Enhanced Kalman Filtering for a 2D CFD NS Wind Farm Flow Model

B M Doekemeijer[1], J W van Wingerden[1], S Boersma[1] and L Y Pao[2]

[1] Faculty of Mechanical Engineering, Delft University of Technology, The Netherlands
[2] Faculty of Electrical Engineering, University of Colorado Boulder, United States

E-mail: B.M.Doekemeijer@tudelft.nl

Abstract. Wind turbines are often grouped together for financial reasons, but due to wake development this usually results in decreased turbine lifetimes and power capture, and thereby an increased levelized cost of energy (LCOE). Wind farm control aims to minimize this cost by operating turbines at their optimal control settings. Most state-of-the-art control algorithms are open-loop and rely on low fidelity, static flow models. Closed-loop control relying on a dynamic model and state observer has real potential to further decrease wind's LCOE, but is often too computationally expensive for practical use. In this paper two time-efficient Kalman filter (KF) variants are outlined incorporating the medium fidelity, dynamic flow model "WindFarmSimulator" (WFSim). This model relies on a discretized set of Navier-Stokes equations in two dimensions to predict the flow in wind farms at low computational cost. The filters implemented are an Ensemble KF and an Approximate KF. Simulations in which a high fidelity simulation model represents the true wind farm show that these filters are $10^1 - 10^2$ times faster than a regular KF with comparable or better performance, correcting for wake dynamics that are not modeled in WFSim (noticeably, wake meandering and turbine hub effects). This is a first big step towards real-time closed-loop control for wind farms.

1. Introduction

The recent 2015 UN Climate Change Conference in Paris once again stressed the importance of renewable energy sources, among which is wind. To make implementations of wind energy feasible, its levelized cost of energy (LCOE)[1] has to be able to compete with that of other energy sources. For this reason, wind turbines are often placed together in wind farms. However, grouping turbines together gives rise to the development of, and interactions with, wind wakes, often resulting in a plant-wide decreased power capture[2] and increased turbine structural loading compared to an equal number of isolated turbines. Wind farm control aims to counter this and minimize the LCOE of wind. Notable findings in this relatively new area of research include farm layout optimization [1, 2], wake redirection control [2, 3], and active power control [4, 5].

Advancements in wind farm control have gone hand in hand with advancements in modeling, as control algorithms typically rely on low fidelity, control-oriented, static flow models. Unfortunately, this is often limited to open-loop control. Furthermore, high fidelity, dynamic models based on the Navier-Stokes (NS) equations have allowed relatively inexpensive flow

[1] The LCOE is a measure to compare different methods of energy generation in terms of financial feasibility.
[2] Power capture losses in wind plants may be up to 50%, according to some numerical studies [1].

analysis and observer testing at high accuracy, but are computationally too expensive for real-time control applications. Medium fidelity, dynamic flow models aim to bridge this gap, and in some cases allow closed-loop control, further reducing the LCOE of wind. However, the need for a state observer and the increase in computational cost remain challenges.

The goal of this work is to design a time-efficient state observer using the medium fidelity, dynamic flow model "WindFarmSimulator" (WFSim) [6, 7], in pursuit of real-time closed-loop control. Observers will provide estimations of the flow field correcting for unmodeled dynamics (thus allowing the use of a lower fidelity, time-efficient model), measurement noise, and limited availability (temporally and spatially) of measurements. The concept is visualized in Figure 1.

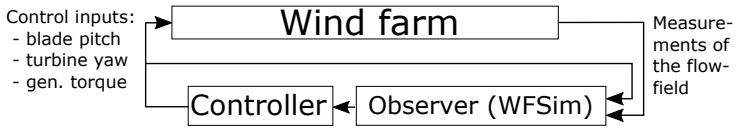


Figure 1: Closed-loop control for wind farms

In this paper, first the WFSim model is outlined in Section 2. Secondly, two enhanced KF designs are described in Section 3. Thirdly, the simulation setup and simulation results are presented in Sections 4 and 5, respectively. Finally, conclusions are drawn in Section 6.

2. WindFarmSimulator

This section introduces WFSim and modifications to this model to allow direct implementation of state-of-the-art filtering algorithms. For a more detailed explanation, please see [8].

2.1. The flow model

WFSim is a medium fidelity, dynamic flow model that predicts the flow velocity vectors in a wind farm at hub height in predefined meshings using the spatially and temporally discretized 2D NS equations following a computational fluid dynamics (CFD) solution [9]. WFSim employs the actuator disk model (ADM) [10] to calculate the aerodynamic forces on the flow by the rotor blades. Furthermore, it includes a mixing length turbulence model [11] to account for wake recovery. What makes WFSim unique is its reformulation into an implicit, nonlinear state space system by projecting away the continuity equations, resulting in the flow model

$$\begin{aligned} E(\alpha_k)\alpha_{k+1} =& A\ \alpha_k + B(\alpha_k)\beta_k + S(\alpha_k), \\ \begin{bmatrix} \overline{u}_k \\ \overline{v}_k \end{bmatrix} =& Q_p \alpha_k + B_p, \end{aligned} \tag{1}$$

with $\alpha_k \in \mathbb{R}^N$ the system state vector at time k, and $\left[\overline{u}_k; \overline{v}_k\right] \in \mathbb{R}^M$ the collocated 2D velocity vectors in the grid. N is proportional to the number of grid points in the mesh (i.e., refinement of flow fields). Note that α_k has no physical interpretation due to the aforementioned projection. The system input is $\beta_k \in \mathbb{R}^O$, with O equal to the number of turbines. β_k is a representation of the axial induction factors, and can be translated into physical turbine settings such as generator torque, blade pitch, and yaw. $E \in \mathbb{R}^{N\times N}$, $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times O}$, $S \in \mathbb{R}^N$, $Q_p \in \mathbb{R}^{M\times N}$, and $B_p \in \mathbb{R}^M$ are system matrices, of which E, B, and S depend on the state α_k.

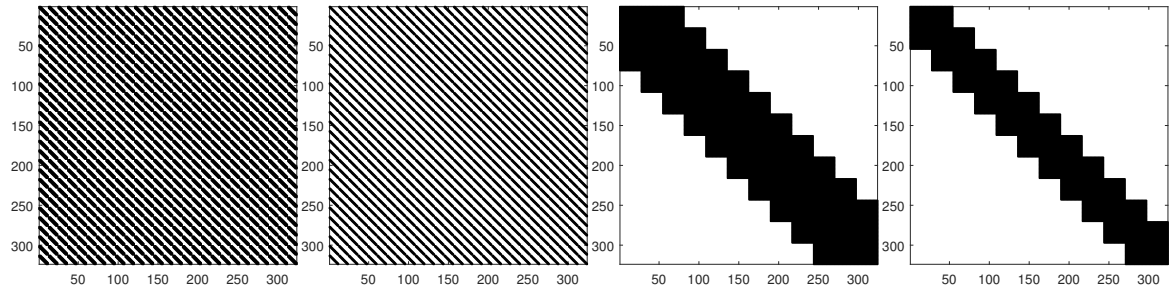
2.2. Model manipulation

While literature on state observers for implicit systems like Equation (1) exists (e.g., [12, 13]), for WFSim these algorithms are too computationally complex for real-time control and often lead to numerical instability. Hence, a different approach is followed. First, Equation (1) is reformulated

in a more common form by extending the state vector with a constant entry allowing elimination of the offset terms $S(\alpha_k)$ and B_p, and performing matrix inversion of $E(\alpha_k)$, resulting in

$$\begin{aligned}\begin{bmatrix}\alpha_{k+1}\\1\end{bmatrix} &= \overbrace{\begin{bmatrix}E(\alpha_k)^{-1}A & E(\alpha_k)^{-1}S(\alpha_k)\\0 & 1\end{bmatrix}}^{F(\alpha_k)}\begin{bmatrix}\alpha_k\\1\end{bmatrix} + \begin{bmatrix}E(\alpha_k)^{-1}B(\alpha_k)\\0\end{bmatrix}\beta_k,\\ \begin{bmatrix}\overline{u}_k\\\overline{v}_k\end{bmatrix} &= \begin{bmatrix}Q_p & B_p\end{bmatrix}\begin{bmatrix}\alpha_k\\1\end{bmatrix}.\end{aligned} \tag{2}$$

In its current form, the calculation of $E(\alpha_k)^{-1}A$ at each time step is too computationally expensive for control applications due to high dimensionality (typically, $N = 10^2 - 10^3$). Therefore, computational cost is reduced by applying the Reverse Cuthill-McKee (RCM) algorithm on Equation (2), typically resulting in a computational effort reduction of 90% or more [14, 15]. The RCM algorithm rearranges the rows and columns of a matrix to transform a sparse matrix into a sparse banded matrix, as demonstrated in Figure 2.



(a) $E(\alpha_k) \in \mathbb{R}^{324\times324}$ (b) $A \in \mathbb{R}^{324\times324}$ (c) RCM(E) $\in \mathbb{R}^{324\times324}$ (d) RCM(A) $\in \mathbb{R}^{324\times324}$

Figure 2: Matrices $E(\alpha_k)$ and A under the RCM algorithm, in which respectively 63% and 76% of the entries are zero. This typically reduces computational cost of $E(\alpha_k)^{-1}A$ by over 90%.

The system in Equation (2) allows direct implementation of state-of-the-art estimators. Note that high dimensionality is still present, and thus emphasis is put on time-efficient algorithms.

3. Observer design

Two observers are implemented, both derived from the traditional Kalman filter (KF). However, they are fundamentally different in their way of reducing computational cost. The first filter, the Approximate KF (ApKF), simplifies the system model while retaining the original update equations. The second filter, the Ensemble KF (EnKF), instead relies on the original system model, yet approximates the KF update algorithm. Both algorithms are briefly described next.

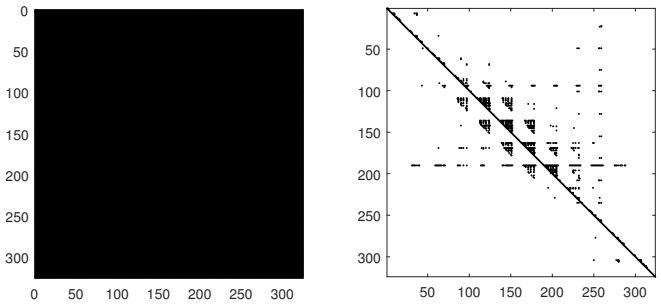
3.1. Approximate Kalman filter

The ApKF relies on the original KF update algorithm, while enforcing sparsification of two main matrices in pursuit of reducing computational cost. Firstly, all off-diagonal elements in the state covariance matrix P_k are neglected, implicitly assuming the system states are uncorrelated. From numerical results it is noted that the diagonal elements are typically a factor $10^1 - 10^2$ larger than the off-diagonal elements, validating this assumption. Secondly, $F(\alpha_k)$ in Equation (2) is made sparse by neglecting all matrix entries $[i, j]$ that meet the condition

$$|F(\alpha_0)[i, j] \cdot \alpha_0(j)| < \text{mean}\left(|F(\alpha_0)[i, :] \cdot \alpha_0|\right) \cdot z, \tag{3}$$

with z typically $0.05 - 0.10$, and i and j the row and column of the matrix, respectively. For $z = 0$, the original matrix F is retrieved. This algorithm enforces a sparsification by neglecting

matrix entries that only have a small contribution to the system update, thereby leaving the resulting update approximately intact. This sparsification typically renders over 95% of the entries sparse, as demonstrated in Figure 3 for system matrix $F(\alpha_k) \in \mathbb{R}^{324\times324}$ with $z = 0.05$.



(a) $F \in \mathbb{R}^{324\times324}$ (b) $F_{\text{sparsified}} \in \mathbb{R}^{324\times324}$

Figure 3: System matrix F under sparsification according to Equation (3), with $z = 0.05$. This typically reduces computational cost of the filter by over 95%.

Furthermore, for $z < 0.10$, typically no loss in accuracy is noted in simulation. An example comparing the ApKF with the KF is displayed in Figure 4. These simplifications reduce computation time by a factor $10^1 - 10^2$ at negligible loss in state reconstruction accuracy. An important remark is that the KF is proven exclusively for linear systems, while the system at hand is nonlinear. Simulations have shown acceptable results, yet caution must be taken.

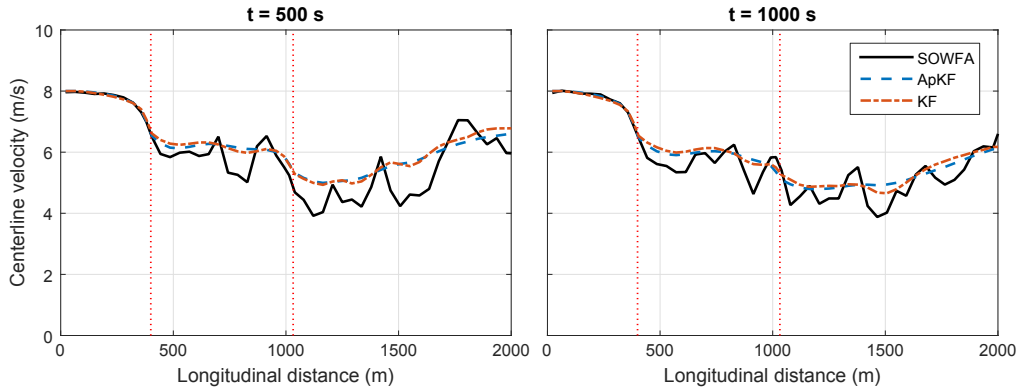


Figure 4: Simulation results showing the mean wake centerline for the ApKF ($z = 0.05$) and the KF. See Section 5 for definitions and an explanation of the simulation setup.

3.2. Ensemble Kalman filter

The second observer is an EnKF. It is a variant of the KF in which the covariance matrices are approximated by a sample covariance. The EnKF is typically applied to nonlinear models of high order ($10^3 - 10^6$ states) for its computational efficiency and ability to deal with nonlinear dynamics. More information concerning the EnKF can be found in a paper by Evensen [16]. The algorithm from the same paper is implemented for its emphasis on computational performance.[3]

[3] There is an important difference between Evensen's algorithm and the algorithm used in WFSim. In Evensen's algorithm, at each time step, the system matrices have to be recalculated for each ensemble member due to the dependency of the system matrices on the states. With a typical ensemble size of 10^2 members, this is very time consuming. Therefore, here, at each time instant the mean of all ensemble members is used to determine one set of system matrices, which are then used to update all ensemble members forward in time.

4. Simulation setup

The observers discussed in Section 3 will be tested using the high fidelity simulation model SOWFA in Section 5. First, in Section 4.1 SOWFA is discussed, after which the simulation domain is depicted in Section 4.2. Finally, turbine operation settings and atmospheric properties are mapped from the SOWFA simulation to WFSim, presented in Section 4.3.

4.1. SOWFA

The Simulator fOr Wind Farm Applications (SOWFA) is a simulation code from the National Renewable Energy Laboratory (NREL) that provides highly accurate flow data at a fraction of the cost of field tests [17]. SOWFA predicts the 3D velocity vectors in a CFD formulation using a large-eddy simulation (LES) method. It relies on the 3D incompressible NS equations for a steady or unsteady flow field, accounting for bouyancy (based on the Boussinesq approximation [18]) and Coriolis effects [19]. LES methods resolve larger scale dynamics directly, but employ a subgrid-scale model for small eddy dynamics to reduce computational cost.

For rotor modeling, SOWFA employs a more sophisticated variant of the ADM: the actuator line model (ALM). Unlike the ADM, the ALM includes individual rotor blade effects [18].

The FAST code [20] is implemented for turbine modeling. This model calculates, among others: the power production of turbines, the blade forces on the flow, the structural loading on turbines, and the dynamics of several turbine components [3].

SOWFA has been used on multiple occasions for model validation, controller testing, and flow analysis in wind farms (e.g., in [2, 3, 18, 21, 22]). Its validation is still an ongoing process. Currently, field tests have shown that SOWFA simulation results are accurate for the first 5 rows of turbines [22]. For a full description of the flow equations, please see the article by Churchfield *et al.* [23].

4.2. Domain and meshing

The SOWFA dataset used for model validation and controller testing initializes simulation from a fully uniform flow field, in which no turbulence has yet developed. A two turbine[4] case spaced 5 rotor diameters apart is simulated in a $3000 \times 3000 \times 1000$ m domain, with an increasingly refined meshing near the turbines and wakes. A horizontal 1000×2000 m plane at hub height is extracted from the SOWFA data as the area of interest, as displayed in Figure 5.

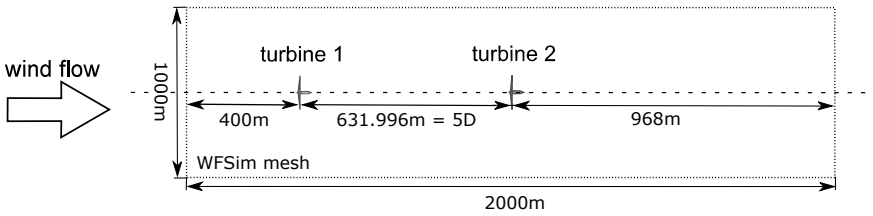


Figure 5: Simulation domain used for observer testing [3]

In WFSim, an exponential 25×50 grid meshing is used with finer spacing near turbines and coarser spacing near the boundaries, resulting in a total number of $N = 1034$ states.

4.3. Mapping SOWFA to WFSim

With the simulation domain outlined in Section 4.2, now the turbine operation settings and atmospheric settings are mapped from SOWFA to WFSim. Notably, SOWFA relies on physical turbine settings such as yaw, blade pitch, and generator torque to determine the force a turbine

[4] The NREL 5-MW turbine is used. Properties can be found in its corresponding technical report [24].

exerts on the flow. WFSim on the other hand relies on yaw and a theoretical axial induction factor derived from momentum theory. A mapping between SOWFA and WFSim is required.

In SOWFA, the upstream turbine (turbine 1) excites the flow by following a pseudo-random binary sequence (PRBS) signal on the collective pitch angle, switching between 0° and 4°. The generator torque and yaw angle are kept constant. Turbine 2 is operated under constant settings throughout the entire simulation. The axial induction factors used in WFSim are mapped accordingly. Using the tower fore-aft bending moment obtained from FAST in combination with momentum theory, the mapping is found to be as in Figure 6. For more details, see [8].

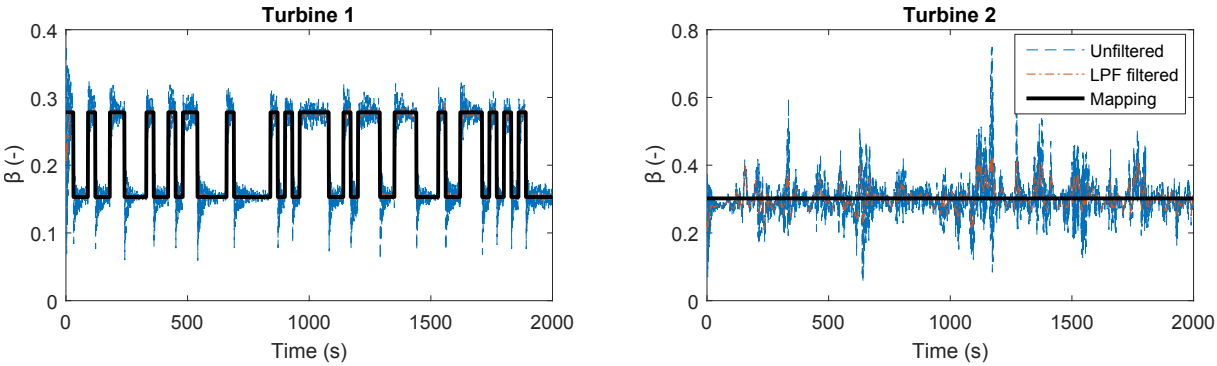


Figure 6: Mapped system input signal β_k for WFSim from SOWFA data. β_k switches between 0.15 and 0.28 for turbine 1 according to the PRBS signal, and is constant at 0.30 for turbine 2.

Note that the fore-aft tower bending moment data is low-pass filtered (LPF) to remove tower vibration effects in the estimate of β_k. From this figure, it is noted that the estimate for turbine 1 is significantly better than for turbine 2, expected to be due to wake formation and its effects. Momentum theory assumes a uniform inflow in front of the turbines. For turbine 1, this is more or less the case as the simulations are initialized from a fully uniform flow field. For turbine 2, this assumption does not hold due to wake formation by the upstream turbine. This is expected to reflect in the simulation results, with poorer estimations in the wake of turbine 2.

Finally, several atmospheric properties are mapped, as displayed in Table 1.

Table 1: Atmospheric and model parameters for WFSim

Parameter [units]	Symbol	Mapping (WFSim)	True value (SOWFA)
Air density [kg·m^3]	ρ	1.2231	N.A. (Time-varying)
Dynamic viscosity [Pa·s]	μ	$0.18 \cdot 10^{-5}$	N.A.
Free-stream flow speed (long.) [m/s]	U_∞	8.00	8.00 ± 0.14
Free-stream flow speed (lat.) [m/s]	V_∞	0.00	0.02 ± 0.04
Mixing length for turbulence model [m]	l_m	0.60	N.A.

Using the simulation settings as outlined in this section, the WFSim farm model can be validated with SOWFA, and the ApKF and the EnKF can be tested and compared.

5. Results

Simulations are performed at a 25×50 grid meshing with $N = 1034$ system states. Equidistantly spaced measurements and measurements around the turbines are fed into the observer, resulting in 23% of the system outputs (longitudinal and lateral velocity values throughout the grid)

available to the ApKF and EnKF. These measurements are disturbed by adding Gaussian noise with standard deviation $\sigma = 0.10$ m/s, comparable to current lidar standards [25].[5]

The observers are tuned according to the root mean square (RMS) error, the variance accounted for (VAF), and the quality of fit (QOF) for the mean wake centerline. The mean wake centerline is defined as the longitudinal wind speeds throughout the domain averaged in lateral direction from rotor end to rotor end. The performance measures are defined as

$$\mathrm{RMS}(\overline{u}, \overline{u}_{\mathrm{est}}) = \sqrt{\mathrm{mean}\left((\overline{u} - \overline{u}_{\mathrm{est}})^2\right)}, \tag{4}$$

$$\mathrm{VAF}(\overline{u}, \overline{u}_{\mathrm{est}}) = \left(1 - \frac{\mathrm{variance}(\overline{u} - \overline{u}_{\mathrm{est}})}{\mathrm{variance}(\overline{u})}\right) \cdot 100\%, \tag{5}$$

$$\mathrm{QOF}(\overline{u}, \overline{u}_{\mathrm{est}}) = \max\left(\left[0, 1 - \frac{||\overline{u} - \overline{u}_{\mathrm{est}}||_2^2}{||\overline{u}||_2^2}\right]\right) \cdot 100\%, \tag{6}$$

in which we want to minimize the RMS error, and maximize the VAF and QOF. Tuning the observers resulted in the settings as shown in Table 2, with n_e the number of ensembles in the EnKF, P_0 the initial state covariance matrix, R the measurement noise covariance matrix, and Q the process noise covariance matrix.

Table 2: Optimal observer settings for WFSim at 25×50 meshing

Filter	**z (-)**	**n_e (-)**	**P_0 (m/s)**	**R (m/s)**	**Q (m/s)**
ApkF	0.05	N.A.	10.0	1.0	0.050
EnKF	N.A.	200	5.0	0.10	1.00

Simulating WFSim without an observer, and with the ApKF and EnKF respectively, yields estimates of the flow field at every time instant. The mean wake centerlines for the entire simulation time can be summarized in a single figure, as displayed in Figure 7.

Figure 7 shows that WFSim predicts an averaged flow, as expected since WFSim does not include sophisticated turbulence or rotor models. The observers further improve the tracking of nonlinear flow dynamics seen in SOWFA. Quantitative results (RMS error, VAF, and QOF) are displayed in Table 3, showing that the observers significantly improve estimations with respect to the case without observer. Furthermore, the EnKF slightly outperforms the ApKF, expected to be because the EnKF deals better with nonlinear dynamics in general.

Secondly, the full flow fields are analyzed. For times $t = 200, 750$, and 1500 s, the flow fields are presented in Figure 8. From these figures, it is further confirmed that WFSim predicts an averaged flow, and neglects smaller scale dynamics, showing satisfactory results in general. The observers correct for a number of factors that WFSim estimates poorly. First of all, it is seen that the wake width is adjusted for: WFSim overestimates the wake width, expected to be due to the very simplified turbulence model. Secondly, in SOWFA, there is no wake formation behind the turbine hub, and thus two wakes form in parallel instead. Due to the simplified actuator disk model in WFSim, a single wake is modeled. The observers correct for this. Thirdly, wake meandering is somewhat accounted for. Fourthly, wake recovery is improved by the observers. Finally, comparing the EnKF to the ApKF, it is seen that the EnKF relies more on stochasticity,

[5] This is not a realistic number or location of measurements. Ongoing research is moving towards a more realistic simulation setup. Initial experiments show that reducing the number of measurements to a realistic amount (comparable to using lidar) still yields significantly better flow field estimates when compared to the absence of an observer.

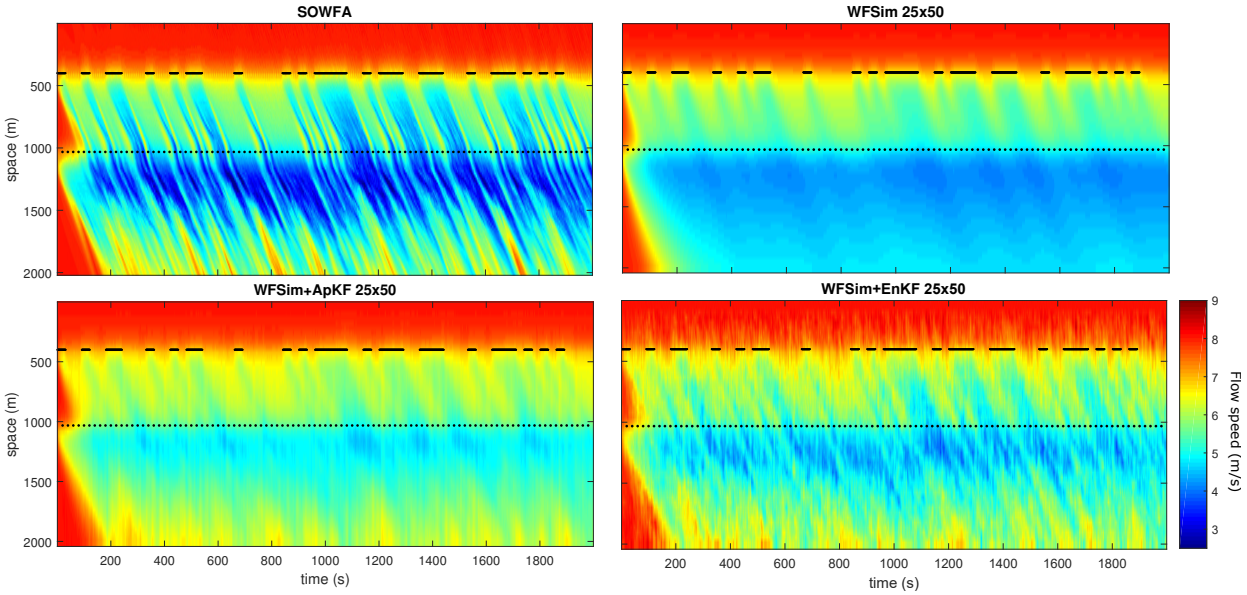


Figure 7: Mean wake centerline velocity in x-direction. The irregularly dashed line indicates flow excitation by turbine 1 ($x = 400$ m). For the gaps, the blade pitch angle is 4°, and otherwise it is 0°. The regularly dotted line indicates turbine 2 ($x = 1032$ m) operating under constant settings. Windflow is from top to bottom, with the color bar defining wind speed in m/s.

Table 3: RMS, VAF, and QOF for the mean wake centerline for WFSim, EnKF, and ApKF

	RMS (m/s)			**VAF (%)**			**QOF (%)**		
Time (s)	**WFSim**	**EnKF**	**ApKF**	**WFSim**	**EnKF**	**ApKF**	**WFSim**	**EnKF**	**ApKF**
1	0.046	0.046	0.046	17.2	17.2	17.0	100	100	100
200	0.491	0.374	0.419	85.3	88.1	88.8	99.4	99.5	99.6
500	0.836	0.438	0.590	64.6	88.5	82.6	98.1	99.1	99.5
1000	0.642	0.367	0.433	77.9	90.7	88.1	98.8	99.5	99.6
1500	0.785	0.415	0.448	68.8	89.2	88.8	98.3	99.4	99.5
1999	0.499	0.315	0.414	88.6	93.5	93.0	99.3	99.5	99.7

but therefore also deals better with model nonlinearity. Noticeably, the EnKF deals well with the underestimated wake depth, while the ApKF does so less well.

Moreover, the computational cost of observer updates are displayed in Table 4, for our simulations with $N = 1034$ and for a larger scale simulation with $N = 4560$. This table clearly shows the increase in efficiency in the enhanced filtering algorithms with respect to the traditional KF. Furthermore, when comparing results for $N = 1034$ and $N = 4560$, it is noted that gains in computational efficiency scale up more than linearly with system size.

Table 4: Average computational cost per iteration in seconds

System size	**WFSim**	**KF**	**EnKF**	**ApKF**
$N = 1034$	$1.0 \cdot 10^{-1}$	$1.3 \cdot 10^{1}$	$7.0 \cdot 10^{-1}$	$8.0 \cdot 10^{-1}$
$N = 4560$	$1.2 \cdot 10^{0}$	$1.1 \cdot 10^{3}$	$8.0 \cdot 10^{0}$	$8.5 \cdot 10^{0}$

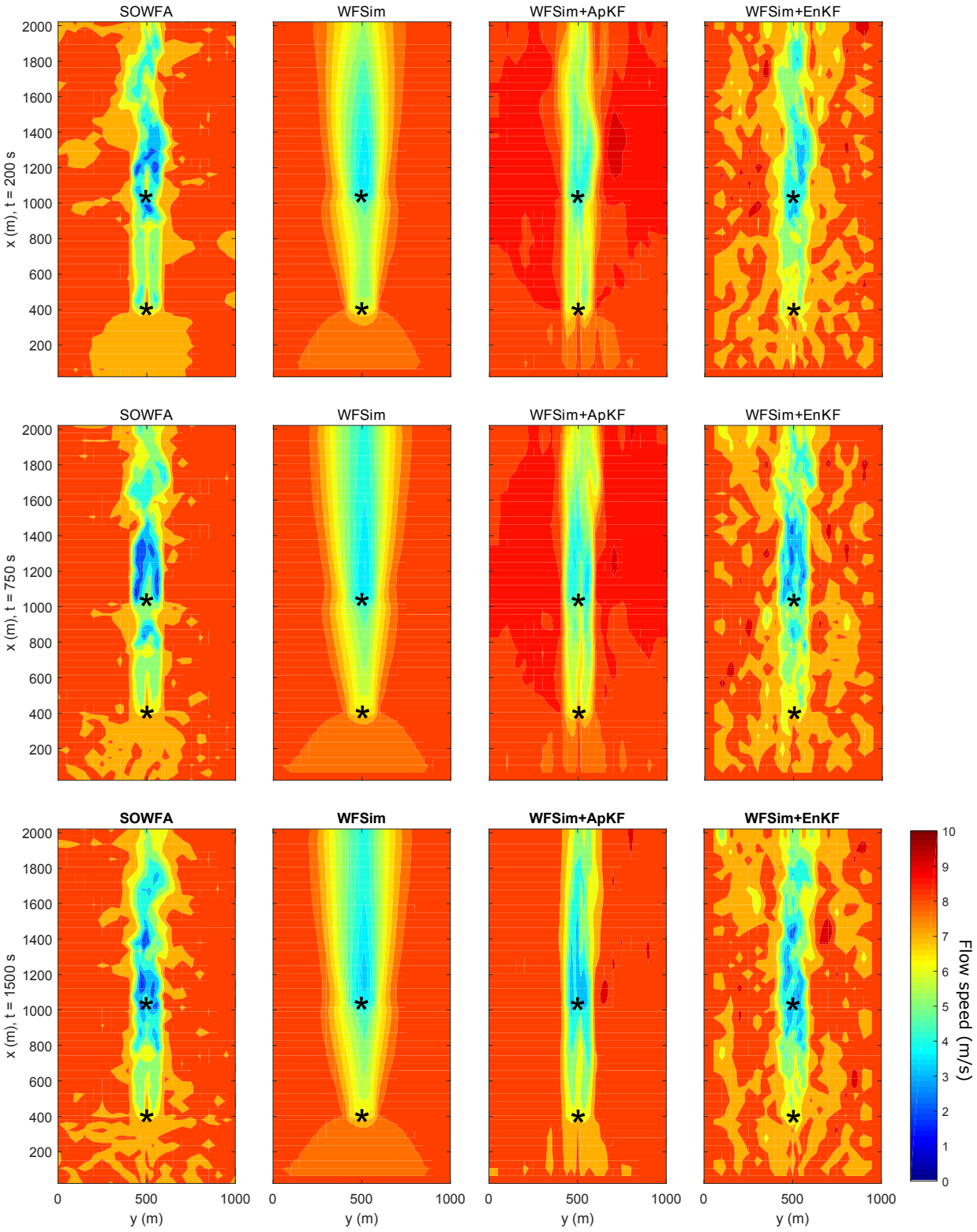


Figure 8: Estimated flow field in x-direction for $t = 200, 750, 1500$ s. Windflow is from bottom to top, with the color bar defining the wind speeds in m/s. Turbines are denoted by an asterisk. This figure shows the improvements on flow field estimations the ApKF and the EnKF make compared to the case without observer. Notably, the observers account for an overestimated wake width, underestimated wake recovery, wake meandering, and turbine hub effects.

6. Conclusions and on-going work

Concluding from Figures 7 and 8 and Tables 3 and 4, two state observer designs have been proposed with good performance at a fraction of the computation cost of a full KF. These computational benefits grow more than linearly with system size. In our simulations with $N = 1034$ states, the observers account for an overestimated wake width, underestimated wake depth, underestimated wake recovery, turbine hub effects, and wake meandering. This is a first major step to time-efficient higher fidelity closed-loop control of wind farms.

The next major step in this research is increasing the degree of realism in simulation – the number of measurements should be significantly decreased to be comparable to what can be achieved in practice (e.g., with the use of lidar). Furthermore, SOWFA simulations are to include more turbulence and different wind farm topologies. Also, improvements in model validation and flow field estimation can be achieved by improving the WFSim model (e.g., including turbine hub effects or improving the turbulence model), refined observer tuning, and using refined domain meshings. Finally, as the focus in this work is limited to axial-induction-based control, wake redirection control is to be explored in future research.

Acknowledgments

This work would not have been possible without the support of the National Renewable Energy Laboratory (NREL) and the University of Colorado Boulder (CU Boulder). Furthermore, we thank Matthias Morzfeld from the University of Arizona for coming up with the initial idea of applying an Ensemble Kalman filter.

References

[1] Goit J and Meyers J 2015 Optimal control of energy extraction in wind-farm boundary layers *Journal of Fluid Mechanics* **768** 5–50

[2] Fleming P A, Ning A, Gebraad P M O and Dykes K 2015 Wind plant system engineering through optimization of layout and yaw control *Wind Energy* **19**(2) 329–44

[3] Gebraad P M O 2014 *Data-driven wind plant control* Ph.D. thesis Delft University of Technology URL `http://repository.tudelft.nl/islandora/object/uuid%3A5c37b2d7-c2da-4457-bff9-f6fd27fe8767`

[4] Fleming P A, Aho J, Gebraad P M O, Pao L Y and Zhang Y 2016 Simulation study of active power control in wind plants *American Control Conference*

[5] Ela E *et al.* 2014 Active power controls from wind power: Bridging the gaps Tech. Rep. NREL/TP-5D00-60574 National Renewable Energy Laboratory

[6] Boersma S, van Wingerden J W, Vali M and Kühn M 2016 Quasi linear parameter varying modeling for wind farm control using the 2D Navier Stokes equations *American Control Conference*

[7] Torres P, van Wingerden J W and Verhaegen M 2011 Modeling of the flow in wind farms for total power optimization *Control and Automation* **9** 963–8

[8] Doekemeijer B M 2016 *Enhanced Kalman filtering for a 2D CFD Navier-Stokes wind farm model* MSc thesis Delft University of Technology URL `http://repository.tudelft.nl/islandora/object/uuid%3A5aeb5ec3-aae2-4cbd-b9ea-17d88742cf0d`

[9] Versteeg H K and Malalasekera W 2007 *An introduction to Computational Fluid Dynamics: The Finite Volume Method* 2nd ed (Edinburgh Gate, England: Pearson Education Limited) ISBN 978-0-13-127498-3

[10] Bianchi F D, de Battista H and Mantz R J 2007 *Wind Turbine Control Systems: Principles, Modeling and Gain Scheduling Design* 1st ed (London, United Kingdom: Springer) ISBN 978-1-84-628492-2

[11] Prandtl L 1925 Über die ausgebildete turbulenz *Journal of Fluid Mechanics* **5** 136–9

[12] Ishihara J Y, Terra M H and Bianco A F 2010 Recursive linear estimation for general discrete-time descriptor systems *Automatica* **46**(4) 761–6

[13] Ishihara J Y and Terra M H 2008 Robust state prediction for descriptor systems *Automatica* **44**(8) 2185–90

[14] Cuthill E and McKee J 1969 Reducing the bandwidth of sparse symmetric matrices *Proceedings of the 1969 24th National Conference* ACM '69 (New York, United States: ACM) pp 157–72

[15] George A and Liu J W 1981 *Computer Solution of Large Sparse Positive Definite Systems* (Englewood Cliffs, United States: Prentice Hall) ISBN 978-0-13-165274-3

[16] Evensen G 2003 The ensemble kalman filter: theoretical formulation and practical implementation *Ocean Dynamics* **53**(4) 343–367

[17] Churchfield M and Lee S 2015 SOWFA | NWTC information portal URL `https://nwtc.nrel.gov/SOWFA`

[18] Fleming P A, Gebraad P M O, Lee S, van Wingerden J W, Johnson K, Churchfield M, Michalakes J, Spalart P and Moriarty P 2013 High-fidelity simulation comparison of wake mitigation control strategies for a two-turbine case *International Conference on Aerodynamics of Offshore Wind Energy Systems and Wakes* (Lyngby, Denmark)
[19] Annoni J, Seiler P, Johnson K, Fleming P and Gebraad P 2014 Evaluating wake models for wind farm control *American Control Conference* 2517–23
[20] Jonkman J M and Buhl M L 2005 FAST v6.0 user's guide Tech. Rep. NREL/EL-500-3823 National Renewable Energy Laboratory
[21] Fleming P A, Gebraad P M O, Lee S, van Wingerden J W, Johnson K, Churchfield M, Michalakes J, Spalart P and Moriarty P 2014 Evaluating techniques for redirecting turbine wakes using SOWFA *Renewable Energy* **70** 211–8
[22] Fleming P A, Gebraad P M O, Lee S, Churchfield M, Scholbrock A, Michalakes J, Johnson K and Moriarty P 2013 SOWFA super-controller: a high fidelity tool for evaluating wind plant control approaches *Conference Proceedings of EWEA* (Vienna, Austria)
[23] Churchfield M J, Lee S, Michalakes J and Moriarty P J 2012 A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics *Journal of Turbulence* **13** 1–32
[24] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-MW reference wind turbine for offshore system development Tech. Rep. NREL/TP-500-38060 National Renewable Energy Laboratory
[25] Courtney M, Wagner R and Lindelöw P 2008 Commercial lidar profilers for wind energy: a comparative guide *Conference Proceedings of EWEC* (Brussels, Belgium)