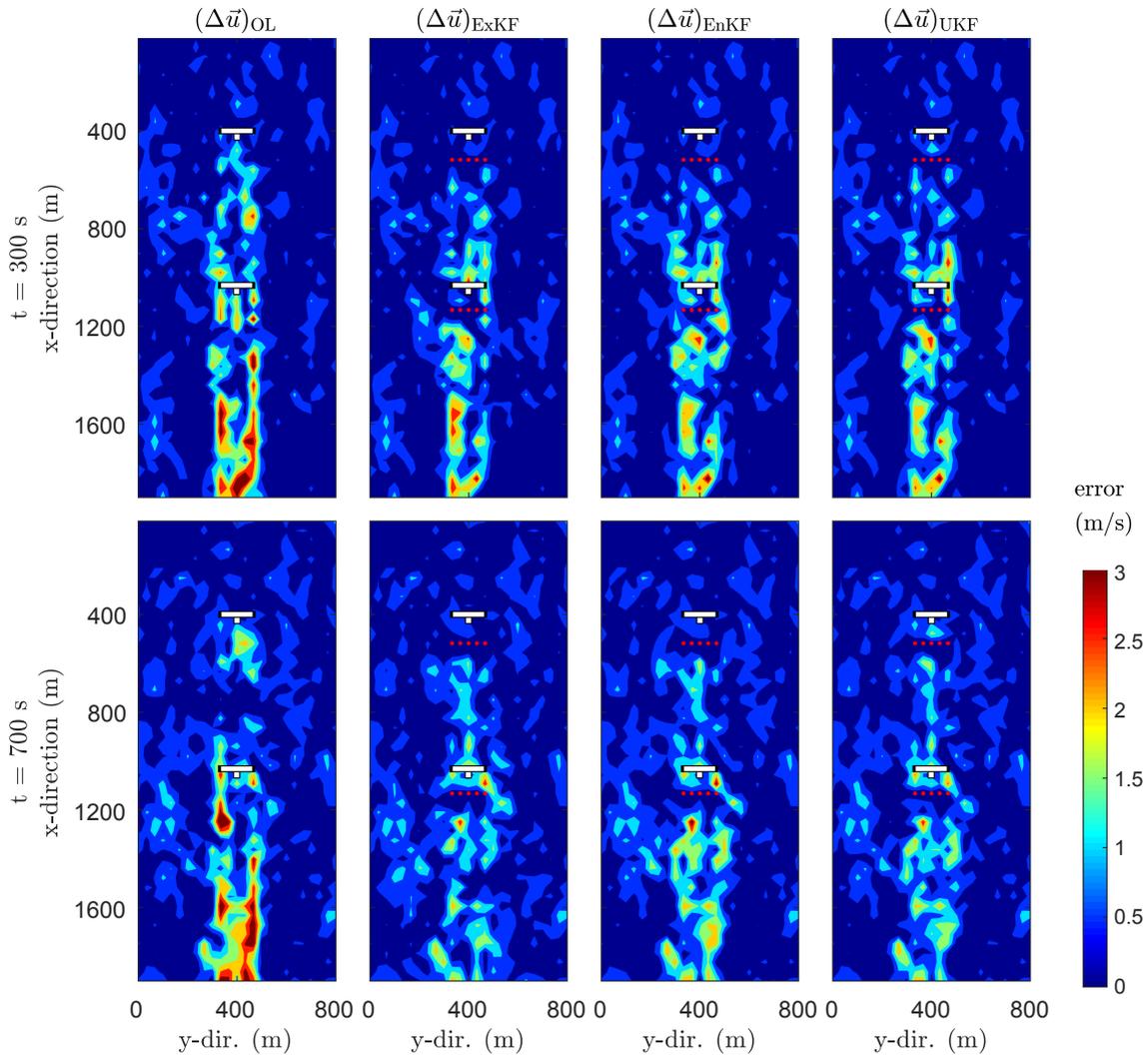


Figure 5. Comparison of absolute values of the estimation errors (in long, flow fields) for state-only estimation with the ExKF, EnKF and UKF at $t = 300$ s and $t = 700$ s, with $(\Delta \mathbf{u})_{\bullet} = |\mathbf{u}_{\bullet} - \mathbf{u}_{\text{SOWFA}}|$. The model and KF settings are depicted in Tables 1, 2, and 3. Wind is coming in from the top, flowing towards the bottom. The measured states are depicted by red dots in the flow, not to be confused with estimation error. The KFs consistently improve the instantaneous flow field estimations, noticeably near-nearby the turbine rotors, where measurements (red dots) are nearby.

The KF settings are displayed in Tables 2 and 3. In Fig. 6 it can be seen that SCADA data allows comparable performance
5 compared to the use of flow measurements, making the proposed closed-loop control solution feasible for implementation in existing wind farms, without the need for additional equipment. Furthermore, this modular framework allows the use of a combination of LiDAR systems, measurement towers, and/or SCADA data, whichever is available, for model calibration.

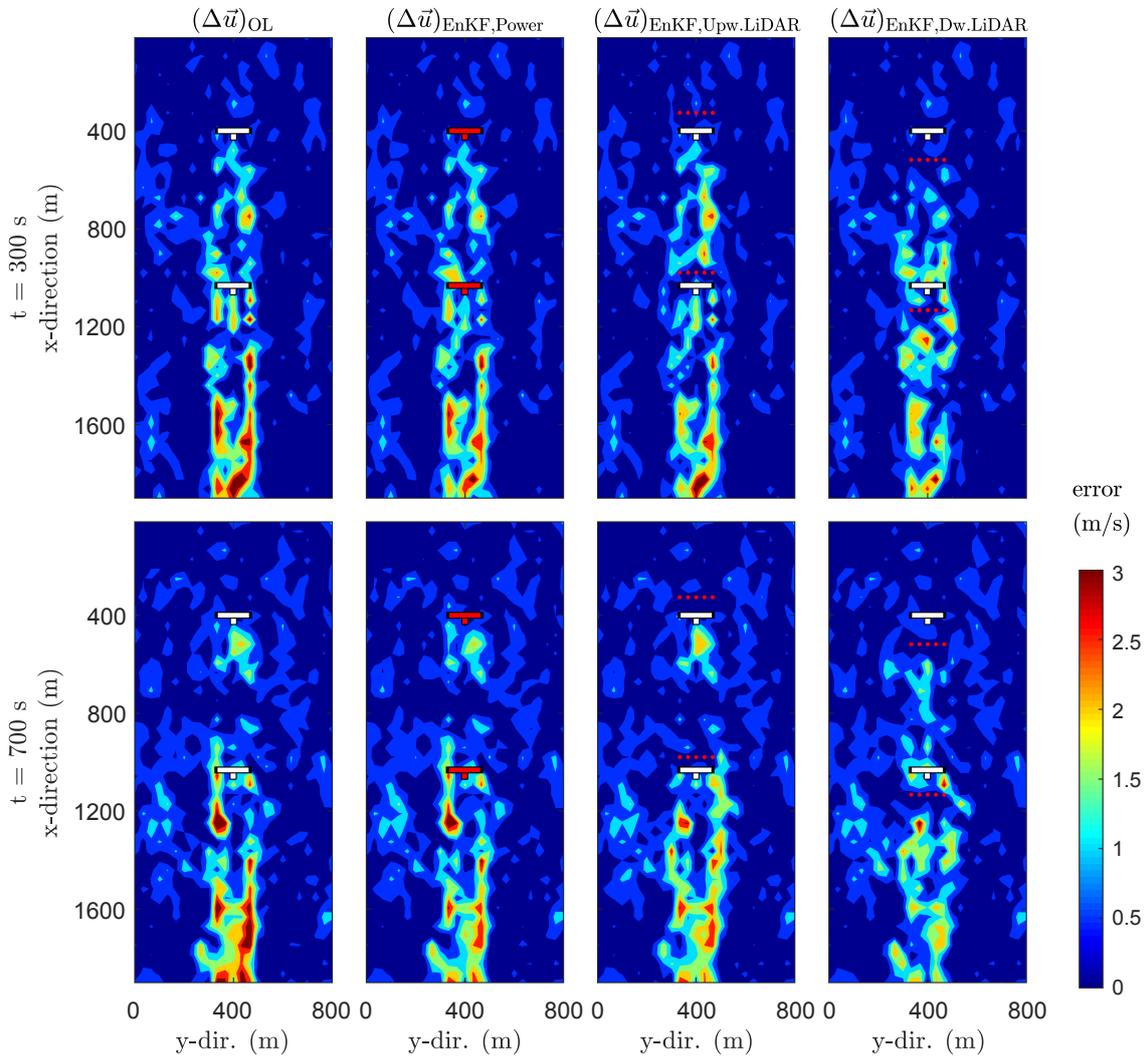


Figure 6. Comparison of absolute values of the estimation errors (in long, flow fields) for state-only estimation with the EnKF for various sensor configurations: using only-turb. power measurements(SCADA), using flow measurements with a LiDAR system pointing upstream, and using flow measurements with a LiDAR system pointing downstream of the rotor. The freestream wind Here, $(\Delta u)_\bullet = |u_\bullet - u_{SOWFA}|$. Wind is coming in from the top of the page, and flows flowing towards the bottom. Measurements-The sensors are depicted by red dots (turbine/flow dots meas.) are indicated in or red turbines (power meas.), not to be confused with estimation error.

4.2.3 State and parameter (dual) Joint state-parameter estimation

Accurate long-term forecasting demands Forecasting, as used in predictive control, benefits from the calibration of model parameters such as ℓ_s in addition to the states (flow fields). Dual model states. Joint state-parameter estimation using flow measurements downstream of each turbine disturbed by zero-mean white noise with $\sigma = 0.10$ m/s (as shown in the rightmost

plots in Fig. 6) disturbed by zero-mean white noise with $\sigma = 0.10$ m/s for the EnKF and UKF is displayed in Fig. 7, where the turbulence model tuning parameter l_s is additionally estimated. The higher l_s , the more wake recovery is modeled by WFSim. The KF settings are identical to those shown in Tables 2 and 3. From this figure, it becomes clear that the flow field estimates are not only improved for short-term forecasting the 3-minute forecast, but are also consistently better than the non-calibrated (open-loop) model's forecast for longer-term forecasting 10-minute forecast due to the real-time adaptation of the turbulence model to the actual atmospheric conditions. Furthermore, it can be seen that estimation of l_s .³ Furthermore, the EnKF performs comparably to the UKF, but at a much at a lower computational cost. Note that the EnKF even outperforms the UKF in this simulation, but this is expected to be due to randomness in the EnKF. On average, the EnKF is expected to perform similar to the UKF in terms of estimation accuracy.

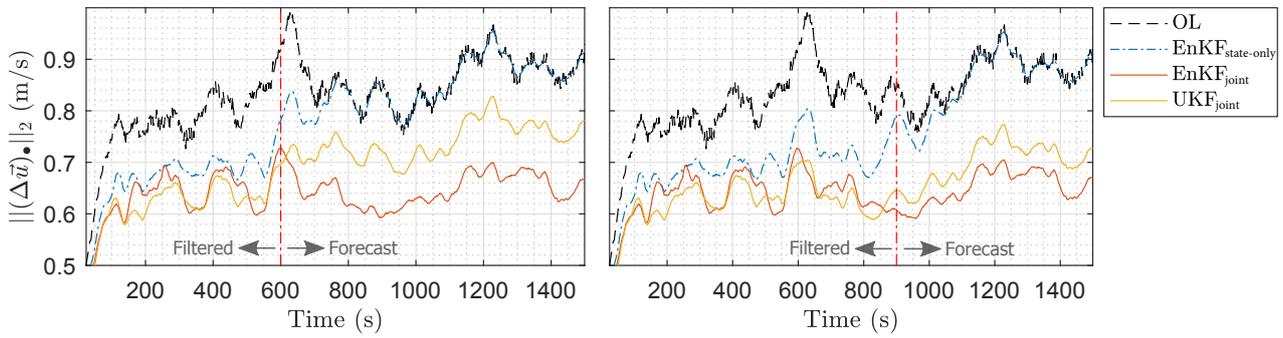


Figure 7. Comparison of forecasting performance for state-only and joint state-parameter (l_s) dual estimation with the EnKF and UKF, where measurements are available up until the vertical red-dashed lines, after which the estimation becomes a forecast. The KFs consistently improve both the short-term and long-term forecasts Here, since the turbulence model 2-norm of the estimation error is now also calibrated, plotted along the y -axis, with $(\Delta u)_{\bullet} = |u_{\bullet} - u_{SOWFA}|$.

4.3 9-turbine ALM with turbulent inflow

In this section, we investigate the performance of the EnKF-based model calibration solution under a more realistic 9-turbine wind farm scenario, which employs the EnKF for dual. The purpose of this case study is to highlight the need for state-parameter estimation for accurate wind farm modeling. The wind farm contains nine NREL 5-MW baseline turbines, oriented in a three by three layout, separated five and three rotor diameters apart in streamwise and crosswise $5D$ and $3D$ in stream- and cross-wise direction, respectively. The turbines start with a 30° yaw misalignment, but are then aligned with the mean wind direction within the first 30 s of simulation. The turbine layout and numbering is shown in the top-left subplot of Fig. 9. This LES simulation has been used before in the literature, and is described in more detail in Boersma et al. (2017b). A number of important simulation properties are listed in Table 4 for SOWFA and WFSim, respectively.

³Note that this is highly dependent on the frequency at which the freestream conditions change in the atmosphere.

Table 4. Overview of several settings for the SOWFA and the WFSim 9-turbine wind farm simulation.

Variable	Symbol	SOWFA	WFSim
Domain size	-	3.5km × 3.0km × 1.0km	1.9km × 0.80km
Number of states $N \sim \mathcal{O}(10^8)$ $\mathcal{O}(10^9)$ 1.2 · 10⁴ Cell size near rotors	-	3m × 3m × 3m	25m × 38m
Cell size outer regions	-	12m × 12m × 12m	25m × 38m
Rotor model	-	ALM	ADM ($c_f = 2.0$, $c_p = 0.97$)
Inflow wind speed	U_∞	12.03 m/s	12.00 9.0 m/s and 12.0 m/s (OL) 9.009.0 m/s (EnKF) $d' = 3.8 \cdot 10^1$ m
Atmospheric turbulence	-	$TI_\infty = 4.7\%$	$d = 5.2 \cdot 10^2$ m $\ell_s = 3.9 \cdot 10^{-2}$

~~Note that~~ Compared to the 2-turbine case, N has increased by a factor 4. In the UKF, this would result in the same factor of additional particles. Thus, not only is each particle more expensive to calculate, there are also more particles. Rather, in the EnKF, the approach is heuristic. None of the EnKF settings needed to be changed for good performance compared to Section 4.2, as displayed in Tables 2 and 3.

- 10 As shown in Table 3, the EnKF has a low computational cost of 1.2 s/iteration (8 cores, parallel). In this case study, both the complete model state (flow field), the turbulence model parameter ℓ_s , and the freestream flow speed U_∞ are estimated in real-time using exclusively readily available power measurements from the turbines. The EnKF and one of the open-loop simulations (OL) will deliberately be initialized with a poor value for ℓ_s and U_∞ to investigate convergence. The ~~performance will be compared to an other~~ open-loop simulation of WFSim will be initialized with a poor value for ℓ_s , ~~but with~~ but a correct value for U_∞ ~~for comparison.~~

In Fig. 8, it can be seen that the EnKF is successful in estimating ~~the freestream wind speed~~ U_∞ and ~~the turbulence model parameter~~ ℓ_s after about 300 s using only wind turbine power measurements. Furthermore, the flow fields of ~~the to-be-estimated model (SOWFA)~~ SOWFA, of the open-loop (OL) simulation with $U_\infty = 9.0$ m/s, and of the EnKF at various time instants are displayed in Fig. 9. From this figure, it can be seen that the EnKF has ~~very~~ large errors at the start of the simulation. However, after 10 s, the error in flow states surrounding each turbine significantly decreases through the use of ~~wind~~ turbine power measurements. This estimated flow then propagates downstream, “clearing up” the errors in the vicinity of the wind turbines. As time further propagates, the freestream estimation improves, and ~~the errors in front of the first row of turbines also reduce.~~ Finally, the turbulence model also adapts and the EnKF outperforms the open-loop simulation consistently. finally the estimation error converges.

- 10 ~~Comparison of power forecasting using the EnKF with measurements available up until time $t = 600$ s. After convergence of the freestream wind speed (as seen as a positive power slope for the first row of turbines), the turbulence model is also calibrated. After convergence, forecasting is significantly better than in open-loop. Oscillatory behavior is still present due to~~

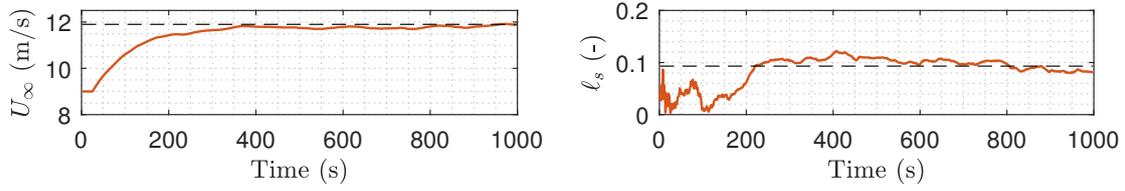


Figure 8. Convergence of ℓ_s and U_∞ using the EnKF. In dashed lines are the grid-searched optimal constant values for the open-loop simulation. With power measurements only, the model-EnKF is able to estimate these parameters successfully in addition to the model states.

~~an oscillatory input signal (C_T^i), turbulent flow field, and the absence of inertia in the rotor model. Adding rotor inertia in the surrogate model would smooth the results to better resemble true power data.~~

15 The power forecasting performance is shown in Fig. 10. ~~In this figure, the power forecast for the OL is compared to that of the EnKF, where we define the error in the time-series of the generated power of a single turbine i as $(\Delta P)_\bullet^i \in \mathbb{R}^{T_k - T_f}$ as~~

$$(\Delta P)_\bullet^i = \left[P_{k=T_f+1}^i - P_{\text{SOWFA},k=T_f+1}^i \quad P_{k=T_f+2}^i - P_{\text{SOWFA},k=T_f+2}^i \quad \cdots \quad P_{k=T_k}^i - P_{\text{SOWFA},k=T_k}^i \right]^T,$$

~~with T_k the total number of discrete simulation timesteps, and T_f the discrete timestep at which the forecast starts.~~

20 ~~As previously and Table 5. As also~~ seen in Fig. 8, the EnKF converges ~~reasonably well~~ after 300 s, and indeed the power forecasts outperform those of the OL ~~system simulation~~ at $t = 300$ s. Furthermore, it is interesting to see that the filtered power estimates of the first row of turbines ($i = 1, 2, 3$) starts low at $t = 1$ s, but converges to the true power at ~~approximately $t = 200t \approx 200$ s~~. This can be related to the mismatch in U_∞ estimates, which takes approximately ~~200–400~~300 s to converge to the true value of 12 m/s, as seen in Fig. 8. The oscillatory behavior in both the OL and EnKF power predictions is due to the absence of rotor inertia in the rotor model, turbulent structures in the flow, and large fluctuations on the excitation signal C_T^i .

Table 5. Turbine-averaged RMSE in power timeseries of Fig. 10 (compared to SOWFA). The lower the RMSE, the better the forecast.

<u>turbine row</u>	<u>OL ($U_\infty = 9.0$ m/s)</u>	<u>OL ($U_\infty = 12.0$ m/s)</u>	<u>EnKF</u>
<u>1</u>	<u>1.46 MW</u>	<u>0.19 MW</u>	<u>0.16 MW</u>
<u>2</u>	<u>1.61 MW</u>	<u>0.30 MW</u>	<u>0.18 MW</u>
<u>3</u>	<u>1.78 MW</u>	<u>0.82 MW</u>	<u>0.32 MW</u>

25 Finally, the forecasts for flow at times $t = 300$ s and $t = 600$ s are examined in Fig. 11. The large flow estimation mismatch in the EnKF at $t < 250$ s quickly reduces and for ~~$t > 250t \geq 250$ s~~ the EnKF estimation is consistently better than ~~the OL case~~both the OL cases. This has to do with the convergence of the model parameters ℓ_s and U_∞ , and the estimation of the states surrounding the turbines using the power measurements.

5 A crucial remark with the simulations presented here is that low-frequency changes in the atmosphere are neglected. In a real wind farm, atmospheric properties such as the mean wind direction and turbulence intensity change continuously, and this will impact the estimation and forecasting performance. The EnKF uses an assumption of persistence for the atmospheric

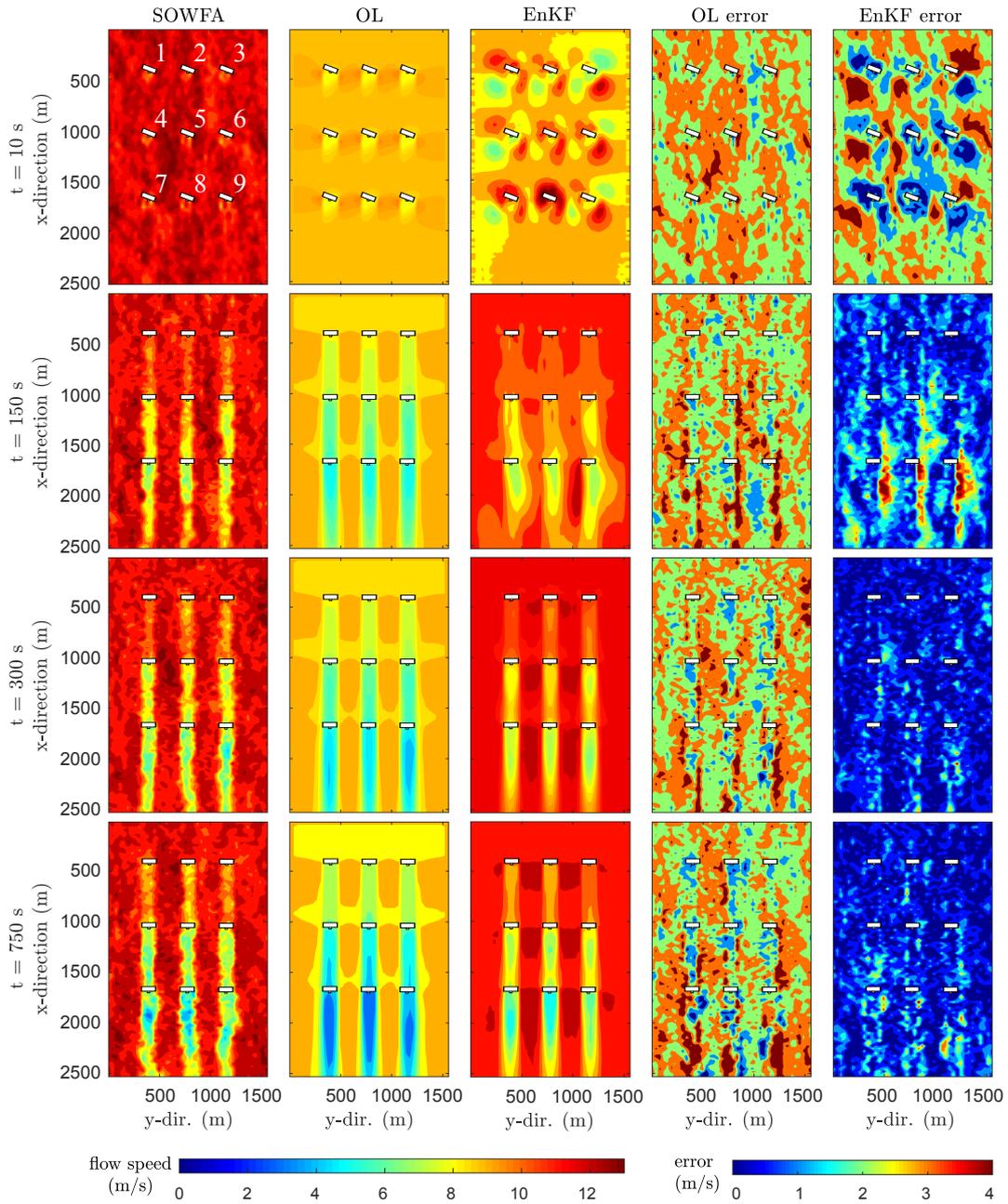


Figure 9. Comparison of absolute values of the estimation errors (in long flow fields) for state-parameter estimation with the EnKF. Wind is coming in from the top and flows downwards. The variables U_∞ and ℓ_s are incorrectly initialized in both the OL and the EnKF. In the EnKF, U_∞ and ℓ_s are estimated in addition to the states, using only turbine power measurements. The open-loop (OL) simulation is initialized with a poor ℓ_s but correct U_∞ . The EnKF quickly converges for the states, and more slowly for ℓ_s and U_∞ . After several hundreds of seconds 300 s, the EnKF has converged and consistently reconstructs the wind flow in the farm to a negligible estimation error.

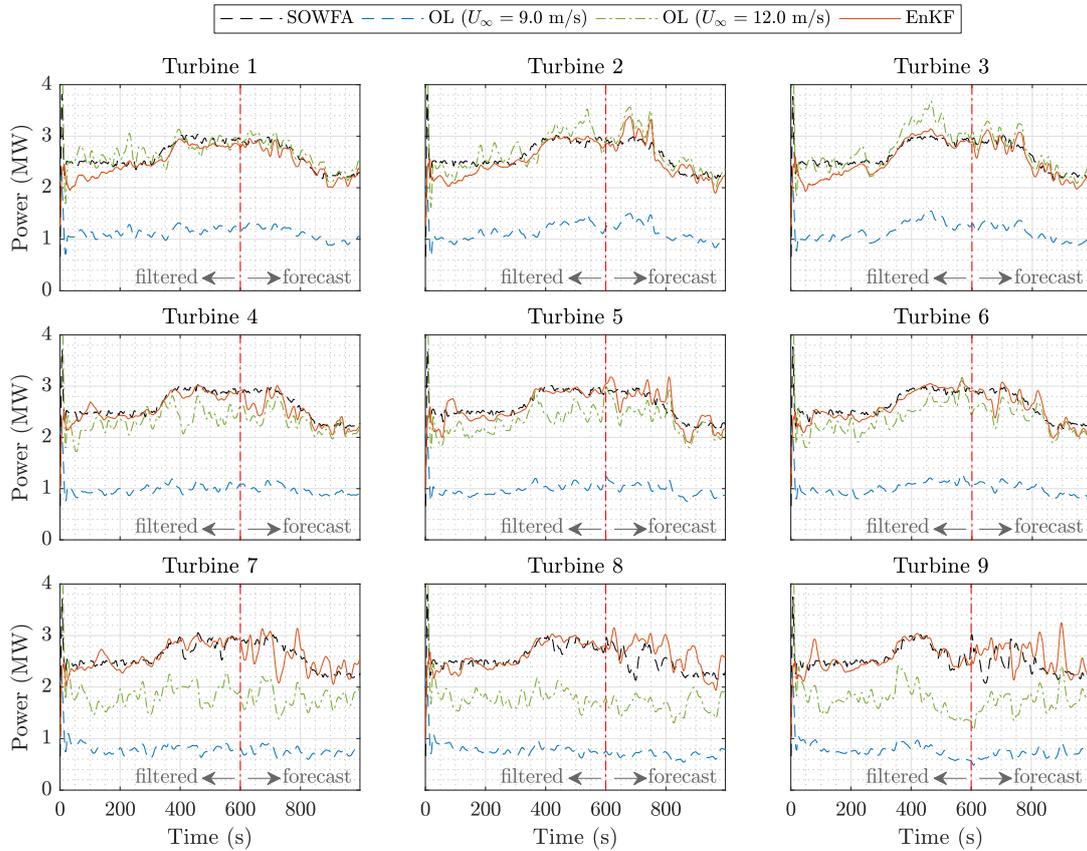


Figure 10. Comparison of power forecasting using the EnKF with measurements available up until time $t = 600$ s. After convergence U_∞ (as seen as a positive power slope for the first row of turbines), l_s is also calibrated. After convergence, forecasting is significantly better than in open-loop. Oscillatory behavior is still present due to an oscillatory input signal (C_T'), turbulent flow field, and the absence of inertia in the rotor model. Adding rotor inertia in the surrogate model would smooth the results to better resemble true power data.

properties at the time of forecasting, and thus a change in mean wind direction may invalidate the model forecast. In future work, the algorithm presented here should be tested under high-fidelity simulations with such realistic low-frequency changes. This would provide insight into the potential of the work at hand, and advance towards a practical wind farm implementation.

5 Conclusion

- This paper presented a real-time model calibration algorithm for the dynamic ~~surrogate~~-wind farm model “WFSim”, relying on an Ensemble Kalman filter (EnKF) at its core. The joint state-parameter calibration solution was tested in two ~~distinct~~ high-fidelity ~~wind farm simulations~~simulation case studies. Using exclusively SCADA measurements which are readily available in current wind farms, the adaptability to model discrepancies ~~and time-varying atmospheric conditions (namely, the~~

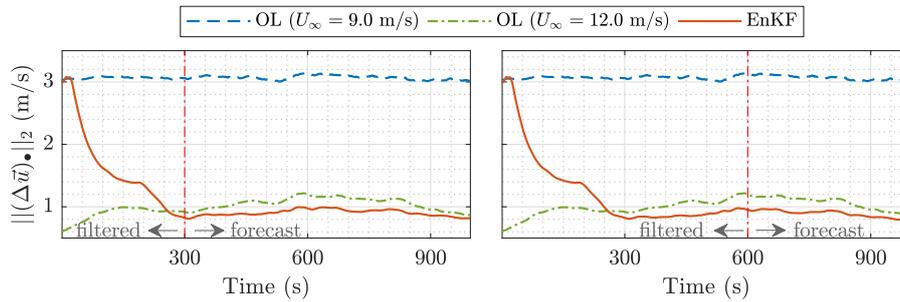


Figure 11. Comparison of flow field estimation using for the EnKF9-turbine case. On the left, measurements are available only until $t = 300$ s (left) and on the right until $t = 600$ s (right), respectively. The EnKF first has converges to converge due to the large mismatch between the estimated and true U_∞ after 300 s. After convergence, the forecasts are significantly better than in open-loop. Note that a relatively large region of (poorly observable) freestream flow is included in (Δu) , and hence the results appear suppressed compared to Fig-9.

turbulence intensity and freestream wind speed) in a 9-turbine wind farm simulation was shown, at a low computational cost of approximately 1.2 s per timestep on an 8-core CPU. Specifically, the atmospheric parameters freestream wind speed and turbulence intensity were shown to converge to their optimal values within 300 s. Furthermore, the EnKF was shown to perform comparably in terms of accuracy to the state-of-the-art algorithms in the literature, at a computational cost of multiple orders of magnitude lower. Additionally, estimation using flow measurements from LiDAR was compared to estimation using SCADA data, and it was shown that SCADA data can effectively be used for real-time model calibration. Using the proposed adaptation solution, the calibrated wind farm model can be used for accurate forecasting and optimization. In future work, the algorithm presented here should be tested under high-fidelity simulations with realistic low-frequency changes. This would provide insight into the potential of the work at hand, and advance towards a practical wind farm implementation. This work presented an essential building block for real-time closed-loop wind farm control using surrogate dynamic wind farm models.

5 Code and data availability.

The surrogate model and state-parameter estimation solutions presented in this article are open-source, available at <https://github.com/TUDELFT-DataDrivenControl/>. SOWFA is available at <https://github.com/NREL/SOWFA>. All rights for SOWFA and the simulation data presented in this work belong to the National Renewable Energy Laboratory.

Acknowledgements. The authors would like to thank Matti Morzfeld for the insightful discussions concerning the Ensemble Kalman filter. However, any mistakes in this work remain the authors' own. This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 727477. 

References

- Aho, J., Bucksapan, A., Laks, J., Fleming, P. A., Jeong, Y., Dunne, F., Churchfield, M., Pao, L. Y., and Johnson, K.: A tutorial of wind turbine control for supporting grid frequency through active power control, American Control Conference (ACC), pp. 3120–3131, 2012.
- 15 Annoni, J., Gebraad, P. M. O., Scholbrock, A. K., Fleming, P. A., and van Wingerden, J. W.: Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model, *Wind Energy*, 19, 1135–1150, 2016a.
- Annoni, J., Gebraad, P. M. O., and Seiler, P.: Wind farm flow modeling using an input-output reduced-order model, American Control Conference (ACC), pp. 506–512, 2016b.
- Boersma, S., Doekemeijer, B. M., Gebraad, P. M. O., Fleming, P. A., Annoni, J., Scholbrock, A. K., Frederik, J. A., and van Wingerden, J. W.: A tutorial on control-oriented modeling and control of wind farms, American Control Conference (ACC), pp. 1–18, 2017a.
- 20 Boersma, S., Doekemeijer, B. M., Vali, M., Meyers, J., and van Wingerden, J. W.: A control-oriented dynamic wind farm model: WFSim, *Wind Energy Science Discussions*, 2017, 1–34, 2017b.
- Churchfield, M., Wang, Q., Scholbrock, A. K., Herges, T., Mikkelsen, T., and Sjöholm, M.: Using high-fidelity computational fluid dynamics to help design a wind turbine wake measurement experiment, *Journal of Physics: Conference Series*, 753, 2016.
- 25 Churchfield, M. J., Lee, S., Michalakes, J., and Moriarty, P. J.: A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics, *Journal of Turbulence*, 13, 1–32, 2012.
- Doekemeijer, B. M., van Wingerden, J. W., Boersma, S., and Pao, L. Y.: Enhanced Kalman filtering for a 2D CFD NS wind farm flow model, *Journal of Physics: Conference Series*, 753, 2016.
- Doekemeijer, B. M., Boersma, S., Pao, L. Y., and van Wingerden, J. W.: Ensemble Kalman filtering for wind field estimation in wind farms, American Control Conference (ACC), pp. 19–24, 2017.
- 30 Ela, E., Gevorgian, V., Fleming, P. A., Zhang, Y. C., Singh, M., Muljadi, E., Scholbrock, A. K., Aho, J., Bucksapan, A., Pao, L. Y., Singhvi, V., Tuohy, A., Pourbeik, P., Brooks, D., and Bhatt, N.: Active power controls from wind power: bridging the gaps, Tech. Rep. NREL/TP-5D00-60574, National Renewable Energy Laboratory, 2014.
- Evensen, G.: The Ensemble Kalman Filter: theoretical formulation and practical implementation, *Ocean Dynamics*, 53, 343–367, 2003.
- 35 Fleming, P. A., Gebraad, P. M. O., Lee, S., van Wingerden, J. W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P.: Simulation comparison of wake mitigation control strategies for a two-turbine case, *Wind Energy*, 18, 2135–2143, 2015.
- Fleming, P. A., Aho, J., Gebraad, P. M. O., Pao, L. Y., and Zhang, Y.: Computational fluid dynamics simulation study of active power control in wind plants, American Control Conference (ACC), pp. 1413–1420, 2016.
- Fleming, P. A., Annoni, J., Scholbrock, A. K., Quon, E., Dana, S., Schreck, S., Raach, S., Haizmann, F., and Schlipf, D.: Full-scale field test of wake steering, *Journal of Physics: Conference Series*, 854, 2017a.
- 5 Fleming, P. A., 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, 229–239, 2017b.
- Gaspari, G. and Cohn, S. E.: Construction of correlation functions in two and three dimensions, *Quarterly Journal of the Royal Meteorological Society*, 125, 723–757, 1999.
- Gebraad, P., Thomas, J. J., Ning, A., Fleming, P. A., and Dykes, K.: Maximization of the annual energy production of wind power plants by optimization of layout and yaw-based wake control, *Wind Energy*, 20, 97–107, 2017.
- 10 Gebraad, P. M. O. and van Wingerden, J. W.: Maximum power-point tracking control for wind farms, *Wind Energy*, 18, 429–447, 2015.

- Gebraad, P. M. O., Fleming, P. A., and van Wingerden, J. W.: Wind turbine wake estimation and control using FLORIDyn, a control-oriented dynamic wind plant model, American Control Conference (ACC), pp. 1702–1708, 2015.
- Gebraad, P. M. O., Teeuwisse, F. W., van Wingerden, J. W., Fleming, P. A., Ruben, S. D., Marden, J. R., and Pao, L. Y.: Wind plant power optimization through yaw control using a parametric model for wake effects - a CFD simulation study, *Wind Energy*, 19, 95–114, 2016.
- Gillijns, S., Mendoza, O. B., Chandrasekar, J., De Moor, B. L. R., Bernstein, D. S., and Ridley, A.: What is the ensemble Kalman filter and how well does it work?, American Control Conference (ACC), 2006.
- Goit, J. P. and Meyers, J.: Optimal control of energy extraction in wind-farm boundary layers, *Journal of Fluid Mechanics*, 768, 5–50, 2015.
- Houtekamer, P. L. and Mitchell, H. L.: Ensemble Kalman filtering, *Quarterly Journal of the Royal Meteorological Society*, 131, 3269–3289, 2005.
- International Energy Agency: World Energy Outlook 2017, Tech. rep., International Energy Agency, 2017.
- Iungo, G. V., Santoni-Ortiz, C., Abkar, M., Porté-Agel, F., Rotea, M. A., and Leonardi, S.: Data-driven reduced order model for prediction of wind turbine wakes, *Journal of Physics: Conference Series*, 625, 2015.
- Iungo, G. V., Santhanagopalan, V., Ciri, U., Viola, F., Zhan, L., Rotea, M. A., and Leonardi, S.: Parabolic RANS solver for low-computational-cost simulations of wind turbine wakes, *Wind Energy*, 21, 184–197, <https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2154>, 2017.
- Kalman, R. E.: A new approach to linear filtering and prediction problems, *ASME Journal of Basic Engineering*, 82, 35–45, 1960.
- Knudsen, T., Bak, T., and Svenstrup, M.: Survey of wind farm control – power and fatigue optimization, *Wind Energy*, 18, 1333–1351, 2015.
- Meyers, J. and Meneveau, C.: Large Eddy Simulations of Large Wind-Turbine Arrays in the Atmospheric Boundary Layer, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, 2010.
- Mittelmeier, N., Blodau, T., and Kühn, M.: Monitoring offshore wind farm power performance with SCADA data and an advanced wake model, *Wind Energy Science*, 2, 175–187, 2017.
- Munters, W. and Meyers, J.: An optimal control framework for dynamic induction control of wind farms and their interaction with the atmospheric boundary layer, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 375, 2017.
- Petrie, R. E.: Localization in the Ensemble Kalman filter, Master’s dissertation, University of Reading, 2008.
- Rotea, M. A.: Dynamic programming framework for wind power maximization, *IFAC Proceedings Volumes*, 47, 3639 – 3644, 19th IFAC World Congress, 2014.
- Schreiber, J., Nanos, E. M., Campagnolo, F., and Bottasso, C. L.: Verification and calibration of a reduced order wind farm model by wind tunnel experiments, *Journal of Physics: Conference Series*, 854, 2017.
- Shapiro, C. R., Bauweraerts, P., Meyers, J., Meneveau, C., and Gayme, D. F.: Model-based receding horizon control of wind farms for secondary frequency regulation, *Wind Energy*, 20, 1261–1275, we.2093, 2017a.
- Shapiro, C. R., Meyers, J., Meneveau, C., and Gayme, D. F.: Dynamic wake modeling and state estimation for improved model-based receding horizon control of wind farms, American Control Conference (ACC), pp. 709–716, 2017b.
- Simley, E. and Pao, L. Y.: Evaluation of a wind speed estimator for effective hub-height and shear components, *Wind Energy*, 19, 167–184, 2016.
- Siniscalchi-Minna, S., Bianchi, F. D., and Ocampo-Martinez, C.: Predictive control of wind farms based on lexicographic minimizers for power reserve maximization, American Control Conference (ACC), 2018.
- Sorensen, J. N. and Shen, W. Z.: Numerical modeling of wind turbine wakes, *Journal of Fluid Mechanics*, 124, 393–399, 2002.

- Thomas, J. J., Gebraad, P. M. O., and Ning, A.: Improving the FLORIS wind plant model for compatibility with gradient-based optimization, *Wind Engineering*, 41, 313–329, 2017.
- 665 Vali, M., Van Wingerden, J. W., Boersma, S., Petrović, V., and Kühn, M.: A predictive control framework for optimal energy extraction of wind farms, *Journal of Physics: Conference Series*, 753, 2016.
- Vali, M., Petrović, V., Boersma, S., Van Wingerden, J. W., and Kühn, M.: Adjoint-based model predictive control of wind farms: Beyond the quasi steady-state power maximization, *IFAC World Congress*, 50, 4510 – 4515, 2017.
- Van Wingerden, J. W., Pao, L. Y., Aho, J., and Fleming, P. A.: Active power control of waked wind farms, *IFAC World Congress*, pp. 4570–4577, 2017.
- 670 Verhaegen, M. and Verdult, V.: *Filtering and System Identification: A least squares approach*, Cambridge University Press, 2007.
- Wan, E. A. and Van Der Merwe, R.: The unscented Kalman filter for nonlinear estimation, *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.