Dear reviewer,

We appreciate your valuable feedback. This further improves the scientific quality and relevance of the article. Find attached our responses, and a manuscript showing the changes we made accordingly.

Best regards,
Bart Doekemeijer

- *RC1: Thanks for your detailed answers to my points.*
  *I still think the paper is good in general, but way too long for its actual contribution.*
  Thank you for your positive assessment. There is always a trade-off between clarity of the presentation and the length of a paper.

- *RC2: I suggested some parts to be removed that are known or available in other publications, and to just refer to them. Your point that the audience should not necessarily know what a KF is valid, but not convincing to retain the KF definitions (the audience would anyway not learn KF from your paper). The model also is presented elsewhere.*
  Dear reviewer. We appreciate your feedback on the paper, and we shortened the paper. The chapter on the surrogate model has significantly been shortened. Furthermore, the part on linearized Kalman filtering and the equations of the Extended Kalman filter have been omitted in the revised version. This further reduces the paper by 3 pages, in addition to the 3-page reduction in the last revision. We sincerely hope that you will consider this a sufficient reduction.

- *RC3: The open loop prediction by the KFs is, as far as I am concerned, also of little value. With or without parameter estimation included, mathematically, I don't see why an (extended) KF should necessarily perform better than an OL model when there are no measurements (provided that the OL model is not unstable, which should be the case). In the paper you state that in Fig 7 for predictions after the dashed line no measurements are used by the models, while in your response to my comment you write "...we are feeding in noisy measurements..." as argument for the better performance, which is rather embarrassing?!? I would recommend to completely remove Fig 7, unless the prediction is made under different wind conditions than during the closed-loop operation.*
  Dear reviewer. We apologize for causing any possible confusion. For the simulations with joint state-parameter estimation, the open-loop (OL) model and the Kalman filters are started with the same initial value for $l_s$. Then, every second, (noisy) measurements are fed into the KFs, and the state vector as well as the model parameter $l_s$ are estimated. However, for the open-loop simulation, no measurements are fed in: the state vector is simply updated with the nominal model, and the value for $l_s$ remains the same.
  Now, after 600 s (left plot in Fig. 6) and 900 s (right plot in Fig. 6), we start a forecast, meaning no measurements are available after this time. At this moment, the OL model still has the same (poor) value for $l_s$ as at t=0 s, but the value for $l_s$ in the EnKF is different (better, hopefully). This leads to the OL and joint KF forecasts to differ from each other. Noticeably, in terms of control, $l_s$ is assumed to be a random walk model when estimated with the EnKF, essentially putting a pole at zero (cont. time), making the system marginally stable. In that case, the OL and KF solutions do not necessarily converge anymore. This is indeed what we see in Fig. 6. For state-only estimation, we have a stable system, and the forecasts converge within 100-200 seconds. For joint state-parameter estimation, we have a marginally stable system, and the forecasts do not converge. Actually, since the EnKF-joint and UKF-joint algorithms do their forecast with more accurate values for $l_s$, their forecasts have lower errors.

  For clarification, I have added a more elaborate explanation of the effects of parameter estimation in Section 4.2.3.

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

Bart Doekemeijer[1], Sjoerd Boersma[1], Lucy Pao[2], Torben Knudsen[3], and Jan-Willem van Wingerden[1]

[1]Delft Center for Systems and Control, Delft University of Technology, The Netherlands
[2]Electrical, Computer & Energy Engineering, University of Colorado Boulder, United States of America
[3]Department 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 benefitsin the growth of the wind energy industry~~these developments have clear benefits, an important problem with wind energy is that the rotational speed of ~~almost all~~ most commercial turbines is ~~currently~~ decoupled from the electricity grid frequency via each turbine's power electronics (Aho et al., 2012). As the current grid-connected fossil fuel plants are replaced by non-synchronous renewable energy plants, the inertia of the electricity grid will decrease~~. Thus, the grid will become less stable, making it~~ , making it less stable and 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 mainly focused on maximizing the power capture (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 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 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, consisting of three components, is shown in Fig. 1.

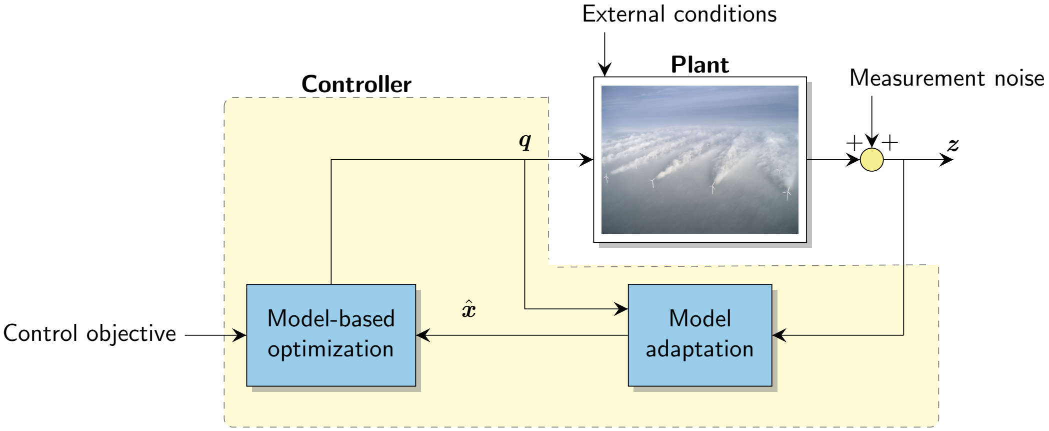

**Figure 1.** Closed-loop wind farm control framework. Measurements $z$ (e.g., SCADA~~, met mast,~~ or LiDAR data) are fed into the controller. First, the state of the surrogate ~~wind farm~~ model $x$ is estimated to represent the actual atmospheric and turbine conditions inside the wind farm. Secondly, 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 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). 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 steady state, limiting their use for APC. 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 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 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 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.

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 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 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, Iungo 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 computational tractability.

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) 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.

The focus of this work is on a model adaptation algorithm for WFSim, which trades off estimation accuracy with computational complexity. In previous work (Doekemeijer et al., 2016, 2017), 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~~:

- the additional adaptation to a mismatch in atmospheric conditions (specifically, the ~~freestream~~ ambient wind speed and turbulence~~intensity), 2~~),
- the option to use turbine's power signals in addition to, or instead of, flow measurements, ~~3)~~
- a further reduction in the computational complexity, ~~and 4)~~
- and a comparison of the EnKF with the state of the art in the literature.

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

## 2 The surrogate model

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). ~~WFSim has shown success in reconstructing the flow field and turbine power signals of high-fidelity simulation data (Boersma et al., 2017b; Doekemeijer et al., 2017).~~ This model is particularly suited ~~for the framework presented in Fig. 1~~ as it includes both yaw and axial induction actuation and yields a relatively high accuracy with a relatively low computational cost.[1]

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 ??, the governing equations of the model are presented. The turbulence and turbine model are described in Sections ?? and ??, respectively. The spatial and temporal discretization process is described in Section ??.~~

### 2.1 ~~Governing equations~~

~~The WFSim model~~ Fundamentally, WFSim is based on the two-dimensional unsteady incompressible Navier-Stokes (NS) equations~~to maintain computational tractability compared to a three-dimensional model.~~. 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:~~

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

~~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, $\boldsymbol{\tau}_H$ are the subgrid stresses (turbulence model), $p$ is pressure, and $\mathbf{f}$ contains the forcing terms (turbine model) acting on the flow. Equation deviates from the~~. WFSim deviates from a traditional 2D NS ~~equations~~ model 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~~ dissipation term in the lateral direction in the continuity equation is multiplied

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

by a factor 2 to approximate flow dissipating in the vertical flow dimension. Other vertical flow contributions such as vertical meandering and shear are neglected.

## 2.1 ~~Subgrid-scale (turbulence) model~~

The subgrid-scale model is formulated using an eddy-viscosity assumption in combination with Prandtl's mixing length model. The mixing length is parametrized as a function of the spatial location, increasing linearly with distance from the downstream rotor, starting at zero at distance $d'$ downstream and peaking at distance $d$,

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

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~~

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

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

$$\ell_u^i(x_i', y_i') = \begin{cases} (x_i' - d')\ell_s, & \text{if } d' \le x_i' \le d \text{ and } -\frac{D}{2} \le y_i' \le \frac{D}{2}, \\ 0, & \text{otherwise,} \end{cases}$$

~~where~~ $\ell_s$ 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 $\ell_s$ is a gradient parameter for the~~ the mixing length.

~~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).~~

## 2.1 ~~Turbine model~~

~~Turbine forces in WFSim~~ Basically, the larger $\ell_s$, the quicker wakes recover to their freestream properties. Furthermore, the turbines are modeled using the ~~classical~~ non-rotating (static) actuator disk model, 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~~ Both the turbine forcing term ~~in , $\mathbf{f}$, at spatial location $\boldsymbol{s} = \begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^2$ is expressed as~~

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

~~with $\mathrm{H}[\bullet]$ the heaviside function, $D$ the rotor diameter, $\phi$ the mean wind direction on the rotor plane, $\delta[\bullet]$ the Dirac delta function, and $\boldsymbol{e}_{\perp,i} \in \mathbb{R}^2$ the unit vector perpendicular to the $i^{\text{th}}$ rotor disk with position $\boldsymbol{t}_i \in \mathbb{R}^2$. The scalar $C'_{T_i}$ is the thrust coefficient of turbine $i$ defined according to Meyers and Meneveau (2010). The scalar $\gamma_i$ is the yaw misalignment angle of turbine $i$, and $U_i$ is the average flow velocity over the rotor. The scalar $c_f$ is a static tuning variable 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 $\gamma_i$~~ and ~~$C'_{T_i}$ for $i = 1, ..., N_T$, with $N_T$ the number of turbines. Furthermore, the instantaneous power capture of the wind farm $P_{\text{farm}}$ is calculated in a similar approach by~~

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

~~with scalar $c_p$ the fifth tuning factor used~~ the turbine power output are scaled by tuning factors $c_f$ and $c_p$, respectively, to account for ~~numerical grid effectsand 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 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).~~ unmodeled effects. Together with the three parameters from the turbulence model, this leads to a total of five tuning parameters.

### 2.1 ~~Discretization, boundary conditions, and computational cost~~

~~Equation 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.~~ These NS equations are solved over a spatially and temporally discretized domain (Boersma et al., 2017b). Dirichlet boundary conditions for ~~$u$ and $v$~~ the longitudinal and lateral velocity 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~~ The surrogate model reduces to a nonlinear discrete-time deterministic state-space model, ~~described by~~ as

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

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

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

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 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, $\boldsymbol{q}_k \in \mathbb{R}^O$ includes the system inputs, i.e., the turbine control settings: the turbine yaw angles $\gamma_i$ and the thrust coefficients $C'_{T_i}$ for $i=1,...,N_T$, with $N_T$ the number of turbines. The system outputs $\boldsymbol{z}_k \in \mathbb{R}^M$ are defined by sensors. It can include, among others, flow field measurements ($\boldsymbol{z}_k \subset \boldsymbol{x}_k$) and power measurements. We define the integer $M_{u,v} \in \mathbb{Z}$ with $0 \leq M_{u,v} \leq M$ 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 two-turbine wind ~~farm~~ farms with $N = 3 \cdot 10^3$ states (e.g., in Doekemeijer et al. (2017)), to $1.2$ s for $N = 1 \cdot 10^5$ states for medium-sized wind farms (e.g., in Boersma et al. (2017b)), for a single time-step forward simulation on a single desktop CPU core. The computational complexity of the model is what motivates the use of time-efficient estimation algorithms in this work, and time-efficient predictive control methods for optimization in related work (Vali et al., 2017). Here, the limits of computational cost are explored to maximize model accuracy while still allowing real-time control. 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 nonlinear state estimation algorithms are presented in Sections ~~??~~ 3.3 to 3.5, 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 3.6, 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 $\boldsymbol{x}$ as *state(-only) estimation*. The estimation of both model states and model parameters such as $\ell_s$ is referred to as *(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, which limits accuracy (Boersma et al., 2017b). Secondly, an estimation solution relying on WFSim is sensitive to instability when the ~~estimated~~ model state sufficiently deviates from the continuity equation~~in~~. 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 control, and thus one needs to find a trade-off between accuracy ~~on the one hand, while guaranteeing~~ while guaranteeing state updates at a low computational cost~~on the other hand~~.

### 3.2 General formulation

This section ~~details~~ summarizes the basics of the Kalman Filter (KF), which is the literature standard for state estimation in control. The goal of a 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,

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{q}_k) + \boldsymbol{w}_k, \tag{1}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{q}_k) + \boldsymbol{v}_k, \tag{2}$$

where $k$ is the time index, $\boldsymbol{x} \in \mathbb{R}^N$ is the unobserved system state, $\boldsymbol{z} \in \mathbb{R}^M$ are the measured outputs of the system, $\boldsymbol{q} \in \mathbb{R}^O$ and $\boldsymbol{w} \in \mathbb{R}^N$ are the controllable inputs and process noise respectively that drive the system dynamics, and $\boldsymbol{v} \in \mathbb{R}^M$ is measurement noise. Furthermore, we assume $\boldsymbol{w}$ and $\boldsymbol{v}$ to be zero-mean white Gaussian noise with covariance matrices

$$\mathrm{E}\left[\begin{bmatrix}\boldsymbol{v}_k \\ \boldsymbol{w}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_\ell^T & \boldsymbol{w}_\ell^T\end{bmatrix}\right] = \begin{bmatrix}\boldsymbol{R}_k & \boldsymbol{S}_k^T \\ \boldsymbol{S}_k & \boldsymbol{Q}_k\end{bmatrix}\Delta_{k-\ell}, \quad \text{where} \quad \Delta_{k-\ell} = \begin{cases}1, & \text{if } k = \ell, \\ 0, & \text{otherwise,}\end{cases} \tag{3}$$

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

$$\hat{\boldsymbol{x}}_{k|\ell} = \mathrm{E}\left[\boldsymbol{x}_k | \boldsymbol{\mathcal{Z}}_\ell\right], \quad \text{with} \quad \boldsymbol{\mathcal{Z}}_\ell = \boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2 \ldots \boldsymbol{z}_\ell, \quad \boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2 \ldots \boldsymbol{q}_\ell. \tag{4}$$

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

In reality, the assumed model described by $f$ and $h$ always has mismatches with the true system, and the assumptions in (3) often do not hold. Further, the matrices $\boldsymbol{Q}_k$, $\boldsymbol{R}_k$, and $\boldsymbol{S}_k$ are usually not known and rather considered tuning parameters, used to shift the confidence levels between the internal model and the measured values. For $\boldsymbol{R} \ll \boldsymbol{Q}$, estimations will heavily rely on the measurements, while for $\boldsymbol{Q} \ll \boldsymbol{R}$, estimations will mostly rely on the internal model. Kalman filtering remains one of the most common methods of recursive state estimation. KF algorithms typically consist of two steps, namely:

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

$$\hat{\boldsymbol{x}}_{k|k-1} = \mathrm{E}\left[f(\boldsymbol{x}_{k-1}, \boldsymbol{q}_{k-1}) + \boldsymbol{w}_{k-1}|\boldsymbol{\mathcal{Z}}_{k-1}\right], \tag{5}$$

$$\hat{\boldsymbol{z}}_{k|k-1} = \mathrm{E}\left[h(\boldsymbol{x}_{k}, \boldsymbol{q}_{k}) + \boldsymbol{v}_{k}|\boldsymbol{\mathcal{Z}}_{k-1}\right], \tag{6}$$

$$\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1} = \mathrm{Cov}\left(\boldsymbol{x}_k, \boldsymbol{x}_k|\boldsymbol{\mathcal{Z}}_{k-1}\right) = \mathrm{E}[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})^T], \tag{7}$$

$$\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1} = \mathrm{Cov}\left(\boldsymbol{z}_k, \boldsymbol{z}_k|\boldsymbol{\mathcal{Z}}_{k-1}\right) = \mathrm{E}[(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})^T], \tag{8}$$

$$\boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1} = \mathrm{Cov}\left(\boldsymbol{x}_k, \boldsymbol{z}_k|\boldsymbol{\mathcal{Z}}_{k-1}\right) = \mathrm{E}[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})^T]. \tag{9}$$

In (5) and (6), $\hat{\boldsymbol{x}}_{k|\ell}$ and $\hat{\boldsymbol{z}}_{k|\ell}$ 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:

$$\boldsymbol{L}_k = \boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1} \cdot \left(\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1}\right)^{-1} \tag{10}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k\left(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}\right), \tag{11}$$

$$\boldsymbol{P}^{\boldsymbol{x}}_{k|k} = \mathrm{Cov}\left(\boldsymbol{x}_k, \boldsymbol{x}_k|\boldsymbol{\mathcal{Z}}_{k}\right) = \boldsymbol{P}^{\boldsymbol{x}}_{k|k-1} - \boldsymbol{L}_k\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1}\boldsymbol{L}_k^T. \tag{12}$$

Here, $\left(\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1}\right)^{-1}$ in (10) is the pseudo-inverse of $\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1}$, since this matrix is not necessarily invertible.

### 3.3 ~~Linear Kalman filter~~

Traditionally, state estimation for linear dynamic models is done using the linear KF (Kalman, 1960). ~~In the idealized situation where the following three criteria hold: 1) the assumptions on noise in are correct, 2)~~ However, this is not a viable option here, as the surrogate model ~~$f$ and $h$ perfectly match reality, and 3) $f$ and $h$ are linear in $\boldsymbol{x}$ and $\boldsymbol{q}$, with~~

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

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

~~where $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, $\boldsymbol{C}_k$, $\boldsymbol{D}_k$ 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, $\mathrm{E}(\boldsymbol{x}_k) = \hat{\boldsymbol{x}}_k$, with minimal mean-square error (the trace of $\boldsymbol{P}^{\boldsymbol{x}}_{k|k}$ is minimized). For to , one can derive that in the linear case,~~

~~$\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}_{k-1}\boldsymbol{P}^{\boldsymbol{x}}_{k-1|k-1}\boldsymbol{A}^T_{k-1} + \boldsymbol{Q}_{k-1},$~~

~~$\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1} = \boldsymbol{C}_k\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\boldsymbol{C}^T_k + \boldsymbol{R}_k,$~~

~~$\boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1} = \boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\boldsymbol{C}^T_k + \boldsymbol{S}_k.$~~

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

### 3.3 Extended Kalman filter (ExKF)

Linearization of the surrogate model is the most popular and straight-forward solution to the issue of model nonlinearity~~. In~~ , as done in the Extended KF (ExKF)~~, the surrogate model is linearized around some point ($\boldsymbol{x}^{\text{lin}}, \boldsymbol{q}^{\text{lin}}$) w. r.t. $\boldsymbol{x}$ and $\boldsymbol{q}$ at every timestep $k$:~~

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

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

~~Using the linearized system matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, $\boldsymbol{C}_k$, $\boldsymbol{D}_k$, one can directly apply to for state estimation, where to 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~~ . The ExKF has shown success in academia and industry (Wan and Van Der Merwe, 2000) ~~.~~

and is perhaps the most popular nonlinear KF. However, it has a number of disadvantages. 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 (10) includes a matrix inversion with a computational complexity of $\mathcal{O}(M^3)$, and the ExKF furthermore 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, ~~dominates~~ these matrix multiplications dominate the computational cost. The ExKF has a CPU time in the order of $10^1$ s for a two-turbine wind farm, which may be too large for ~~online model calibration if APC is the objective.~~ our purposes. 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.

### 3.4 The Unscented Kalman filter (UKF)

The Unscented Kalman filter (UKF) relies on the so-called “unscented transformation” to estimate the means and covariance matrices described by (5) to (9). The conditional state probability distribution of $\boldsymbol{x}_k$ knowing $\boldsymbol{\mathcal{Z}}_k$ 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 $\hat{\boldsymbol{x}}_{k|k}$ and their covariance is equal to $\text{Cov}\,(\boldsymbol{x}_k, \boldsymbol{x}_k)$. 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 $i^{\text{th}}$ particle as $\boldsymbol{\psi}^i_{k|\ell} \in \mathbb{R}^N$, which is a realization of the conditional probability distribution of $\boldsymbol{x}_k$ given $\boldsymbol{\mathcal{Z}}_\ell$. The UKF follows a very similar forecast and analysis update approach as the traditional KF in (5) to (12), yet applied to a finite set of particles (Wan and Van Der Merwe, 2000).

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 $p\left[\boldsymbol{x}_{k-1}|\boldsymbol{\mathcal{Z}}_{k-1}\right]$, by

$$\boldsymbol{\psi}^i_{k-1|k-1} = \begin{cases} \overline{\boldsymbol{\psi}}_{k-1|k-1} & \text{for } i = 1, \\ \overline{\boldsymbol{\psi}}_{k-1|k-1} + \left(\sqrt{(N+\lambda)\cdot \boldsymbol{P}^{\boldsymbol{x}}_{k-1|k-1}}\right)_i & \text{for } i = 2, ..., N+1, \\ \overline{\boldsymbol{\psi}}_{k-1|k-1} - \left(\sqrt{(N+\lambda)\cdot \boldsymbol{P}^{\boldsymbol{x}}_{k-1|k-1}}\right)_{i-N-1} & \text{for } i = N+2, ..., Y, \end{cases} \tag{13}$$

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

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

   and $\beta$ 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

$$\begin{aligned} \boldsymbol{\psi}^i_{k|k-1} &= f(\boldsymbol{\psi}^i_{k-1|k-1}, \boldsymbol{q}_{k-1}) && \text{for} \quad i = 1, ..., Y, \\ \boldsymbol{\zeta}^i_{k|k-1} &= h(\boldsymbol{\psi}^i_{k|k-1}, \boldsymbol{q}_k) && \text{for} \quad i = 1, ..., Y, \end{aligned} \tag{14}$$

   where $\boldsymbol{\zeta}^i_{k|\ell}$ is defined as the system output corresponding to the particle $\boldsymbol{\psi}^i_{k|\ell}$.

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

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1} = \overline{\boldsymbol{\psi}}_{k|k-1} &= \sum_{i=1}^{Y}\left(\boldsymbol{w}^i_{\text{mean}} \cdot \boldsymbol{\psi}^i_{k|k-1}\right), \\ \hat{\boldsymbol{z}}_{k|k-1} = \overline{\boldsymbol{\zeta}}_{k|k-1} &= \sum_{i=1}^{Y}\left(\boldsymbol{w}^i_{\text{mean}} \cdot \boldsymbol{\zeta}^i_{k|k-1}\right), \end{aligned} \tag{15}$$

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

$$\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1} = \sum_{i=1}^{Y}\left(\boldsymbol{w}^i_{\text{cov.}}\left(\boldsymbol{\psi}^i_{k|k-1} - \overline{\boldsymbol{\psi}}_{k|k-1}\right)\left(\boldsymbol{\psi}^i_{k|k-1} - \overline{\boldsymbol{\psi}}_{k|k-1}\right)^T\right) + \boldsymbol{Q}_{k-1}, \tag{16}$$

$$\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1} = \sum_{i=1}^{Y} \left( \boldsymbol{w}^{i}_{\text{cov.}} \left( \boldsymbol{\zeta}^{i}_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1} \right) \left( \boldsymbol{\zeta}^{i}_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1} \right)^{T} \right) + \boldsymbol{R}_{k}, \tag{17}$$

$$\boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1} = \sum_{i=1}^{Y} \left( \boldsymbol{w}^{i}_{\text{cov.}} \left( \boldsymbol{\psi}^{i}_{k|k-1} - \overline{\boldsymbol{\psi}}_{k|k-1} \right) \left( \boldsymbol{\zeta}^{i}_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1} \right)^{T} \right) + \boldsymbol{S}_{k}. \tag{18}$$

2. For the analysis step, one can apply the same equations as in (10) to (12).

The UKF has been shown to consistently outperform the ExKF in terms of accuracy, since it uses the nonlinear model for forecasting and covariance propagation. However, this does come at 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 2-turbine wind farm in WFSim, a computational cost of $1 \cdot 10^{2}$ s per iteration ($k \to k+1$) would not be surprising. While (14) can easily be parallelized, computational complexity remains troublesome, especially for larger wind farms. The issue of computational complexity is tackled by the Ensemble KF.

### 3.5 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 $\boldsymbol{x}_k$ knowing $\boldsymbol{\mathcal{Z}}_k$. However, whereas the UKF relies on a systematic way of distributing the particles such that the mean and covariance of the conditional probability distribution $p\,[\boldsymbol{x}_k|\boldsymbol{\mathcal{Z}}_k]$ are equal to that of the particles, the EnKF relies 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 $i^{\text{th}}$ particle as $\boldsymbol{\psi}^{i}_{k|\ell} \in \mathbb{R}^{N}$, which is a realization of the conditional probability distribution $p\,[\boldsymbol{x}_k|\boldsymbol{\mathcal{Z}}_\ell]$. The forecast step is:

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

$$\begin{aligned} \boldsymbol{\psi}^{i}_{k|k-1} &= f(\boldsymbol{\psi}^{i}_{k-1|k-1}, \boldsymbol{q}_{k-1}) + \hat{\boldsymbol{w}}^{i}_{k-1} && \text{for} \quad i = 1, ..., Y, \\ \boldsymbol{\zeta}^{i}_{k|k-1} &= h(\boldsymbol{\psi}^{i}_{k|k-1}, \boldsymbol{q}_{k}) + \hat{\boldsymbol{v}}^{i}_{k} && \text{for} \quad i = 1, ..., Y. \end{aligned} \tag{19}$$

   (ii) The expected state and output are calculated identically as in the UKF using (15) with $\boldsymbol{w}^{i}_{\text{mean}} = (Y-1)^{-1}$. The covariance matrices are (re-)estimated from the forecasted ensemble, by

$$\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1} = \frac{1}{Y-1} \sum_{i=1}^{Y} \left( \left( \boldsymbol{\zeta}^{i}_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1} \right) \left( \boldsymbol{\zeta}^{i}_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1} \right)^{T} \right), \tag{20}$$

$$\boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1} = \frac{1}{Y-1} \sum_{i=1}^{Y} \left( \left( \boldsymbol{\zeta}^{i}_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1} \right) \left( \boldsymbol{\psi}^{i}_{k|k-1} - \overline{\boldsymbol{\psi}}_{k|k-1} \right)^{T} \right). \tag{21}$$

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

$$\boldsymbol{\psi}^i_{k|k} = \boldsymbol{\psi}^i_{k|k-1} + \boldsymbol{L}_k \left( \boldsymbol{z}_k - \boldsymbol{\zeta}^i_{k|k-1} \right) \qquad \text{for} \quad i = 1, ..., Y. \tag{22}$$

Note that, in contrast to the ExKF and the UKF, the state covariance matrix $\boldsymbol{P}^{\boldsymbol{x}}$ (see (7) and (12)) 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.

#### 3.5.1 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 $\boldsymbol{P}^{\boldsymbol{x}}$ 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 $\boldsymbol{P}^{\boldsymbol{x}}$ (or rather, the ensemble from which $\boldsymbol{P}^{\boldsymbol{x}}$ is calculated) is “inflated” to correct for the underestimated state uncertainty (Petrie, 2008). Mathematically, this is achieved by applying

$$\boldsymbol{\psi}^i_{k|k-1} = \overline{\boldsymbol{\psi}}_{k|k-1} + r \left( \boldsymbol{\psi}^i_{k|k-1} - \overline{\boldsymbol{\psi}}_{k|k-1} \right) \qquad \text{for} \quad i = 1, ..., Y, \tag{23}$$

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. 2. In particle-based approaches, the covariance terms cannot be captured exactly. This may lead to the formation of small yet nonzero covariance terms between

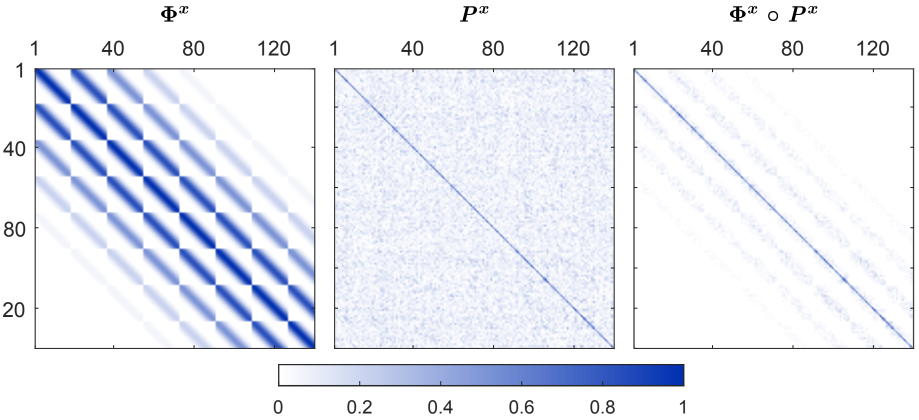

**Figure 2.** Long-range spurious correlations arise in the case where a covariance matrix is described by a small number of particles. Using physical knowledge of the system, these undesired correlations can be corrected. $\boldsymbol{\Phi}^x$ is the localization matrix. Applying localization, the covariance 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 $\boldsymbol{\Phi}^x \circ \boldsymbol{P}^x$, where $\circ$ is the element-wise product.

states and outputs which, in reality, are uncorrelated. This can lead to the drift of unobservable states, and eventually to instability of the 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 $i^{\text{th}}$ state entry $(\boldsymbol{x}_k)_i$ belongs to a physical location in the farm $\boldsymbol{s}_i$. Then, looking at an arbitrary state covariance term $(i,j)$,

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

we define the physical distance between these two states as $\Delta s_{i,j} = ||\boldsymbol{s}_i - \boldsymbol{s}_j||_2$. 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

$$\phi(c_{i,j}) = \begin{cases} -\frac{1}{4}c_{i,j}^5 + \frac{1}{2}c_{i,j}^4 + \frac{5}{8}c_{i,j}^3 - \frac{5}{3}c_{i,j}^2 + 1 & \text{if } 0 \le c_{i,j} \le 1, \\ \frac{1}{12}c_{i,j}^5 - \frac{1}{2}c_{i,j}^4 + \frac{5}{8}c_{i,j}^3 + \frac{5}{3}c_{i,j}^2 - 5c_{i,j} + 4 - \frac{2}{3}\frac{1}{c_{i,j}} & \text{if } 1 < c_{i,j} \le 2, \\ 0 & \text{otherwise,} \end{cases} \tag{24}$$

with $c_{i,j} = \frac{||\Delta \boldsymbol{s}_{i,j}||_2}{L}$ a normalized distance measure, with $L$ the cut-off distance. Applying (24) for the covariance matrices $\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1}$ and $\boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1}$ , we can define the localization matrices

$$\boldsymbol{\Phi}^z = \begin{bmatrix} \phi(c^z_{1,1}) & \cdots & \cdots\phi(c^z_{1,M}) \\ \vdots & \ddots & \\ \phi(c^z_{M,1}) & & \phi(c^z_{M,M}) \end{bmatrix}, \qquad \boldsymbol{\Phi}^{xz} = \begin{bmatrix} \phi(c^{xz}_{1,1}) & \cdots & \cdots\phi(c^{xz}_{1,M}) \\ \vdots & \ddots & \\ \phi(c^{xz}_{N,1}) & & \phi(c^{xz}_{N,M}) \end{bmatrix},$$

where $c^z_{i,j}$ is the normalized distance between two measurements $i$ and $j$, and $c^{xz}_{i,j}$ is the normalized distance between state $i$ and measurement $j$, respectively. Finally, localization and inflation can be incorporated into (20) and (21) by

$$\boldsymbol{P}^{\boldsymbol{z}}_{k|k-1} = \quad \boldsymbol{\Phi}^z \circ \frac{1}{Y-1}\sum_{i=1}^{Y}\left(\left(\boldsymbol{\zeta}^i_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1}\right)\left(\boldsymbol{\zeta}^i_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1}\right)^T\right), \tag{25}$$

$$\boldsymbol{P}^{\boldsymbol{xz}}_{k|k-1} = r\cdot\boldsymbol{\Phi}^{xz} \circ \frac{1}{Y-1}\sum_{i=1}^{Y}\left(\left(\boldsymbol{\zeta}^i_{k|k-1} - \overline{\boldsymbol{\zeta}}_{k|k-1}\right)\left(\boldsymbol{\psi}^i_{k|k-1} - \overline{\boldsymbol{\psi}}_{k|k-1}\right)^T\right), \tag{26}$$

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. 3. A significant increase in performance is shown, especially for smaller numbers of particles. This is in agreement with what was seen in previous work (Doekemeijer et al., 2017). Furthermore, performance is more consistent. Additionally, note that there is no increase in computational cost, as the covariance matrices are made sparse, leading to a cost reduction in the calculation of (10), which makes up for the extra operations of (25) and (26). Also, note that the localization matrices are time-invariant and can be calculated offline.

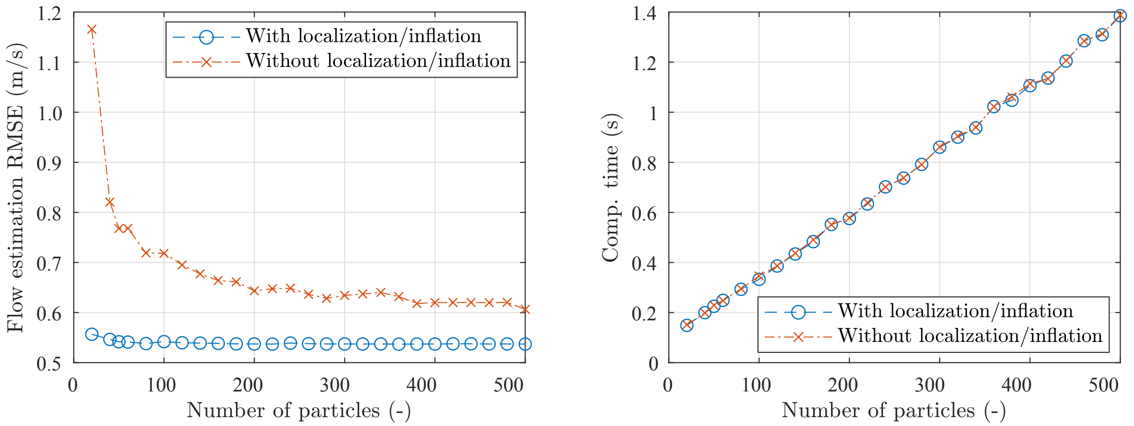

**Figure 3.** This figure shows the estimation performance and computational cost (parallelized, 8 cores) of the EnKF for a range of ensemble sizes, with and without inflation and localization. Great improvement is seen for estimation accuracy, at no additional computational cost. The simulation scenario is described in detail in Section 4.2, and the results presented here are rather meant as an indication.

### 3.6 Synthesizing an online model calibration solution

Certain model parameters such as $\ell_s$ 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, $\ell_s$ is concatenated to the state vector as random walk model, with a certain standard deviation (covariance). Higher values of $\ell_s$ lead to more wake recovery, making the calibration solution adaptable to varying turbulence levels. This adds one scalar entry to $\boldsymbol{x}_k$, which is a negligible addition in terms of computational cost.

Furthermore, a proposal is made for the estimation of the freestream wind speed $U_\infty$. This is suggested to be done using the turbine's power generation measurements, following the ideas of Gebraad et al. (2016) and Shapiro et al. (2017b). Using the wind vanes 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 $\text{Ш} \in \mathbb{Z}^{\text{Ж}}$ as a vector specifying the upstream turbines, with Ж 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~~ by inverting the turbine power expression from WFSim (Boersma et al., 2017b). One wind-farm-wide freestream wind speed $U_\infty$ is then calculated using actuator disk theory. Smoothing results with a low-pass filter with time-constant $c_{u_\infty}$ on the average of $U_{\infty_i}$ for each upstream turbine $i$, we obtain

$$c_{u_\infty}\frac{\partial U_\infty}{\partial t} = \frac{1}{\text{Ж}}\sum_{i\in\text{Ш}}\left(\sqrt[3]{\frac{P^{\text{meas.}}_{\text{turb},i}}{\frac{c_p}{2}\rho A C'_{T_i}\cos\left(\gamma_i\right)^3}}\cdot\left(1+\frac{1}{4}C'_{T_i}\right)\right) - U_\infty, \qquad (27)$$

where it is assumed that $U_{\infty_i} \approx U_{r_i}\left(1+\frac{1}{4}\cdot C'_{T_i}\right)$ ~~,~~ when $\gamma_i \approx 0$, with $U_{r_i}$ the wind speed at the rotor of turbine $i$. Research is currently ongoing on how to best incorporate the effects of turbine yaw ($\gamma \neq 0$) into the definition of $C'_T$. Furthermore, ~~$c_{u_\infty}$~~

~~is the time constant of the first-order low-pass filter~~$\rho$ is the air density, $A$ is the rotor swept area, and $P_{\text{turb},i}^{\text{meas.}}$ is the measured instantaneous power capture of turbine $i$.[2]

Combining these elements yields an efficient, modular, and accurate model calibration solution for WFSim. The model states are estimated using SCADA and/or LIDAR data, of which the former is readily available, and the latter becoming more popular. State estimation paired with parameter estimation improves the accuracy of the surrogate model, potentially leading to more accurate control. Additionally, the freestream wind speed is estimated using readily available SCADA data. This control solution is implemented in MATLAB, and leverages the numerically efficient precompiled solvers and parallelization for model propagation. 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.

# 4 Results

In this section, the calibration solution detailed in Section 3 will be validated using high-fidelity simulations. First, the model used to generate the 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 presented results, pressure terms are ignored in the state vector, as they appeared unnecessary for state estimation in previous work (Doekemeijer et al., 2017). Furthermore, for simplicity and due to lack of information, the process and measurement noise will be assumed to be uncorrelated, $\boldsymbol{S}_k = 0$, and $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ 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.

## 4.1 SOWFA

High-fidelity simulation data is generated using the Simulator fOr Wind Farm Applications (SOWFA), developed by the National Renewable Energy Laboratory. SOWFA provides accurate flow data at a fraction of the cost of field tests. 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, and turbulent structures smaller than the discretization are approximated using subgrid-scale models to suppress computational cost. (Churchfield et al., 2012). The turbine rotor is modeled using an actuator line representation as derived from Sorensen and Shen (2002). 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., 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.

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

### 4.2 2-turbine ALM with turbulent inflow

In this section, 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. This simple wind farm contains two NREL 5-MW baseline turbines with $D = 126.4$ m, separated $5D$ in stream-wise direction. This LES simulation was described in more detail in Annoni et al. (2016). 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 $\ell_s$ 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 | - | $3.0\text{km} \times 3.0\text{km} \times 1.0\text{km}$ | $1.9\text{km} \times 0.80\text{km}$ |
| Cell size near rotors | - | $3\text{m} \times 3\text{m} \times 3\text{m}$ | $38\text{m} \times 33\text{m}$ |
| Cell size outer regions | - | $12\text{m} \times 12\text{m} \times 12\text{m}$ | $38\text{m} \times 33\text{m}$ |
| Rotor model | - | ALM | ADM ($c_f = 1.4$, $c_p = 0.95$) |
| Inflow wind speed | $U_\infty$ | 8.0 m/s | 8.0 m/s |
| Atmospheric turbulence | - | Turbulent inflow, $\text{TI}_\infty = 5.0\%$ | $d' = 1.8 \cdot 10^2$ m, $d = 6.1 \cdot 10^2$ m, $\ell_s = 1.8 \cdot 10^{-2}$ |

Firstly, the three KF variants will be compared in Section 4.2.1. Secondly, in Section 4.2.2, estimation using different information sources is compared. Thirdly, the 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

In this simulation study, four estimation cases are compared: 1) the ExKF, 2) the UKF, 3) the EnKF, and 4) the open-loop (OL) simulation, i.e., without estimation. The focus here is on state-only estimation, thus excluding $\ell_s$. Flow measurements downstream of each turbine are assumed (e.g., using LiDAR), their locations denoted as red dots in Fig. 4, 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 4 shows state (flow field) estimation of the three KF variants for two time instants, $t = 300$ s and $t = 700$ s. In this figure, $(\Delta \boldsymbol{u})_\bullet \in \mathbb{R}^{N_u}$ is defined as the absolute error between the estimated and true longitudinal flow velocities in the field.

Looking at Fig. 4, 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).

**Table 2.** Covariance settings for the KF variants, with $\boldsymbol{I}_\bullet$ the $\mathbb{R}^{\bullet\times\bullet}$ identity matrix. The full cov. matrices are diagonal concatenations of the entries. For example, $\boldsymbol{P}_0$ is $\textbf{diag}\left(\boldsymbol{P}_{0,u},\ \boldsymbol{P}_{0,v}\right)$ and $\textbf{diag}\left(\boldsymbol{P}_{0,u},\ \boldsymbol{P}_{0,v},\ \boldsymbol{P}_{0,\ell_s}\right)$ for state-only and state-parameter estimation, respectively.

| **Variable** | **Symbol** | **Units** | **Value** |
|---|---|---|---|
| Init. state error cov. of $\boldsymbol{u}_k$ | $\boldsymbol{P}_{0,u}$ | (m/s)$^2$ | $1.0 \cdot 10^{-1} \cdot \boldsymbol{I}_{N_u}$ |
| Init. state error cov. of $\boldsymbol{v}_k$ | $\boldsymbol{P}_{0,v}$ | (m/s)$^2$ | $1.0 \cdot 10^{-1} \cdot \boldsymbol{I}_{N_v}$ |
| Init. state error cov. of $\ell_{s_k}$ | $\boldsymbol{P}_{0,\ell_s}$ | – | $5.0 \cdot 10^{-1}$ |
| Model error cov. of $\boldsymbol{u}_k$ | $\boldsymbol{Q}_{0,u}$ | (m/s)$^2$ | $1.0 \cdot 10^{-2} \cdot \boldsymbol{I}_{N_u}$ |
| Model error cov. of $\boldsymbol{v}_k$ | $\boldsymbol{Q}_{0,v}$ | (m/s)$^2$ | $1.0 \cdot 10^{-4} \cdot \boldsymbol{I}_{N_v}$ |
| Model error cov. of $\ell_{s_k}$ | $\boldsymbol{Q}_{0,\ell_s}$ | – | $1.0 \cdot 10^{-4}$ |
| Meas. error cov. of flow | $\boldsymbol{R}_{u,v}$ | (m/s)$^2$ | $1.0 \cdot 10^{-2} \cdot \boldsymbol{I}_{M_{u,v}}$ |
| Meas. error cov. of $P$ | $\boldsymbol{R}_P$ | (W)$^2$ | $1.0 \cdot 10^{8} \cdot \boldsymbol{I}_{N_T}$ |

**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.** | **2-turb.** | **9-turb.** |
|---|---|---|---|---|
| **Variable** | **ExKF** | **UKF** | **EnKF** | **EnKF** |
| Number of particles, $Y$ | – | 4275 | 50 | 50 |
| Tuning parameters | – | $\alpha$ 1.0<br>$\beta$ 2.0<br>$\kappa$ 0 | $L$ 131 m<br>$r$ 1.025 | $L$ 131 m<br>$r$ 1.025 |
| Comp. cost/it. | 16.2 s | 14.0 s | 0.25 s | 1.2 s |

### 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. 5. 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.

The KF settings are displayed in Tables 2 and 3. In Fig. 5 it can be seen that SCADA data allows comparable performance 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.

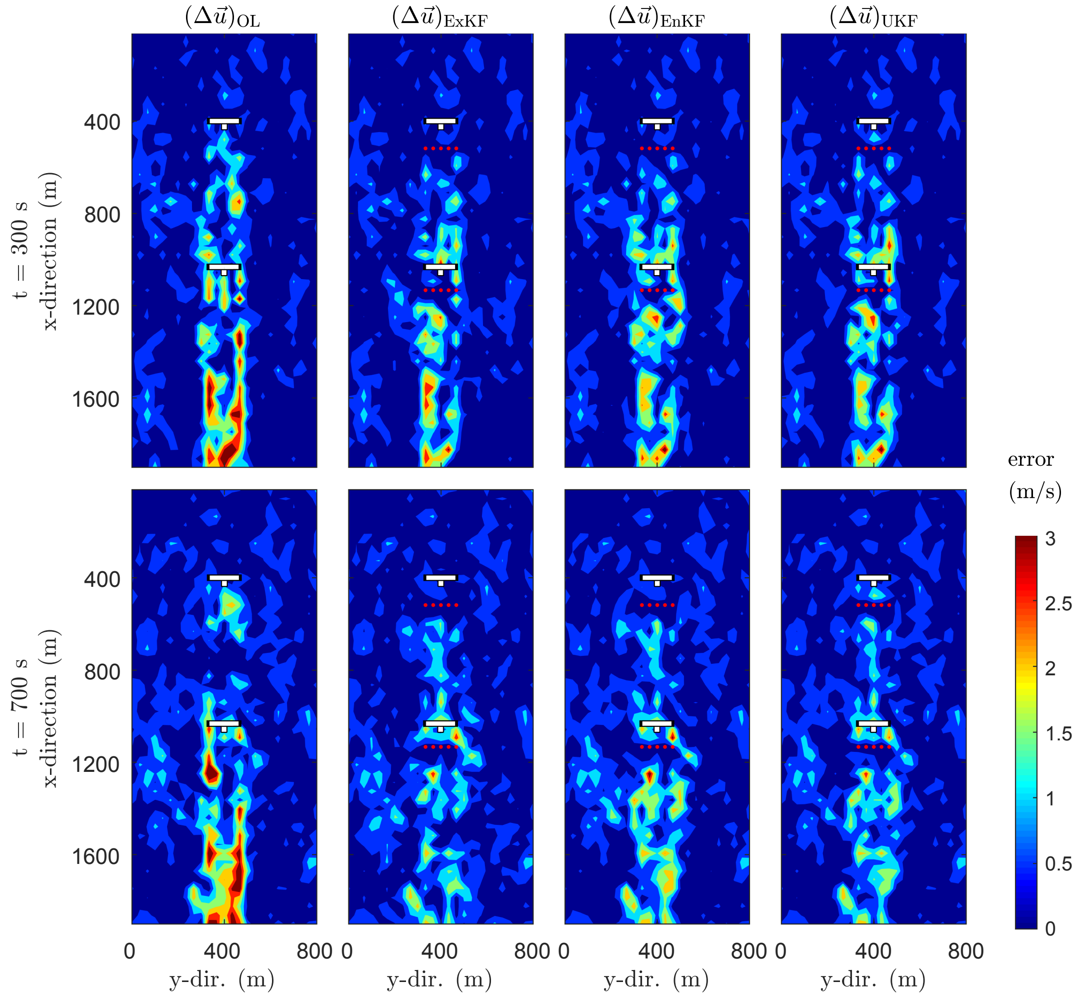

**Figure 4.** Comparison of absolute values of the estimation errors (in long. flow fields) for state-only estimation with the ExKF, EnKF and UKF at $t = 300$ s and $t = 700$ s, with $(\Delta\boldsymbol{u})_\bullet = |\boldsymbol{u}_\bullet - \boldsymbol{u}_{\text{SOWFA}}|$. The model and KF settings 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 nearby the measurements.

### 4.2.3 Joint state-parameter estimation

Forecasting, as used in predictive control, benefits from the calibration of model parameters ~~such as $\ell_s$~~ in addition to the ~~model~~ states. Joint state-parameter estimation using flow measurements downstream of each turbine (as shown in the rightmost plots in Fig. 5) disturbed by zero-mean white noise with $\sigma = 0.10$ m/s for the EnKF and UKF is displayed in Fig. 6~~, where $\ell_s$ is additionally estimated. The higher $\ell_s$, the more wake recovery is modeled by WFSim. The~~. The KF settings are shown in Tables 2 and 3. ~~From this figure~~At $t = 0$ s, both the OL and the KF simulations start with the same (wrong) value for $\ell_s$. Then, every second, (noisy) measurements are fed into the KFs, and the state vector as well as the model parameter $\ell_s$ are estimated.

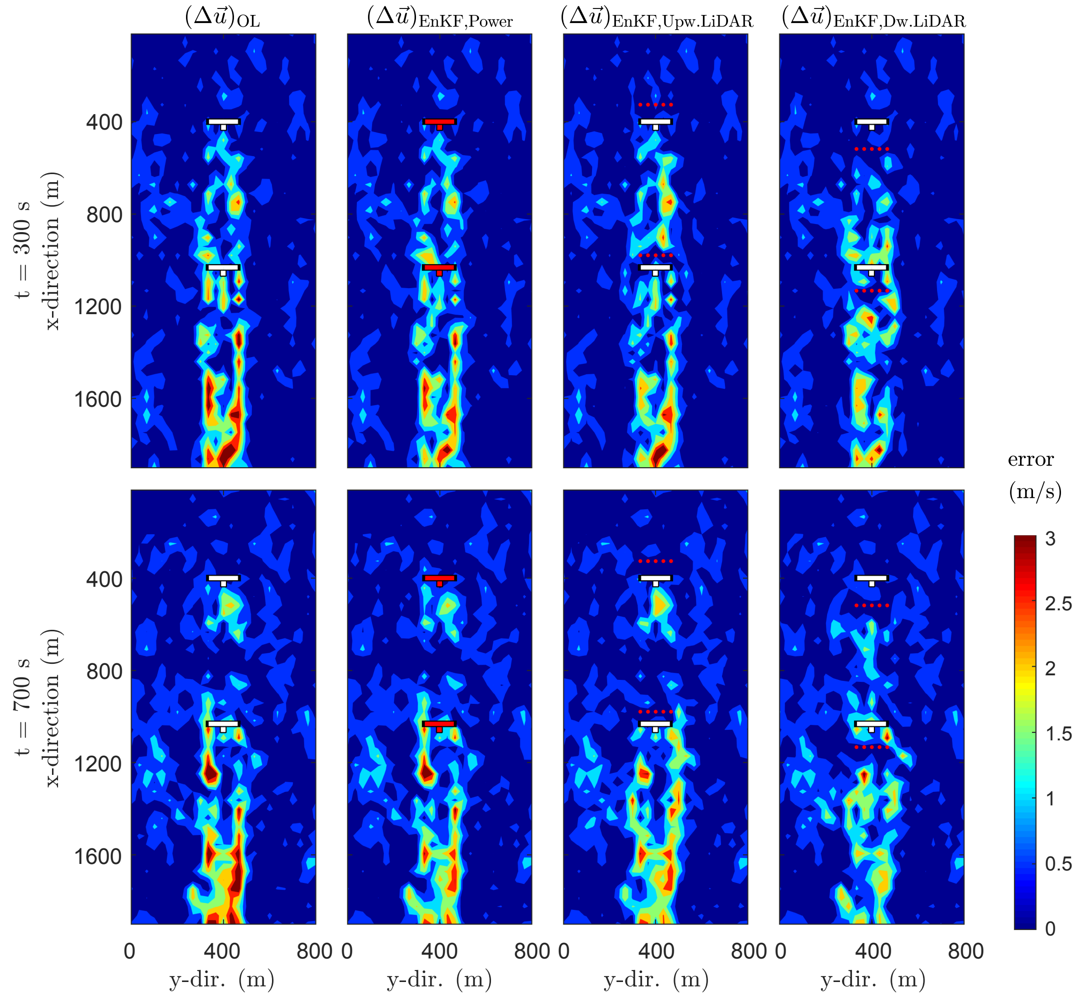

**Figure 5.** Comparison of absolute values of the estimation errors (in long. flow fields) for state-only estimation with the EnKF for various sensor configurations: using turb. power measurements, using flow measurements with a LiDAR system pointing upstream, and using flow measurements with a LiDAR system pointing downstream of the rotor. Here, $(\Delta \boldsymbol{u})_{\bullet} = |\boldsymbol{u}_{\bullet} - \boldsymbol{u}_{\text{SOWFA}}|$. ~~Wind is coming in~~ Here, wind flows from ~~the~~ top ~~, flowing towards the~~ to bottom. The sensors are depicted by red dots (flow meas.) or red turbines (power meas.), not to be confused with estimation error.

However, for the OL simulation, no measurements are fed in: the state vector is simply updated with the nominal model, and the value for $\ell_s$ remains the same throughout the simulation. Now, after 600 s (left plot in Fig. 6) and 900 s (right plot in Fig. 6), a forecast is started, meaning no measurements are available after that time. At that moment, the OL model still has the same (poor) value for $\ell_s$ as at $t = 0$ s, while the value for $\ell_s$ in the KFs has improved. From Fig. 6, it becomes clear that the ~~flow field~~ estimates are not only improved for the 3-minute forecast, but are also consistently better than the non-calibrated (open-loop) model's 10-minute forecast due to the ~~real-time~~ estimation of $\ell_s$.[3] Furthermore, the EnKF performs comparably to the UKF at a lower computational cost. Note that the EnKF even outperforms the UKF in this simulation, 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.~~

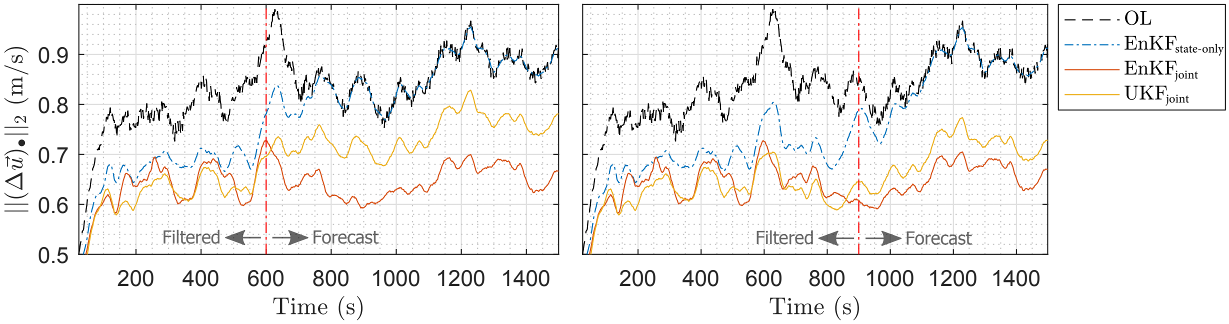

**Figure 6.** Comparison of forecasting performance for state-only and joint state-parameter ($\ell_s$) estimation with the EnKF and UKF, where measurements are available up until the vertical dashed lines, after which the estimation becomes a forecast. Here, the 2-norm of the estimation error is plotted along the $y$-axis, with $(\Delta \boldsymbol{u})_\bullet = |\boldsymbol{u}_\bullet - \boldsymbol{u}_{\text{SOWFA}}|$.

### 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. 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 $5D$ 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. 8. 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.

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.

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 $\ell_s$, and the freestream flow speed $U_\infty$ are estimated in real-time using exclusively ~~readily available~~ (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 $\ell_s$ and $U_\infty$ to investigate convergence. The other open-loop simulation will be initialized with a poor value for $\ell_s$ but a correct value for $U_\infty$ for comparison.

[3] Note 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 $\times$ 3.0km $\times$ 1.0km | 1.9km $\times$ 0.80km |
| Cell size near rotors | - | 3m $\times$ 3m $\times$ 3m | 25m $\times$ 38m |
| Cell size outer regions | - | 12m $\times$ 12m $\times$ 12m | 25m $\times$ 38m |
| Rotor model | - | ALM | ADM ($c_f = 2.0$, $c_p = 0.97$) |
| Inflow wind speed | $U_\infty$ | 12.03 m/s | 9.0 m/s and 12.0 m/s (OL)<br>9.0 m/s (EnKF) |
| Atmospheric turbulence | - | Turbulent inflow, $\mathrm{TI}_\infty = 4.7\%$ | $d' = 3.8 \cdot 10^1$ m, $d = 5.2 \cdot 10^2$ m, $\ell_s = 3.9 \cdot 10^{-2}$ |

In Fig. 7, it can be seen that the EnKF is successful in estimating $U_\infty$ and $\ell_s$ after ~~about~~ 300 s using only wind turbine power measurements. Furthermore, the flow fields of 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. 8. From this figure, it can be seen that the EnKF has 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 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 finally the estimation error converges.

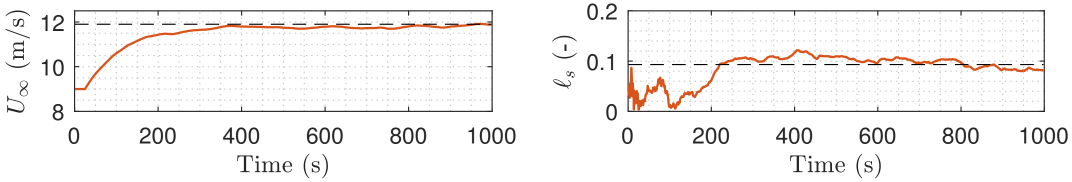

**Figure 7.** Convergence of $\ell_s$ and $U_\infty$ using the EnKF. In dashed lines are the grid-searched optimal constant values for the open-loop simulation. With power measurements only, the EnKF is able to estimate these parameters successfully in addition to the model states.

The power forecasting performance is shown in Fig. 9 and Table 5. As also seen in Fig. 7, the EnKF converges after 300 s, and indeed the power forecasts outperform those of the OL 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 $t \approx 200$ s. This can be related to the mismatch in $U_\infty$, which takes approximately 300 s to converge to the true value of 12 m/s, as seen in Fig. 7. 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 $C'_T$.

Finally, the forecasts for flow at times $t = 300$ s and $t = 600$ s are examined in Fig. 10. The large flow estimation mismatch in the EnKF at $t < 250$ s quickly reduces and for $t \geq 250$ s the EnKF estimation is consistently better than both the OL cases. This has to do with the convergence of the model parameters $\ell_s$ and $U_\infty$, and the estimation of the states surrounding the turbines using the power measurements.

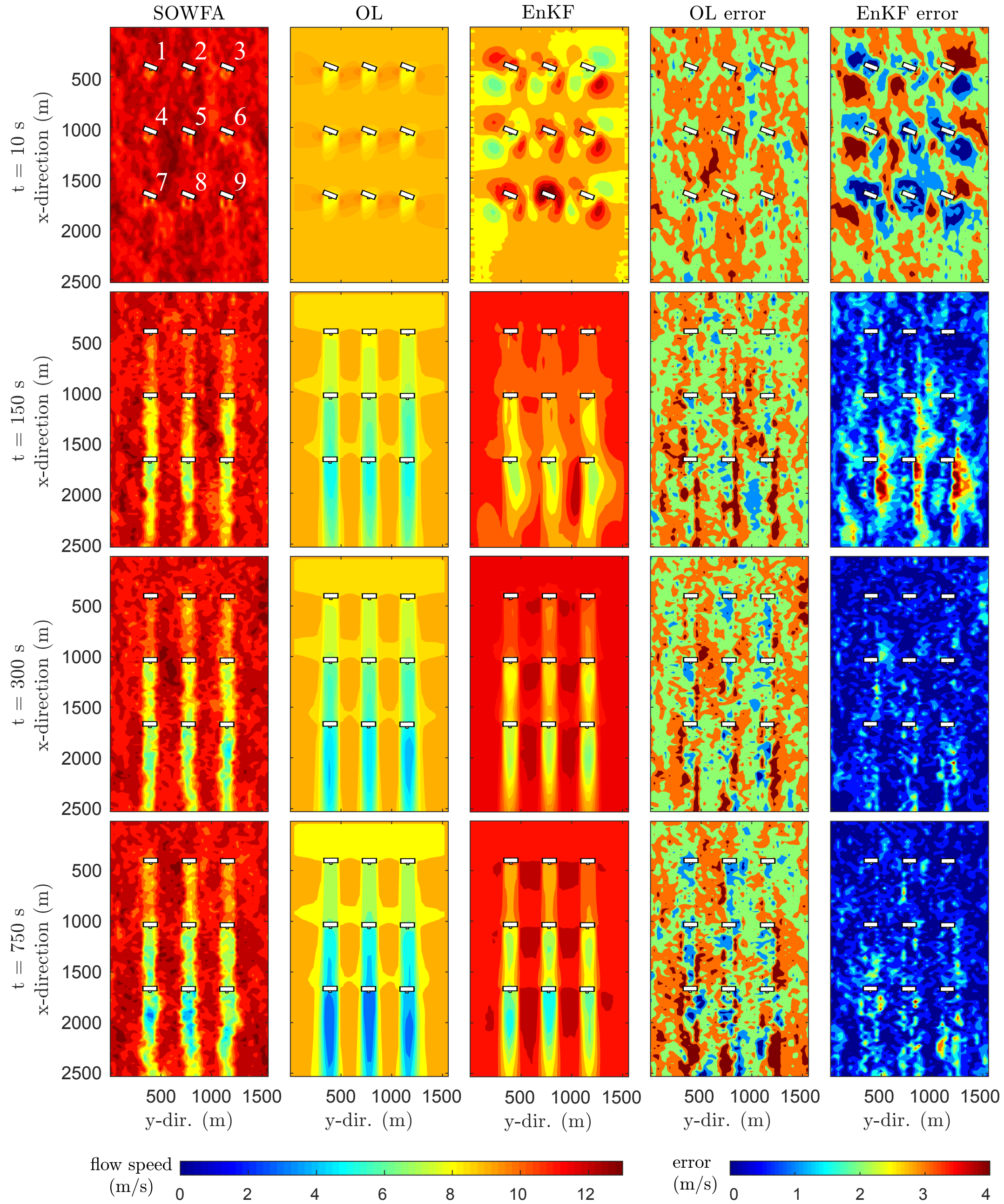

**Figure 8.** Comparison of absolute values of the estimation errors (in long. flow fields) for state-parameter estimation with the EnKF. Wind is coming in from the top and flows downwards. The variables $U_\infty$ and $\ell_s$ are incorrectly initialized in both the OL and the EnKF. In the EnKF, $U_\infty$ and $\ell_s$ are estimated in addition to the states, using only turbine power measurements. The EnKF quickly converges for the states, and more slowly for $\ell_s$ and $U_\infty$. After 300 s, the EnKF has converged to a negligible estimation error.

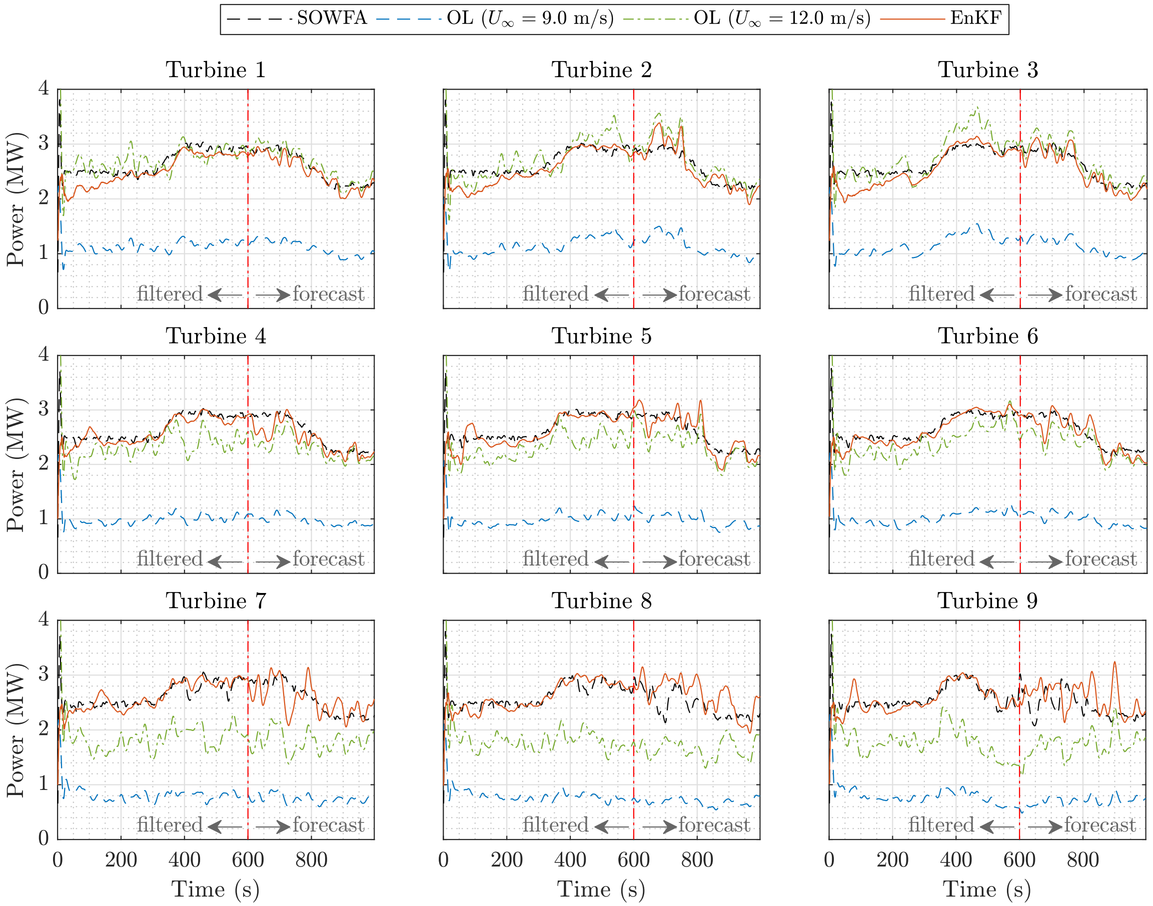

**Figure 9.** Comparison of power forecasting using the EnKF with measurements available up until time $t = 600$ s. After convergence $U_\infty$ (as seen as a positive power slope for the first row of turbines), $\ell_s$ is also calibrated. After convergence, forecasting is ~~significantly~~ better than 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.~~

**Table 5.** Turbine-averaged RMSE in power timeseries of Fig. 9 (compared to SOWFA). The lower the RMSE, the better the forecast.

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

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

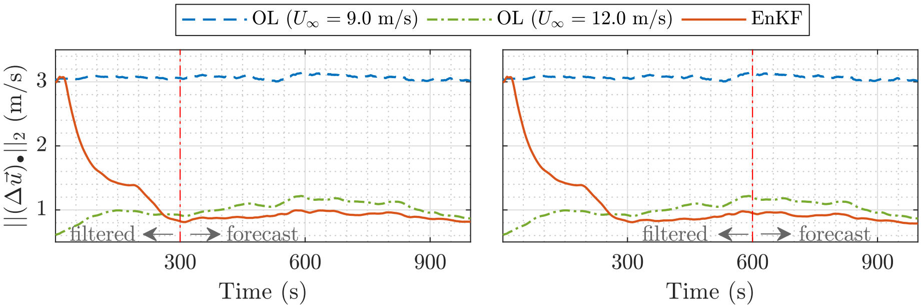

**Figure 10.** Comparison of flow field estimation for the 9-turbine case. Measurements are available until $t = 300$ s (left) and $t = 600$ s (right), respectively. The EnKF converges to the true $U_\infty$ after 300 s. After convergence, the forecasts are significantly better than in open-loop.

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 wind farm model "WFSim", relying on an Ensemble Kalman filter (EnKF) at its core. The joint state-parameter calibration solution was tested in two high-fidelity simulation case studies. Using exclusively SCADA measurements which are readily available in current wind farms, the adaptability to model discrepancies in a 9-turbine wind farm simulation was shown, at a low computational cost of 1.2 s per timestep on an 8-core CPU. Specifically, the 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. 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.

*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., Buckspan, 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.

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, 2016.

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.

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. 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.

Ela, E., Gevorgian, V., Fleming, P. A., Zhang, Y. C., Singh, M., Muljadi, E., Scholbrock, A. K., Aho, J., Buckspan, 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.

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.

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.

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.

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.

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.