Using particle filters to track wind turbine wakes for improved wind plant controls

P. A. Fleming\textsuperscript{1}, P. M. O. Gebraad\textsuperscript{2}, M. J. Churchfield\textsuperscript{1}, J. W. van Wingerden\textsuperscript{2}, A. K. Scholbrock\textsuperscript{1}, P. J. Moriarty\textsuperscript{1}

\textbf{Abstract}—Recently there has been interest in the design of wind farm control systems that can coordinate individual turbine controllers to improve global plant performance. This improvement comes from accounting for the way in which turbines interact through wakes. Often however, controllers are designed assuming steady and known environmental conditions, without turbulence or wake meandering. This raises the concern that these methods will fail to perform well in practice because it could be difficult to apply methods based on steady wakes to a situation where wake locations are changing and not measurable. In this paper, a particle filter is used to continually estimate the wake locations in a stochastic setting by combining all of the available turbine measurements. The design of the algorithm is documented, and is shown to employ sensors that are available on modern turbines. Using a high-fidelity wind farm simulator, we show the effectiveness of the proposed framework using several multi-turbine scenarios and compare the wake locations predicted against the wakes observable in flow-field slices taken from the simulator output.

\textbf{I. INTRODUCTION}

There has been growing interest in the design of wind plant control systems to coordinate the controls of individual turbines to achieve improvements in the overall wind plant performance. In these proposed methods, individual turbine controls are adjusted to improve the global output of the plant above what would be achieved if each turbine pursued its individual optimal output. In general, these methods look to improve performance by accounting for the way turbines interact in a plant through their wakes, which has been shown to negatively impact performance [1].

Within the literature of wind plant controls, there are several proposed approaches using different wind turbine control actuation. One approach modifies the power capture level, or axial induction, of individual turbines when multiple turbines are aligned in the inflow direction. One example of this approach is [2], which uses a learning approach to obtain optimal axial induction settings. Other examples in the literature include [3], [4].

An alternative method uses intentional yaw misalignment of the upstream turbine in a waked configuration to deflect the wake away from a downstream turbine. This method is explored in [5], [6], [7].

These approaches show promise to improve wind plant performance, both in terms of overall power capture as well as reducing the loads experienced by individual turbines. However, studies to date are based on simulations and assume steady and known conditions. Reference [5] is the exception, employing a scaled test site; however, the experiment is limited to fixed yaw positions and a specific known wind direction.

One practical issue in bringing these methods to physical wind plants is the complexity introduced by the real environment. The wind direction is continually changing, and turbine wakes will meander as they propagate downstream. This introduces a problem for techniques that expect to have known information about the wake locations. For example, the game theoretic axial induction methods need to compare past and current performance under similarly waked conditions in order to learn correctly. Additionally, the wake redirection approaches use information about the current wake locations in order to be able to make decisions on how to re-direct the wake.

Therefore, what would be beneficial is a method to track turbine wakes within a wind plant by combining the local measurements of individual turbines. This is made difficult by the changing and stochastic nature of the environment, as well as the imperfect and noisy measurements produced by the turbine sensors (typically vane and nacelle anemometers). Additionally, to be practically useful, the method should perform in a reasonable amount of time if it is to be used with active wind turbine controllers.

In this paper, we propose a method for tracking the locations of turbine wakes within a wind plant based on the particle filter algorithm. Particle filters, as will be discussed later, have properties that make them a good match to this challenge. We will design the structure of individual particles, and present the method for importance weighting of the particles. Finally, using high-fidelity simulations of wind plants, we will evaluate the ability of particle filters to track wake locations.

The output of this paper will be an approach for tracking wind turbine wakes using turbine sensor measurements that
are commonly available on commercial wind turbines. The output of this algorithm, tracked wake locations, could then be coupled with wind plant control techniques, such as modified axial induction or wake redirection, to form a complete closed-loop control system within a stochastic and unknown environment.

The remainder of this paper is organized as follows. Section II provides a brief background on wind turbine wakes and on particle filters. Next, Section III documents the design of the particle filter for tracking wind turbine wakes. Section IV presents the results of the high-fidelity simulations for testing the performance of the designed particle filter. These results are discussed in Section V and possible improvements are presented. Finally, conclusions are given in Section VI.

II. BACKGROUND

In this section, we first review wind turbine wakes and then particle filters.

A. Wind Turbine Wakes

Wind turbines extract energy from the kinetic energy of the wind flowing through a turbine’s rotor. This results in a reduced wind speed downstream of the turbine, creating a wake. When wind turbines are sited near each other in a wind plant, conditions can arise where one turbine is in the wake of another. This produces the situation where the downstream turbine will underperform, motivating the study of wind plant control described earlier.

The trajectory of the wake as it propagates downstream is a topic of recent research. Because the wake is in a turbulent three-dimensional wind flow, the wake will move in transverse directions in addition to the dominant wind flow direction [8]. Given this effect, the location of the wake at a position downstream is challenging to predict given only measurements taken at the turbines. In many cases, with limited information, a probable range of wake locations is the best that can be determined. Particle filters therefore represent a good approach for attempting to track turbine wake locations.

B. Particle Filters

A particle filter is a method for predicting the state of a partially observable Markov chain in discrete time [9]. The central idea of particle filters is to represent the probability density function of the state as a set of samples or particles [10]. In the literature, there are various implementations of particle filters; the version used in this paper will be adapted from those developed for robotics and documented in [9], [10], [11].

In the algorithm used in this paper, the particle filter is composed of a collection of particles. Each particle consists of a state variable and an importance weight. The particle state represents a hypothesis of the state of the world [9]. The importance weight is determined by comparison of the state with available observations and could be thought of as a likelihood that this particle is correct given the current observations. The full set of weighted particles can be used to reconstruct the underlying probability density function.

Each iteration of the particle filter includes:

1) Perturbation: Every particle is perturbed by Gaussian noise.
2) Weight Updating: The importance weights of the particles are recomputed given the latest observations.
3) Resampling: Using the weights, a new set of particles is chosen from the existing set by choosing with replacement according to the particle weights.

The above process is repeated at each time step. The perturbation step is often called the prediction step in robotics when it combines known control actuation with noise; however, in this work we assume only unknown forcing. Weight updating represents most of the specific design for this application and will be discussed in depth in a later section. Resampling is the act of choosing a new set of particles from the currently weighted sample. The main idea is to remove from consideration the least weighted particles and focus on the highest weighted particles. This causes the particle filter to concentrate the search in the most likely sectors of the space and helps the particle filter operate with a limited number of particles.

III. PARTICLE FILTER FOR WAKE TRACKING

Particle filters, as described above, are very good candidates for the challenge of tracking wind turbine wakes for use in wind plant controllers. First, the method is inherently probabilistic; rather than searching for an “optimal” fit, the filter searches across all likely spaces. Similarly, the particle filter is not uni-modal, that is, it can accommodate separate clusters of probability. This is useful, for example in the case when a wake could be in several noncontinuous locations (e.g., on either side of a turbine). Finally, particle filters are “anytime,” meaning that the accuracy of the filter can be adjusted to meet available resources (e.g., by changing the number of particles) [11].

In this section, we will document the design of a particle for the application of tracking wind turbine wakes. First we will present the structure of the particle state, and then the method of importance weighting a particle given turbine measurements.

A. Particle State Structure

For the application of particle filters for wake tracking, we propose a structure in which a single particle is an estimate of the trajectory of the wake for each turbine in the plant. The particle filter is composed of N particles. The measurement is the combined sensor data from all turbines, which is used to assign a weight to each particle. This concept is illustrated in Fig. 1.

Having defined that the particle filter is composed of many particles and each particle is composed of one wake trajectory per turbine, we now define the structure of a single wake trajectory. This is illustrated in Fig. 2.

Fig. 2 shows that a single trajectory is composed of several joining line segments. The trajectory is defined by
A single particle includes one wake trajectory per turbine. The particle filter includes many different individual particles. At each time step, the particles are perturbed, weighted and resampled and a single representative particle is obtained. Particles tend to cluster in the most highly weighted regions.

Fig. 1: Wake tracking with particle filters.

(a) A single particle includes one wake trajectory per turbine
(b) The particle filter includes many different individual particles
(c) At each time step, the particles are perturbed, weighted and resampled and a single representative particle is obtained. Particles tend to cluster in the most highly weighted regions.

Fig. 2: Illustrating the values that define a single wake trajectory. A particle includes one trajectory per turbine. The variables are defined in Table I.

several values, some of which are unchanging, such as the number of segments and segment length. These are uniform across all trajectories. Others variables are adjusted in each perturbation step: the initial angle of the trajectory and the angles between segments. Finally, values such as the angle between each segment and the horizontal, are calculated values and used in weighting a particle. The list of values is provided in Table I.

B. Perturbation

At each time step, each particle is perturbed to generate a new set of set of particles. Currently, this perturbation is applied to both \( \theta_{\text{init}} \) and to each of the joint angles \( \alpha_{1,2,...} \). The amount of perturbation is chosen from a normal distribution with a standard distribution of 2 degrees, which was selected experimentally.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>( L_{\text{seg}} )</td>
<td>Segment length</td>
</tr>
<tr>
<td></td>
<td>( N_{\text{seg}} )</td>
<td>Number of segments</td>
</tr>
<tr>
<td>Variable</td>
<td>( \alpha_{1,2,...,N_{\text{seg}}-1} )</td>
<td>Angle between segments</td>
</tr>
<tr>
<td>Derived</td>
<td>( \theta_{\text{init}} )</td>
<td>Initial angle between wake and horizontal</td>
</tr>
<tr>
<td></td>
<td>( \theta_{1,2,...,N_{\text{seg}}-1} )</td>
<td>Angle at joint to horizontal</td>
</tr>
<tr>
<td></td>
<td>( d_{\text{int}} )</td>
<td>Intersection distance</td>
</tr>
</tbody>
</table>

C. Weighting

Weighting of the particles is done by comparing the particle state with available measurements from the wind turbines. We principally assume available sensor measurements include wind speed and direction at each turbine via an anemometer and vane. This is true for a typical commercial turbine. As an extra sensor, we make use of blade root strain measurements. While this sensor is not completely unusual, it is also not always present. However, the particle filter would function without it; the extra information is used to fine tune the results by providing an indication of which side of the rotor a wake is impacting.

Combining these sensors provides several ways in which the probability of a given particle being correct can be assessed. The weighting strategy pursued here defines separate subweights. Each one is a “penalty”—a larger weight is assigned for a less likely particle. When each subweight is computed, the weights are totaled after being scaled by a scaling parameter that allows a trade-off of the importance of each subweight. Selection of these scaling parameters is done manually, primarily to assure that the range of each subweight is comparable such that no single subweight dominates. The total weight is inverted in the last step, so that finally the largest weighted particle is the most likely, which is the normal ordering.

In the following sections we review the separate subweight
computations. Note that when a weighting function applies to a single wake trajectory, the total subweight value is the sum of the computed penalty for each trajectory in the particle.

1) Deflection: This first subweight is meant to penalize an initial wake angle ($\theta_{\text{init}}$), which conflicts with assumptions about the relationship between yaw misalignment and the deflection of the wake with respect to the inflow direction. In the literature, simulation and experimental studies of this effect exist, see for example [5], [7], [12]. These studies seek to determine a relationship between yaw misalignment and the deflection of the wake based on turbine properties, such as the thrust surface. The amount of deflection is still a topic of research. It is turbine-specific, and will vary depending on properties of the turbine. The direction of the deflection is consistent across turbines, however, and Fig. 3 illustrates this relationship.

Fig. 3: Illustrating the relationship between yaw misalignment and wake deflection.

In Fig. 3, $\theta_{\text{inflow}}$ is the angle of the inflow angle with respect to 0, while $\theta_{\text{NacErr}}$ is the angle between the yaw alignment of the turbine with respect to the inflow direction. $\theta_{\text{Init}}$ is the initial angle of a given particle with respect to 0. Finally, $\theta_{\text{Deflect}}$ is the angle between $\theta_{\text{Init}}$ and $\theta_{\text{inflow}}$ and represents the particle’s prediction of the deflection angle. Based on the literature, we could assign a weighting based on a model-predicted $\theta_{\text{Deflect}}$; however, the most generic weighting is to simply apply no penalty if the deflection is in the correct angle, and apply an increasing penalty when the deflection is in the incorrect direction. A final stipulation is that if $\theta_{\text{NacErr}}$ is somewhat small, no weighting should be applied in any case as the deflection would probably be small relative to random variation. $\theta_{\text{NacErr}}$ is measured by individual turbines, and $\theta_{\text{inflow}}$ is computed by combining $\theta_{\text{NacErr}}$ with the known yaw angle of the upstream turbines. Both signals are filtered using a recursive filter with 2 s time constant to reduce the effects of noise. This yields the following weighting algorithm:

$$W_{\text{Deflect}} = \begin{cases} 0 & \text{if } \text{abs}(\theta_{\text{NacErr}} < 5) \\ \text{else if } \text{sign}(\theta_{\text{NacErr}}) \neq \text{sign}(\theta_{\text{Deflect}}) & \text{then} \\ \text{else} & \text{if} \text{sign}(\theta_{\text{NacErr}}) \neq \text{sign}(\theta_{\text{Deflect}}) & \text{then} \\ \text{end if} \\ \text{end if} \\ \end{cases}$$

Fig. 4: Weighting functions applied in waked and unwaked cases. Note that for ambiguous weighting, weight is always 0. $\Delta\text{Speed}$ indicates the difference between the estimated inflow speed and the local wind speed measurement.

Given this, the value $d_{\text{Int}}$ for a given particle and given downstream turbine is defined as the distance along the rotor to the nearest intersection of a wake trajectory along the rotor plane, as shown in Fig. 2. A subweight scheme can be applied that assumes that if the turbine is waked, the intersection should be within the turbine radius ($R$). Similarly, if the turbine is not waked, that is the local velocity is close to the free-stream velocity, as determined by the upstream turbines, the $d_{\text{Int}}$ should be ideally larger than $R$. Finally, if the differential between the local inflow speed is such that waking is ambiguous, no weighting is applied.
For the waked and unwaked cases, the per-turbine weighting scheme for intersection is illustrated in Fig. 4. Note that the amount of weighting is conditioned by the degree of waking or nonwaking observed (represented by the delta speed value between local inflow and plant inflow computed by the upstream turbines). The intersection penalty is computed for each turbine \( W_{\text{Int}(\text{Turb})} \), and the sum across all turbines is the intersection penalty \( W_{\text{Int}} \).

4) Interaction location: This subweighting makes use of blade root strain gauge measurements. As mentioned, such sensors are not completely unusual, but are not standard, and so this subweighting is limited to capable turbines. However, when available, all blade measurements are combined to compute a yawing moment observed at the rotor that gives an indication of which side of the rotor the wake is more likely to be impacting. This in turn can be used to penalize particles with wake intersections on the side of a rotor less likely to be correct. The subweight is zero if a turbine has no intersecting trajectories or if the nearest trajectory is on the correct side. If the nearest intersecting trajectory is on the wrong side, then a subweight is applied maximally when the wake is centered at the radius and dropping off in both directions. As for the intersection penalty, the interaction location penalty \( W_{\text{IntLoc}} \) is the sum of the penalties computed for each turbine.

5) Straightness and smoothness: Two final weights penalize particle trajectories that are not straight:

\[
W_{\text{straight}} = \sum_{i=1}^{N_{\text{seg}}-1} \text{abs}(\alpha_i)
\]  

And trajectories that are not smooth:

\[
W_{\text{smooth}} = \sum_{i=2}^{N_{\text{seg}}} \text{abs}(\alpha_i - \alpha_{i-1})
\]

In general, these weights are given lower priority than the ones based on sensor measurements. The purpose of these weights is, other factors being equal, a straighter and smoother wake trajectory is probably more likely.

D. Representative Particle

The final step performed is the selection of a single “representative particle” from the full set of particles. This could be done in several ways: selecting the highest weighted particle and computing a weighted average are two common solutions. However, in this application, the highest weighted particle can lead to too noisy of a signal as the highest weighted particle might jump a bit from step to step. Weighted averaging is also problematic, because in a multimodal case, such as trajectories clustered on either side of a downstream turbine, the average might split the difference and lead to a very low probability estimate. A compromise solution, from [10], suggests finding the highest weighted particle and then averaging the k-nearest particles. Using k as 10% of the total number of particles, this approach has performed well in testing and is used in this paper.

IV. Testing

Given a structure for the particles, and a method for weighting particles by sensor measurement, it is now possible to test the algorithm. In order to do this, we use a high-fidelity wind plant simulation tool developed at the National Renewable Energy Laboratory (NREL): the Simulator for Onshore/Offshore Wind Farm Applications (SOWFA). Within SOWFA, we run several wind plant scenarios and use the recorded turbine measurements (such as nacelle wind speed, wind direction, and blade bending) to run the particle filtering algorithm designed above. Afterward, we can extract horizontal slices from the flow-field at hub height, in which turbine wakes are clearly visible as a way to evaluate the performance of the technique. In this section we first discuss the SOWFA tool, and then review some experimental cases.

A. SOWFA

SOWFA is a computational fluid dynamics (CFD) tool used to model wind turbines in a flow field. It couples NREL’s FAST turbine modeling tool [13], with a CFD solver based on the OpenFOAM toolbox [14]. The CFD solver uses a large-eddy simulation method to resolve the larger turbulent scales to simulate the atmospheric boundary layer where wind turbines are located. This flow is first created using the CFD solver alone to generate a free-flowing field without the influence of the wind turbines. Once this is done, the inflow is then saved to be used with the wind turbines. The wind turbines are modeled using an actuator line technique [15] coupled with FAST, where a rotating model of the wind turbine’s rotor is used to create time-dependent forces in the fluid that generate wakes that interact with each other as well as with the flow itself. Each blade is represented by a line broken into segments. Each segment has a known airfoil type, twist angle, and chord length. The velocity from the flow field is then used as a local inflow to the blade segment, and corresponding lift and drag tables are then used to determine the force vector at each segment. This force vector is then projected onto the flow-field as volumetric body forces to model the turbine’s interaction with the wind flow. For control, each turbine can operate an individual controller, and an overarching wind plant controller can also be implemented [16]. More details about SOWFA are explained in [17]. For producing measurements, the nacelle wind speed and direction measurements are captured from the simulation via a probe in the flow located at each turbine nacelle, and the blade-bending is returned by each turbine FAST instance.

B. Simulated Experiment Setup

The particle filtering technique is applied to a wind farm consisting of two rows with three NREL 5-MW baseline turbines, with a 5-rotor-diameter spacing in the down-wind direction, and 3 rotor diameters in the cross-wind direction. This setup is placed in a 3-km (length) by 3-km (width) by 1-km (altitude) mesh. The smallest mesh cells, which contain the turbine rotors, the axial induction zones of the rotor, and the wakes between the turbines, have a size of
3mx3mx3m. Farther away from the turbines, the mesh is coarsened to 6mx6mx6m cells, then to 12mx12mx12m cells, resulting in a total of $32 \cdot 10^6$ cells. Using a time step of 0.02 s, a 1000 s simulation is performed. This simulation will take 59 hours to perform using distributed computation with 512 processors. The setup is subjected to a turbulent inflow with a 6% turbulent intensity and an 8 m/s mean velocity at hub height.

Fig. 5 shows the results of a case where the wind turbine rows are aligned with the mean $300^\circ$ flow direction. The mean flow direction for these simulations is nonvarying. Further, there are cases where the rows are rotated $5^\circ$, $10^\circ$, and $15^\circ$ relative to the wind direction.

C. Simulation Results

The designed particle filter was applied first to the case discussed in Section IV-B in which the turbine rows are aligned with the flow. Some of the turbines, however, have been yawed in a way to redirect their wakes. The results from the particle filter at several time steps are overlaid on the flow-field in Fig. 5. Note that although the simulation is run with a time step of 0.02 s, the particle filter is run at 1 s given the time scales of wake propagation and meandering. The figure shows the evolution of the particle filter with time in the middle row, as well as the selected representative particle for each time step in the bottom row. The correspondence with the wake location is good, and the particle filter appears able to cover the range of possible locations with particles, as well as choose a reasonable representative particle.

The middle row of Fig. 5 shows that for the turbine whose wake impacts a downstream turbine, it is possible to narrow the search space a bit whereas turbines at the end of the row have more uncertainty as to the final wake location. This difference in certainty relating to whether or not a downstream turbine is present impacts the correctness of the selected representative particle in the bottom row.

Fig. 6 is a single time step from a simulation where the turbine rows are aligned 10 degrees off the inflow
direction, leading to partial overlap only. This figure also gives insight into how the particle filter works. Notice that the trajectories of turbine 4 are more compactly spaced than turbine 3. Turbine 6 is currently unwaked, and the range of probable trajectories is larger given that no turbines are downwind of turbine 6. As a final point, note that among the trajectories of turbines 1, 2, 3, and 4, the vast majority go “below” the respective downstream turbine, while a few go “above” (technically south and north, respectively). This is an excellent feature of particle filters: the most effort is placed in the highest probable (and it turns out correct region) of the space, but some small search is made of another improbable, but not impossible, space (trajectories here would explain the lack of downstream waking, but require unlikely deflection angles and prolonged cross-stream propagation).

V. DISCUSSION

The results from applying particle filters to the SOWFA simulations are promising. The particle filter is generally accurate in its representative prediction, and the coverage of particles across probable areas for trajectories is encouraging. Further, Fig. 6 is based on a setup using 200 particles and runs at about real time. This is with unoptimized code in MATLAB and could surely be improved. However, given the long time scales of wake propagation, it is encouraging that the algorithm can be brought close to real time with little effort.

That said, it is important to note the shortcomings of the current work and how it can be improved in the future. For one thing, the current implementation assumes that the wake can only move in a two-dimensional plane when actually it moves in three (c.f. [18], [12]). Future work may generalize for three dimensions. More generally, the particle filter as designed could be optimized for efficiency and accuracy. The current settings of subweight scalings, resampling intervals, and particle sizes represent a manual tuning process, however, future work should more rigorously determine good values.

Additionally, there is currently only qualitative analysis of the correctness of the wake centers produced by the algorithm. In the future, it could be helpful to make quantitative comparisons with algorithms for detecting wake centers from slices of CFD data (as was done in [12]), or with wakes predicted by static wake direction models (for example, the wake model from [7]). Simulations in which the mean wind direction varies are also the subject of future work. In addition, comparison with high-fidelity simulation is a good start, but it would be more interesting to compare with field tests in which the flow is measured in some way for comparison, for example by a scanning lidar. Finally, as described in the introduction, tracking the wakes represents a first step; controlling the wake in closed-loop control completes the process. Future work will develop such controls and assess their potential for improved wind plant performance, as well as investigate possible issues such as closed-loop stability.
VI. CONCLUSIONS

In this work, we have presented a design for a particle filter that tracks wind turbine wakes in a wind plant using available turbine sensors. The method is shown to be a good fit to the problem because of the desirable properties of particle filters, and the stochastic approach is appropriate for the wake tracking application. High-fidelity simulations were used to test the approach, and inspection of the results shows good performance of the designed particle filter for the cases simulated. Future work will continue to develop this approach as well as design controllers that make use of the wake location estimates. We believe these controllers could also be formulated in a stochastic framework.

ACKNOWLEDGMENTS

The authors thank Kevin Regimbal, Wesley Jones and the NREL High Performance Computing team for their critical and timely help in running the high-fidelity simulations for this work.

REFERENCES