Received: 19 September 2022 | Revised: 31 August 2023 | Accepted: 2 November 2023

DOI: 10.1002/we.2884

SPECIAL ISSUE PAPER

WILEY

A dynamic model of wind turbine yaw for active farm control

Genevieve M. Starke[1,2] | Charles Meneveau[2] | Jennifer R. King[1] | Dennice F. Gayme[2]

[1]National Wind Technology Center, National Renewable Energy Laboratory, Golden, Colorado, USA

[2]Department of Mechanical Engineering, Johns Hopkins University, Baltimore, Maryland, USA

Correspondence
Genevieve M. Starke, National Wind Technology Center, National Renewable Energy Laboratory, Golden, CO 80401, USA.
Email: genevieve.starke@nrel.gov

Funding information
National Science Foundation, Grant/Award Numbers: CBET-1949778, CMMI-1635430, DGE-1746891, OAC 1920103; US Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office

Abstract

This paper presents a graph-based dynamic yaw model to predict the dynamic response of the hub-height velocities and the power of a wind farm to a change in yaw. The model builds on previous work where the turbines define the nodes of the graph and the edges represent the interactions between turbines. Advances associated with the dynamic yaw model include a novel analytical description of the deformation of wind turbine wakes under yaw to represent the velocity deficits and a more accurate representation of the interturbine travel time of wakes. The accuracy of the model is improved by coupling it with time- and space-dependent estimates of the wind farm inflow based on real-time data from the wind farm. The model is validated both statically and dynamically using large-eddy simulations. An application of the model is presented that incorporates the model into an optimal control loop to control the farm power output.

KEYWORDS
controls oriented modeling, dynamic yaw, wind energy, wind farm modeling, wind turbine yaw, yaw control

1 | INTRODUCTION

Yawing of individual turbines within a windfarm has the potential to increase overall power output by deflecting their wakes from the inlet of downstream turbines. This has motivated a wide range of models aimed at better predicting the wake behind a yawing wind turbine.[1–7] These models provide increasingly accurate representations of flow due to static wind turbine yaw and have been widely used to study yaw configurations that increase power output for given wind conditions. Simulations,[8,9] experiments,[10,11] and, more recently, field tests [4,12–14] examining the potential and effects of wake steering to increase the power output of the farm have provided further evidence of the usefulness of yaw optimization for a wind farm in steady state. While these studies provide important insights into the potential of yaw, they have largely been performed in a static context and therefore neglect the transient response, which can impact the overall power output potential over a given time horizon. In addition, the existing suite of static models is not built to capture the dynamic response of a wind farm to a change in yaw, which limits the development of active control approaches under changing conditions.

The time scale of yaw changes is measured in minutes, which can be slower than interturbine travel time and therefore affect the behavior of the overall power output of the farm. In fact, flow-through times for large wind farms can be multiple minutes, which is of similar order to the yaw action. As wind power continues to increase, accuracy in predicting the transient response and trajectory of the wind farm power output is of increasing importance, particularly as wind farms are asked to provide grid services or operate in the same manner as conventional generators (i.e., follow a demand curve rather than providing all the power available).

Recognizing the importance of the time evolution of farm power output as turbines yaw for our overall understanding of the potential of yawing in farm operation and control has motivated recent efforts to account for time-dependent changes due to yaw. The formulation outlined in Shapiro et al[1] can represent dynamic changes, as it is based on a partial differential equation, although the dynamic version of that model has not yet been fully validated. The FLOw Redirection and Induction in Steady State (FLORIS) model from the National Renewable Energy Laboratory (NREL) has been adjusted to take dynamic changes into account between turbines by implementing time delays in the travel of information between turbines.[15] A similar approach is also employed by Howland et al[16] along with an ensemble Kalman filter to incorporate measurements.

While these approaches take steps toward addressing the need for dynamic models, they miss some of the important physics in the wake dynamics of a yawed turbine. For example, none of the aforementioned models include the known wake shape deformation caused by the turbine yaw. This deformation[17] causes noncircular cross sections (curled wake) that when impinging on downstream turbines can significantly affect the average velocity[6] and thus affect power output. Including such subtle fluid mechanical effects in the context of fast and easy-to-evaluate models for turbine interactions due to wakes is quite challenging. To address this challenge, this paper combines the graph-based modeling approach that was used to model a wind farm under changing wind direction in Starke et al[18] with physical models that incorporate important features of the deformed wake structure due to yaw[6] as well as time-dependent propagation.

The graph-based approach in Starke et al[18] built upon prior work that introduced the idea of representing a wind farm as a graph.[19,20] As in these works, nodes comprised the individual turbines, and the edges were defined by the wake interactions between turbine pairs. However, rather than assuming a static graph, this approach allowed for changes in graph topology by introducing a time-varying graph framework with edge switching.[21] This enabled the resulting dynamic graph model to capture the transient response of the wind farm as the wind inlet direction changed. In this paper, that graph-based modeling approach is adapted to incorporate the effect of dynamic turbine yawing on a wind farm.

The dynamic yaw model incorporates several key extensions over the original formulation that enables it to accurately capture the time evolution of the wind farm power output as individual turbines dynamically yaw. The model states that capture the wake interactions are redefined using a novel physics-based analytical representation of wake deformation due to yaw.[6] This addition includes the propagation of yaw effects through the farm and the effect of yaw on the power of the turbines. Additionally, the time delay formulation was adjusted to incorporate the effects of slower velocity due to the wakes of upstream turbines in the farm.

While the model includes a more advanced representation of the wake shape, the model also includes simplification of the physics to increase computational efficiency and decrease run time in order to facilitate its use in real-time control applications. A necessary trade-off in such simplified models is less accuracy versus high-fidelity approaches, for example, using a large-eddy simulation (LES) model. Data enhancements have been shown to improve the prediction of these models.[16,22] In this work, we focus on the use of data to enable the model to represent heterogeneous velocity profiles in both the streamwise and spanwise flow directions in time. This particular choice is motivated by the significant changes in the inflow conditions across wind farms, which typically span a distance of kilometers. Including these variations in the model, we can better capture local conditions in the inflow that will propagate through the farm and affect wake advection and the power output at each turbine. The ability to capture these local effects was shown to improve predictions even in traditional wake models.[14,23–25]

The final data-augmented model provides a low-order representation that captures both the transient response and the overall final power level due to yaw changes that are emphasized in static approaches to maximizing power output through yaw angle optimization. We demonstrate the utility of the dynamic yaw model in two examples of model-based control of the power output of a wind farm. These applications confirm that the dynamic yaw model accurately predicts the transient response of the wind farm to yaw actions and within an active yaw control loop. In the accompanying discussion, we describe some potential limitations of the approach and power-tracking control with only yaw-based control.

The remainder of this paper is organized as follows: In Section 2, we present the dynamic yaw model formulation, explaining how the approach enables efficient reduced-order modeling of pairwise interactions between only those turbines that are related to each other via wake interactions. We also describe the inclusion of yaw-induced curled wake velocity distributions and deflection. In Section 3, we validate the model both statically and dynamically using large eddy simulation (LES). In Section 4, we present a specific example of the application of the model where yawing is optimized to follow a prescribed power signal for a wind farm to demonstrate the potential use of the model. Finally, Section 5 describes some conclusions and directions of ongoing work.

2 | THE DYNAMIC YAW MODEL

In this section, we formulate the dynamic yaw model. Sections 2.1 and 2.2, respectively, define the graph of the wind farm and the system states. This is followed by a description of the system dynamics and model output equations in Sections 2.3 and 2.4, respectively. Finally, Section 2.5 outlines a method to incorporate real-time data into the formulation to improve the accuracy of the model. The resulting data-augmented dynamic model provides a control-oriented approach to capture the trajectory of farm-level power output changes due to dynamically changing yaw angles of the individual turbines.

2.1 | The wind farm graph

Following the approach in Starke et al,[18] the nodes $\mathcal{N}$ of a wind farm graph comprise the set of turbines and the edges $\mathcal{E}$ as the directed connections between turbines determined by the wake interactions in the farm. The wind farm is thus represented as a directed graph $\mathcal{G}=(\mathcal{N},\mathcal{E})$, where we define a directed edge between the i^{th} turbine and the j^{th} turbine only if the i^{th} turbine is in the wake of the j^{th} turbine. Figure 1A shows an example wind turbine configuration and the modeled wakes of the turbines, and Figure 1B shows the corresponding directed graph. Based on this graph definition, the elements of the wind farm interconnection matrix Λ are $\lambda_{ij}=1$ if the i^{th} turbine is in the wake of the j^{th} turbine and $\lambda_{ij}=0$ if the i^{th} turbine is not in the wake of the j^{th} turbine.

Turbines that are not affected by the wakes of other turbines, for example, Turbines 1 and 2 in Figure 1, are labeled freestream turbines. These turbines act as the head nodes of subgraphs (i.e., leader-follower trees) made up of a freestream leader turbine and all the follower turbines, which are weakly connected to that lead (i.e., turbine through directed edges). In Figure 1, the two leader-follower trees would consist of Turbines 1, 3, and 4 and Turbines 2 and 4, where freestream Turbines 1 and 2, respectively, act as leaders. These leader-follower trees each form their own subgraphs, enabling the farm to be thought of as a stack of multiple connected subgraphs.

We next describe the system states, which are defined in terms of the pairwise connections between turbines.

2.2 | System states

We build upon the approach in Starke et al[18] where the states of the system are the normalized velocity deficits associated to each turbine pair. The states in the approach outlined in Starke et al[18] assumed that all turbines were always facing the incoming wind direction, thus not addressing turbine yaw. The model presented here adds the ability to model dynamic turbine yaw. A dynamic turbine yaw model requires a deficit expression that takes into account both the yaw angle and its effect on the shape of the wake as it propagates downstream. The new velocity deficit model for the states follows Bastankhah et al,[6] which describes the deformation in the wake shape using a relatively simple analytical expression that is easy to evaluate. We now present a short overview of this model, which is explained more fully in Appendix A of this paper.

The basis of the new state description is an expression for the normalized velocity deficit based on the Gaussian model of[26]

$$\frac{\delta u}{U_\infty}=C(x)\exp\left[-\frac{(y-y_c)^2+(z-z_h)^2}{2\sigma(x,\theta)^2}\right], \tag{1}$$

where $\delta u(x,y,z)$ is the wake deficit, where x is the streamwise, y is the lateral (spanwise), and z is the vertical direction. U_∞ is the inflow velocity at hub height, $C(x)$ is the normalized velocity deficit at the wake centerline, y_c is the y-location of the center of the wake (it can depend on downstream distance x due to yaw), z_h is the turbine height, and $\sigma(x,\theta)$ is the wake width, which is a function of both streamwise distance x and polar angle θ. The polar angle-dependent wake width is modeled as a combination of linear downstream growth and deformation due to yaw according to

$$\sigma(x,\theta)=k_w x+0.4\xi(x,\theta), \tag{2}$$

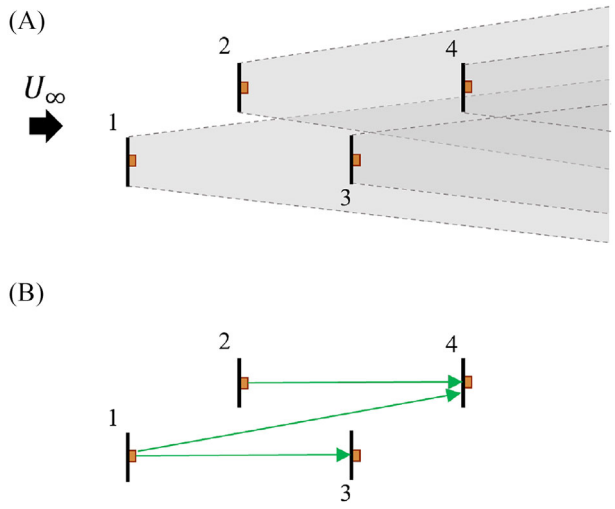

FIGURE 1 Schematic example of how a graph is defined for a wind farm with four turbines, where (A) shows the wind turbines and their wakes and (B) shows the directed graph that is formed from analyzing the wake interactions displayed in (A)

where the first term is the standard linear wake expansion process due to turbulent diffusion and k_w is the wake expansion coefficient. The second term represents the wake shape. The function ξ is an analytical expression that depends on the streamwise distance, polar angle, and turbine yaw angle. For practicality, here, we use an empirically fitted function that smoothly interpolates between various relevant asymptotic regimes, as provided in Appendix B of Bastankhah et al.[6]

The wake centerline normalized wake deficit $C(x)$ is evaluated using the following equation[6]:

$$C(x) = 1 - \sqrt{1 - \frac{C_T \cos^3(\gamma)}{2\tilde{\sigma}^2(x)/R^2}} \tag{3}$$

where C_T is the thrust coefficient of the turbine, γ is the yaw angle, R is the radius of the turbine disk, and $\tilde{\sigma}^2$ is proportional to the wake cross-sectional area. Further detail is provided in Appendix A.

Using the model outlined above, we can compute the normalized velocity deficit associated with a turbine i located in the wake of turbine j as follows. Noting that $\delta u(x,y,z)$ depends on position in the wake, we evaluate the state variable ϕ_i^j as the average normalized velocity deficit at the i^{th} turbine disk (area A_i) according to

$$\phi_i^j = \frac{1}{A_i} \int_{A_i} C(\Delta x_{i,j}) \times \exp\left[-\frac{(y - y_{c,j})^2 + (z - z_{h,j})^2}{2\sigma(\Delta x_{i,j}, \theta_j)^2} \right] dydz, \tag{4}$$

where $\Delta x_{i,j}$ is the distance between turbine i and turbine j and $y_{c,j}$ is the transverse displacement of the wake due to turbine j at turbine i.

The state vector then comprises the collection of all the normalized velocity deficits calculated between each turbine pair in the wind farm:

$$\Phi_k = \left[\phi_1^1 \phi_1^2 \phi_1^3 \ldots \phi_1^N \phi_2^1 \ldots \phi_N^{N-1} \phi_N^N \right]^T. \tag{5}$$

Here, N is the total number of turbines in the farm, and the state vector has a total of N^2 velocity deficits. By representing both the area overlap and the wake deficit strength, the state vector represents the appropriately weighted strengths of interactions among all turbine pairs.

The nonzero elements in the state vector correspond to the edges of the wind farm graph (wake interactions). These graph edges depend on a number of different factors, such as the geometry of the farm, the incoming wind direction, and the wake expansion coefficient k_w, which is a function of atmospheric conditions. This wake expansion coefficient is of particular importance in defining the graph edges because it determines how quickly the wake of a turbine expands and the velocity deficit diminishes as it advects through the farm. It therefore defines the wake area and the associated wake interactions between turbines. For example, two turbines that are not aligned and do not interact when the turbulence intensity is low may interact under conditions of higher turbulence due to the associated increase in the wake expansion coefficient (i.e. wider wake). Since the graph connections are based on the wake deficit area, this widening of the wake can result in connections between turbines that would not exist with a narrower wake. Although, the strength of these connections (edge weights) will be weaker due to the more rapidly decreasing velocity deficit.

Any change in wind direction will also shift the connections between turbines, as the distances and angles between turbines are defined on the incoming wind angle. Additionally, when a turbine is yawed, it affects both the wake shape behind the turbine and also the power produced by the turbine. This, in turn, also changes the connections between turbines. We next describe the update map that enables us to capture the dynamic behavior of the farm under such changing conditions.

2.3 | System dynamics

Having defined the system states we now define the associated dynamics, which are described through the linear map:

$$\Phi_{k+1} = A\Phi_k + E_k, \tag{6}$$

where Φ_k represents the states of the model at time index k and A is the update matrix. Here, we define $A = I$, and the input E_k updates the system states based on changes to interturbine interactions, for example, due changes in turbine yaw angles for which the effects on the graph interconnection propagate downstream.

Our implementation follows the log file approach,[21] in which we start with two static graphs, the initial graph and the final graph, and catalogue the changes over the time period at intervals indexed by k. The update at each time step k is implemented in an event-based framework that is detailed in Starke et al.[18] Each

$$E_k = f(\Phi_k, \tau_k, \Delta\mathcal{E}_k) \tag{7}$$

is a function of the updated state vector Φ_k with values ϕ_i^j for $(j = 1,...,N)$ for the i^{th} turbine and $\Delta\mathcal{E}_{k,i}$ is the list of the edge changes that propagate from a change to turbine i. We next describe the time delays associated with the propagation of the effect of changes to the graph, which are denoted $\tau_{k,i}$.

Equation (7) shows that the edge change input is a function of a time delay. The time delays τ in Equation (7) capture the finite travel time of aerodynamic information between turbines (e.g., the propagation of the effect of the yawing of an upstream turbine for which the wake speed is a function of the wind velocity). For example, a change that happens at the j^{th} upstream turbine in the illustrative Figure 2 will not reach the i^{th} turbine until the air parcels arrive at the i^{th} turbine while traveling at speed $u_j(x)$. Incorporating this time delay and the associated transient behavior of the graph edge set enables the model to capture the dynamic variation of the power output due to a change in wind farm conditions. These transients have been shown to be important in capturing an accurate power output trajectory,[18] which has implications for the accuracy of real-time predictions of the available power and control.

Assuming a constant advection velocity u_j, the time-delay between turbines would be given by $\tau = \Delta x_i^j / u_j$ (Figure 2), where $\Delta x_i^j = x_i - x_j$ is the distance between the two turbines. However, the velocity of the wind varies between the turbines as a result of the wake expansion behind the first turbine. We employ an analytical expression for the time delay that includes the position-dependent velocity deficit in the evolving wake (Equation (1)). As representative of the wake velocity, we have chosen to use the velocity at the centerline of the wake, where $y = y_c$ and $z = z_h$, that is, $u_j(x) = U_\infty[1 - C(x)]$. We find that the centerline velocity provides a sufficiently accurate representation of the advection speed representing the most relevant parts of the wake, and it can be evaluated very efficiently in the modeling. Other more accurate options such as the disk averaged velocity would require significantly more complicated operations. We can then find the time delay associated with each graph edge by integrating downstream displacements over the distance between the turbines:

$$\overline{\tau}_i^j = \int_{x_j+D}^{x_i} \frac{1}{U_\infty(1 - C(x))} dx. \tag{8}$$

Here, since the wake model chosen for this application is more appropriate in the far field, we integrate starting one diameter downstream of the j^{th} (upstream) turbine (at x position denoted by $x_j + D$, up to the position of the i^{th} turbine x_i, where D is the turbine diameter). This approach improves the accuracy of the model compared to the approach in used in Starke et al,[18] which simply approximated u using the freestream velocity U_∞ to directly compute $\tau = \Delta x_i^j / U_\infty$.

Substituting the previous relation in for $C(x)$ gives the following expression:

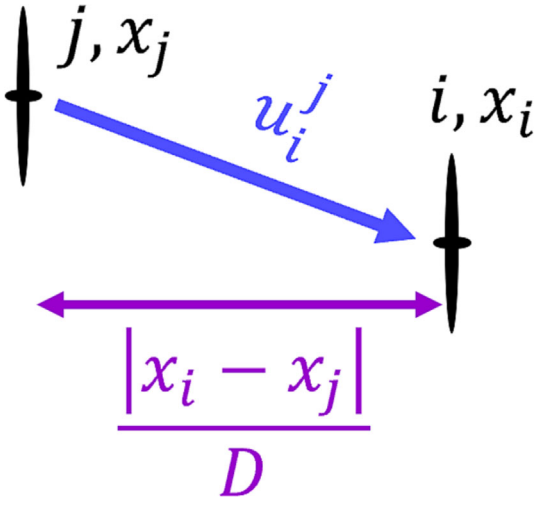

FIGURE 2 A schematic illustrating the geometry of the wind farm used to calculate the time delay between the i^{th} and j^{th} turbines. The velocity u_i^j is the effective velocity that affects the i^{th} turbine due to the j^{th} turbine and is modeled as $U_\infty[1 - C(x - x_j)]$ for x between x_j and x_i

$$\bar{\tau}_i^j = \int_{x_j+D}^{x_i} \frac{1}{U_\infty}\left[1 - \frac{C_T \cos^3(\gamma)}{(2/R^2)\tilde{\sigma}^2(x)}\right]^{-1/2} dx, \tag{9}$$

where $\tilde{\sigma}^2(x)$ is provided in the appendix in Equation (A10). For ease of implementation, we apply a Taylor series expansion $(1-y)^{-1/2} \approx 1 + y/2 + \ldots$ using the parameter

$$y = \frac{C_T \cos^3(\gamma)}{(2/R^2)\tilde{\sigma}^2(x)}. \tag{10}$$

The approximation becomes more accurate for small y, that is, far downstream, as the wake size $\tilde{\sigma}$ exceeds the disk radius R. Using the first-order approximation, we can write the integral (Equation (9)) as

$$\bar{\tau}_i^j = \int_{x_j+D}^{x_i} \frac{1}{U_\infty}\left(1 + \frac{1}{2}\frac{C_T \cos^3(\gamma)}{(2/R^2)\tilde{\sigma}^2(x)}\right) dx. \tag{11}$$

Using the explicit expression for $\tilde{\sigma}^2(x)$ (Equation (A10)) and performing the integration give the following explicit expression for the time delay:

$$\bar{\tau}_i^j = \frac{1}{U_\infty}\left[x_i - x_j - D + \frac{C_T R^2 \cos^3\gamma}{1.6\xi_0 k_w (1-\cos\gamma)} \left(\ln\frac{k_w x_i + 0.4\xi_0}{k_w x_i + 0.4\xi_0 \cos\gamma} - \ln\frac{k_w (x_j + D) + 0.4\xi_0}{k_w (x_j + D) + 0.4\xi_0 \cos\gamma}\right)\right].$$

Although there is an apparent singularity when $\gamma = 0$ due to $(1-\cos(\gamma))$ in the second term of the integral, the natural logarithm terms in the numerator also vanish in such a way that the ratio remains meaningful. Numerically, this was dealt with by adding a very small constant with the same sign as the cosine term in the denominator to keep the expression $(1-\cos(\gamma))$ finite (but negligibly small). Figure 3A shows the indefinite integral as function of downstream distance x, representing the time delay over the bounds of the yaw angles from $\gamma = -20^\circ$ to $\gamma = 20^\circ$, and at $\gamma = 0^\circ$. Note that the lines representing $\gamma = -20^\circ$ and $\gamma = 20^\circ$ fall on top of each other since the model has no yaw direction bias. Figure 3B shows the time delays between turbines (with $\Delta x_i^j = 100$ m) over a range of upstream turbine yaw angles.

The dynamic formulation of this model includes a list of events that specifies specific changes for each turbine. The propagation of these changes as well as a changing velocity inflow necessitates keeping a time history for each state in the system that is longer than the flow-through time of the farm. There is a limited history kept for each state, and this history vector is updated at every time step. Additionally, in order to speed up the model, a history is only created for states that are greater than zero, meaning there exists a directed connection between those two turbines. If a connection is not used in the simulation, then a history is not created. If a new connection is added through the state change vector,

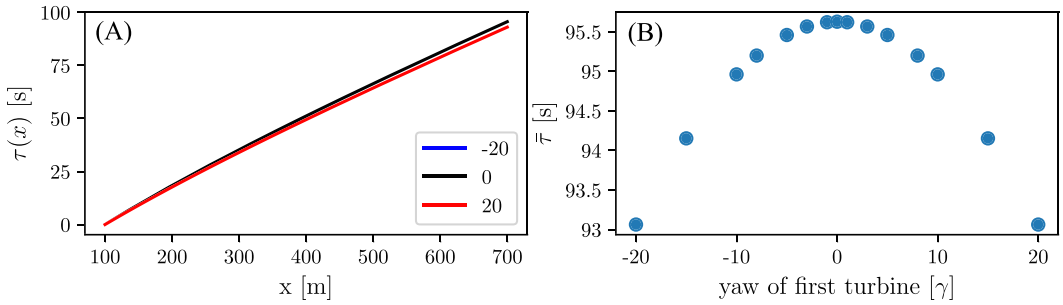

FIGURE 3 (A) Modeled time delay (in seconds) as a function of streamwise distance x, according to Equation (11) where the upper integration limit is x and the different colors represent different yaw angles. (B) Modeled time delays (in seconds) between two turbines as a function of the yaw of the first turbine. For this test, the turbine diameter was 100 m, the turbine hub height was also 100 m, the freestream velocity was $U_\infty = 7.77$ m/s, and was determined by setting the friction velocity $u^* = 0.45$ m/s, then using the equation $U_\infty = (u^* \ln(z_h/z_0))/0.4$ to find the freestream velocity at hub height. The local coefficient of thrust was $C_T' = 4/3$, and the wake expansion coefficient was determined using $k_w = u^*/U_\infty = 0.0579$

the history is initialized with the value and then continues to be updated from its initialization time. A history is also kept of the time delays between the turbines, since this quantity also changes with inflow velocity, yaw changes, and wind direction changes.

2.4 | Output: velocity deficit

We define the output as a normalized cumulative velocity deficit for each turbine (i), $\Delta u_i^* \equiv \Delta u_i / U_\infty$. For each turbine, this value is obtained by a linear superposition of all of the velocity deficits due to all of the upstream turbines whose wakes impact turbine i. The vector formed by all the turbines' cumulative velocity deficit at the next iteration ($k+1$), denoted by Δu_{k+1}^*, can be calculated efficiently by multiplying the matrix $\Lambda(\tau_k)$ with the state vector containing all of the individual turbines' normalized velocity deficits, that is,

$$\Delta u_{k+1}^* = \Lambda(\tau_k)\Phi_k(\tau_k). \tag{12}$$

This matrix $\Lambda(\tau_k)$ is of dimension $N \times N^2$, where the i^{th} row holds the directed wake relationships for the i^{th} turbine. The matrix is also delay-dependent and corresponds to the graph that describes the current turbine interconnections found from the subgraphs present at previous times; that is, it indicates which turbines will impact other turbines in the farm after time delay k. The quantity $\Phi_k(\tau_k)$ is the state vector (array of normalized velocity deficits) at time k, which implements changes according to the time delay associated with the changes in the relationships between turbines, thus incorporating the delay of information moving through the system. In this formulation, the states are updated at every time step, but the effect of that update does not reach the output until after the time delay.

The output is then used to compute the actual velocity deficit at turbine i, which is given by $\delta u_i = [\Delta u^*]_i U_\infty$. This value has to be subtracted from the common background reference velocity U_∞ to find the wind velocity at turbine i. Incorporating the effect of yaw of the turbine itself, we can express the vector of disk velocities for each turbine according to

$$u_{d,k+1} = U_\infty \cos(\gamma_{k+1})\left(1 - \Delta u_{k+1}^*\right)\left(1 - \frac{C_T'}{4 + C_T'}\right) \tag{13}$$

where C_T' is the local coefficient of thrust for the turbine and γ_{k+1} is the current yaw of the turbine, which affects how much of the freestream velocity is normal to the turbine disk. Finally, considering the effect of yaw on the turbine itself, the vector of power generation of all turbines at step $k+1$ is found according to

$$P_{k+1} = \frac{1}{2} A \rho [u_{d,k+1}]^3 C_P' [\cos(\gamma)]^p, \tag{14}$$

where for our case we set $C_P' = C_T'$ (consistent with the optimal values for the local power and thrust coefficients which are defined using the local velocity at the turbine disk) and the exponent p determines the effect of yaw on the power and changes for different turbines. We address the value used for this exponent later in Section 3.

2.5 | Data-augmented model

In this section, we describe a method to incorporate measurements from a wind farm plant to improve the prediction of the graph model. In particular, we use the instantaneous turbine power measurements from the freestream turbines. The freestream turbines are either determined a priori by the wind direction or computed dynamically if the wind direction is unsteady, such as in the example presented in Starke et al.[18] These measurements, which are readily available in operating farms, are run through a temporal smoothing filter, and the filtered power signal is then used to determine the free-stream velocity appropriate for turbine n. The power measurements are represented by $P_n = 1/2\, U_d^3 \rho A C_{P,n}'$, where $U_d = (U_{\infty,n} cos(\gamma_n)) C_{T,n}'/(4 + C_{T,n}')$ is composed of the velocity component normal to the turbine, $U_{\infty,n}\cos(\gamma_n)$, and the axial induction adjustment of the freestream velocity to find the disk velocity, represented here using the local thrust coefficient as $C_{T,n}'/(4 + C_{T,n}')$. This equation is then solved to find the freestream velocity, obtaining the following expression:

$$U_{\infty,n} = \frac{1}{\cos(\gamma_n)}\left(\frac{2P_n}{\rho A C_{P,n}'}\right)^{1/3}\left(\frac{4 + C_{T,n}'}{C_{T,n}'}\right). \tag{15}$$

Figure 4 provides a schematic of the full data-augmented model. Since this method uses the actual power measurements from turbines, it can be applied in real time for model updates that requires the inflow velocity as input. The model can be run simultaneously with the wind farm plant (LES in our case), and the inflow is updated dynamically using the instantaneous power measurements from the plant's turbines. Since the graph model relies on the pairwise interconnections between the turbines and does not resolve a complete velocity field, a changing inflow can be advanced through the farm in the same way as the varying connections between the turbines.

In the next section, we validate the model in static and dynamic settings.

3 | MODEL VALIDATION

The model was first tested under static conditions. The setup has two turbines that are aligned with a distance of 7D between them in the streamwise direction. The first turbine is then set to various yaw positions: $\gamma \in [-30^\circ, 30^\circ]$. We compare average power output predictions from LES under stationary conditions with the model results. Here, we report results from the model described in Section 2 without the data augmentation (since we are comparing to static conditions for a wind turbine pair) as well as a simplified model that does not include the curled wake shape; that is, we set $\xi = R$, constant in Equation (2), which assumes an axisymmetric distribution of the wake throughout its evolution.

The LES is performed with the JHU LESGO code[27] with an actuator disk model and a correction factor accounting for finite grid resolution.[28] This code has been validated for several wind energy applications.[29–31] The simulation uses an inflow generated by the concurrent-precursor approach.[32] The turbines have a diameter and hub height of $D = z_h = 100$ m. A ground surface roughness height of $z_{0,lo} = 0.1$ m is used to prescribe the bottom boundary condition of the LES. The turbulence intensity at hub height from the concurrent-precursor fields was TI = 7.88%. The main simulation domain uses ($N_x \times N_y \times N_Z = 256 \times 256 \times 192$) grid points for a domain size of 3 km $\times$3 km$\times$1 km. The Lagrangian scale-dependent dynamic subgrid-scale model[33] is used to determine the eddy viscosity without tuning parameters. The turbine (local) coefficient of thrust is kept constant at $C_T' = 1.33$ throughout the simulation.

Figure 5 compares the mean turbine power obtained from the LES with the original version of the graph model from Starke et al[18] using the Gaussian wake formulation and the updated graph model presented in this paper that uses the curled wake formulation outlined above. All the results are normalized by the power of the first turbine at zero yaw. The graph model results using the curled wake formulation are represented by the red symbols, and the graph model results using the axisymmetric wake formulation from Starke et al[18] are represented by the blue symbols. The figure shows that in this case, yawing does not meaningfully increase the overall power generation (crosses), as the gains seen by the second turbine are offset by the losses of the first turbine.

The results show that even if the deficits matched for $\gamma = 0$, the axisymmetric formulation would overpredict the wake deficit caused by the first turbine because it does not account for the "curling," which in fact increases the velocity seen by the downstream turbine. As a result, the axisymmetric formulation results in lower power and a power curve that is too shallow for the second turbine. The curled wake does a better job of capturing the wake deficit caused by the first turbine, particularly on the right side of the curve. Although it underpredicts the wake deficit slightly on the left side of the curve, overall, the curled wake method seems to represent the relationship between the first and second turbines better than the axisymmetric model in this case.

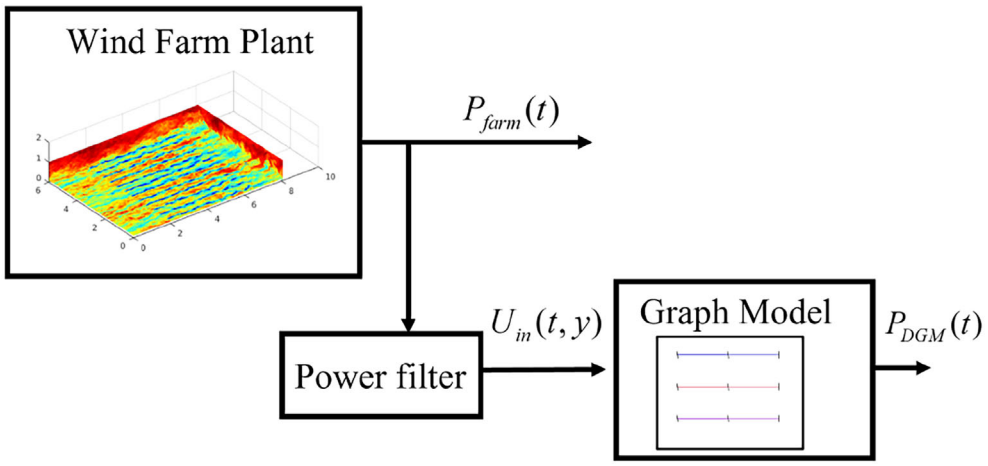

FIGURE 4 A block diagram showing how the dynamic yaw model uses the instantaneous power measurements from the LES wind farm plant to augment the model's power prediction.

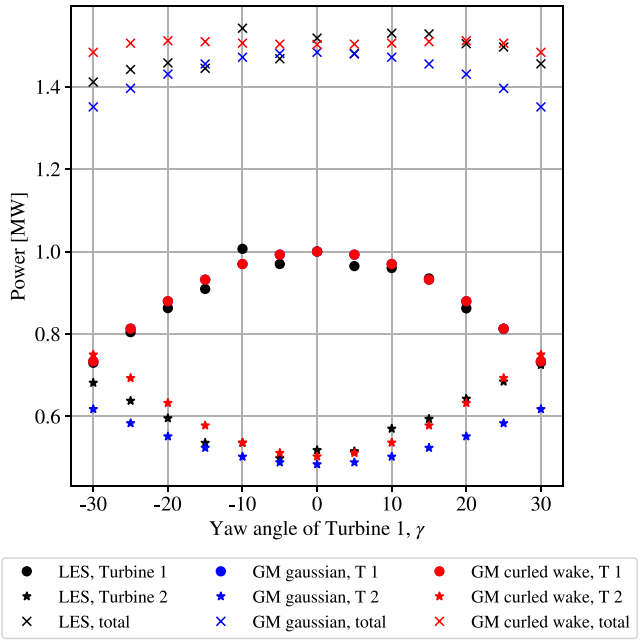

FIGURE 5 Power generation of two turbines where the upstream turbine $(T1)$ is yawed at angle γ. This plot compares the results from the LES study with the graph model using a simple axisymmetric velocity deficit distribution and the graph model using the more accurate curled wake shape formulation for the velocity deficit. Power production of each turbine is shown separately (circles and stars) and added together (crosses)

This two-turbine study was also used to fit the exponent p parameter for the power, shown in Equation (14), similar to the study performed in Gebraad and van Wingerden[34] and Gebraad.[8] The exponent value that best fits the LES turbine models used here best is $p = 0.5$, which will be the value used in the power calculations done in this study going forward.

Next, the dynamic yaw model is tested in a time-dependent setting through comparison to an LES flow generated by a dynamically yawing wind turbine pair. The simulation was done using the JHU LESGO code with the same configurations and parameters as the static case described above. For the time-dependent case, the first turbine is dynamically yawed to 15° in the first 25 s, and then returned back to 0° yaw starting at 150 s. The results are averaged over 120 realizations to smooth out the instantaneous turbulence fluctuations. Figure 6 shows the power from the first turbine (solid lines) and the second turbine (dashed lines) as a function of time during the dynamic yawing from the LES (in black) and the model (in red). As described in Section 2, the model result applies the changes to the strength of the graph edge (wake interaction) due to the change in yaw of the first turbine in discrete steps, and the effect of this is visible in the power of the first turbine, particularly over the first 25 s. The results demonstrate that for this test, the model is able to match the timing of the yaw change, including how long it takes the change from the first turbine to affect the second turbine (the effect begins to be seen at 130 s, where the second turbine increases its power once the wake deflection has reached its location). The model also captures the magnitude, which is how much the power of the second turbine increases as a result of the yawing of the first turbine. The power of the second turbine is reduced again starting near 265 s, once the first turbine's return to the unyawed condition at $t = 150$ s is felt at the second turbine's location.

Finally, we also show a validation of the data-augmented graph model where we use an LES as the wind farm. We apply this to a small wind plant with three turbines aligned to the incoming wind direction in a single column. Here, the aim is to test the performance and effects of the data-driven input to determine the local inflow velocity U_∞ from the measurements. Figure 7 shows a comparison between a time series of the power from the LES and the full data-augmented dynamic yaw model. Figure 7A compares the power from the first turbine in the column, that is, from the "freestream" turbine from the LES and the model. We can see here that the data-augmented graph model follows the LES power very well. This result is expected, as the graph model is using the measured velocity for the first turbine. Figure 7B,C shows the time series for the second and third turbines, respectively. In these cases, the model matches the LES less well, especially missing the high-frequency turbulent fluctuations that are present in the LES, which is expected because these cannot be captured by the wake modeling approach and are faster than the update rate of the model. The dynamic yaw model captures the low-frequency behavior responding to imposed yaw and turbine actions, but without the turbulent fluctuations.

The model is able to follow the low-frequency trend of the second and third turbines. One of the salient features of the model is that it takes the interturbine travel time from the first turbine for the second and third turbines into account and can represent the changes in the inflow (which can be seen in the initial constant value of the third turbine). The second point of note is that the variations of the inflow, such as the large

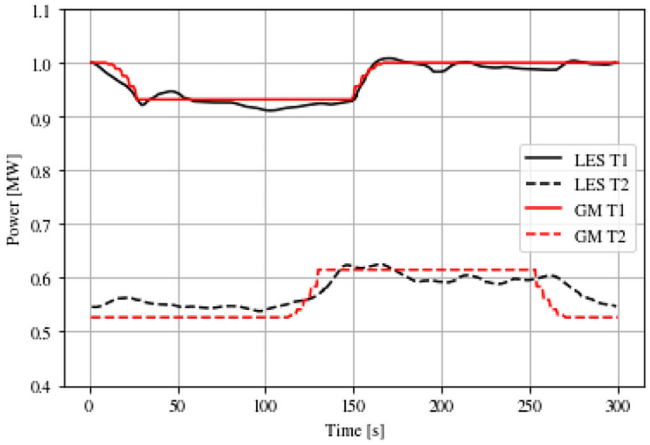

FIGURE 6 Time-dependent power production of two turbines in which the upstream turbine (solid lines) is yawed at $t = 0$ s to 15° and then back to 0° at $t = 150$ s. The downstream turbine power history is shown as dashed lines. The figure compares the results of the dynamic yaw model using the curled wake shape model (red lines) to the results from LES (black lines)

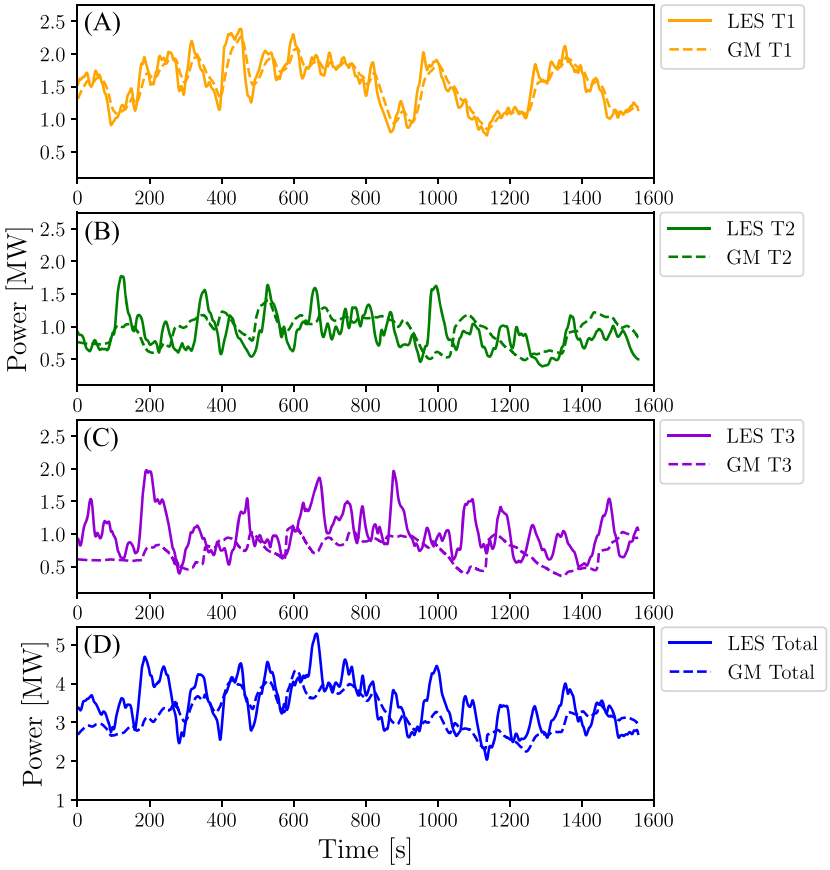

FIGURE 7 A comparison between power from an LES compared with a graph model power output using the same inflow for three turbines that are in an aligned configuration, where (A) is power from the freestream turbine in the column, (B) is the power from the second turbine, and (C) is the power from the third turbine. The total power for the column from the LES and the graph model is shown in panel (D)

troughs and peaks around 800–1200 s in Figure 7A, can be seen propagated to the second and third turbines, such as the trough at 1000 s in Figure 7B. Recalling that the data-augmented approach also allows for inflow velocity dependency in the spanwise direction, present results confirm that this model is able to represent heterogeneous flow in time in the spanwise direction as well as in the streamwise direction due to the

advective character of the model with time delays. Future work involves validating the time-delay formulation on a wider variety of interturbine distances in LES, which may require additional parameter adjustment.

Figure 7D shows a comparison between the LES and the graph model for the entire column of turbines, that is, the sum of the three turbine powers. As mentioned above, while the graph model misses the high-frequency, high-amplitude turbulent fluctuations, it is able to track the low-frequency power output signal of the wind farm. The model resembles a smoothed-out version of the LES signal. In terms of the broader applicability of this type of graph model for control-oriented applications, this artifact could affect the control actions selected using this model if applied in the case of rapid changes (e.g., pitch control). However, this is not an issue for yaw, which inherently takes place at slower timescales. The strength of this approach is its ability to represent changing overall conditions in the wind farm that occur on slower timescales and to represent ensemble-averaged effects of yaw actions, even if applied on relatively fast time-scales (i.e., a few minutes). In the next section, we focus on applications of yaw-based control in which desired effects occur over time-scales of several minutes.

4 | APPLICATION: WIND FARM CONTROL

In this section, we discuss a possible application of the model and demonstrate proof of concept for model-constrained optimal control using dynamic yaw actuation. We first examine how the dynamic graph model could be incorporated into a yaw controller formulation on its own and then present a simple application to a wake model wind farm plant.

4.1 | Controller formulation

The structure of the yaw model-constrained optimal control is shown in Figure 8, where the dashed box outlines the controller. The controller is composed of an optimization solver, which uses information from the cost function, and the dynamic yaw model to find an optimal yaw solution γ_i for each turbine i. This solution is then given to the wind farm plant.

The optimal control power tracking problem is composed of solving a finite-time optimal control problem for each controller update. At each iteration, the optimization solver finds one optimal yaw for each turbine over the finite time horizon T. The resulting yaw is then applied to the wind farm for a time $T_a < T$. The problem is then solved again for a new time frame starting at T_a and ending at $T_a + T$, representing one controller update.

The yaw of each turbine, γ, is used as the control variable in the model-constrained optimal control. For each iteration, the optimization problem finds the best yaw angles over the horizon time according to a cost function that minimizes the difference between the power reference signal and the power output of the farm predicted by dynamic graph model. The cost optimization has the form:

$$\min_{\gamma} \left[\int_0^{T_a} (P_{GM} - P_{ref})^2 dt \right], \tag{16}$$

where P_{GM} is the power from the dynamic yaw model, which is integrated into the future to predict the behavior of the wind farm. Yaw is incorporated into this problem through the dynamic yaw model formulation. The optimization is subject to the limits

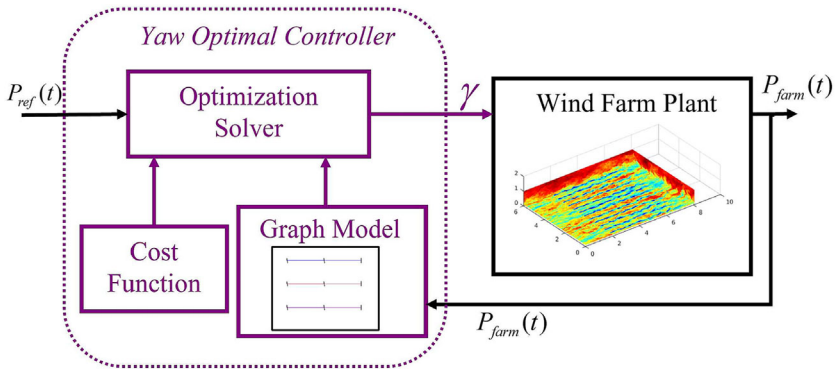

FIGURE 8 Block diagram showing the structure of the model-constrained optimal control of the wind farm. The model-constrained optimal controller (shown in purple) is composed of an optimization solver that minimizes a cost function constrained by the dynamic yaw model. The controller then feeds the optimal yaw to the wind farm plant, represented in the right block with an instantaneous snapshot from a wind farm simulation.

$$-30^\circ \le \gamma \le 30^\circ, \tag{17}$$

since the models for wind turbine yaw are less accurate beyond these angles. The minimization of the objective function is accomplished using Powell's Algorithm, see Appendix B, and is coded in Python in our application. We remark that the version of Python used in our work does not support setting bounds on the minimization. Therefore, the bounds were enforced by not allowing values that fall outside of the allowable yaw range: If a value falls outside the viable range, the optimization function exits that evaluation, and the optimization continues.

We first demonstrate a simple implementation in which the goal of the optimization is to reach a specified steady power output that is greater than the nominal value, but less than the maximum power achievable by the farm. The wind farm configuration used is a six-turbine aligned wind farm arranged in two columns of three turbines each, shown in Figure 9A. In this case, the optimization is attempting to reach a final steady power output of 12 MW, shown in Figure 9B by the black dashed line. The unyawed power of the farm is 11.61 MW, which is the initial condition for the optimization.

Figure 9A shows the final yaw values of the turbines, where only the first two turbines are yawed a significant amount. The optimization process assumes that these optimal yaw values can be applied instantaneously at the start of the time horizon. Figure 9B shows the time evolution of the power of the wind farm (in blue) in reaching this optimal configuration. There is an initial dip when the turbines are yawed, and then we observe two increases in the power, which is the result of the wake effects from the first row decreasing on the second and the third rows. The effect of the wake on the second row is larger than on the third row, which is expected as the wake effects are inversely proportional to distance due to the growth and weakening of the wake. The final value that the optimization achieves is 11.98 MW.

Figure 9B shows that the power output of the farm as the turbines yaw initially drops before reaching the desired value due to the transient response of the farm wakes of the yawing turbines propagating through the farm. This behavior demonstrates that in order for the yaw to be effective, the time horizon needs to be long enough for the improvement in the total error measured by the cost function to outweigh the initial degradation. For example, if the time horizon in Figure 9B was only 4 min, the decrease in the error would not have been enough to outweigh the initial drop in the integral of the total error over the time horizon, and the optimal solution would be to take no action. This initial drop in power

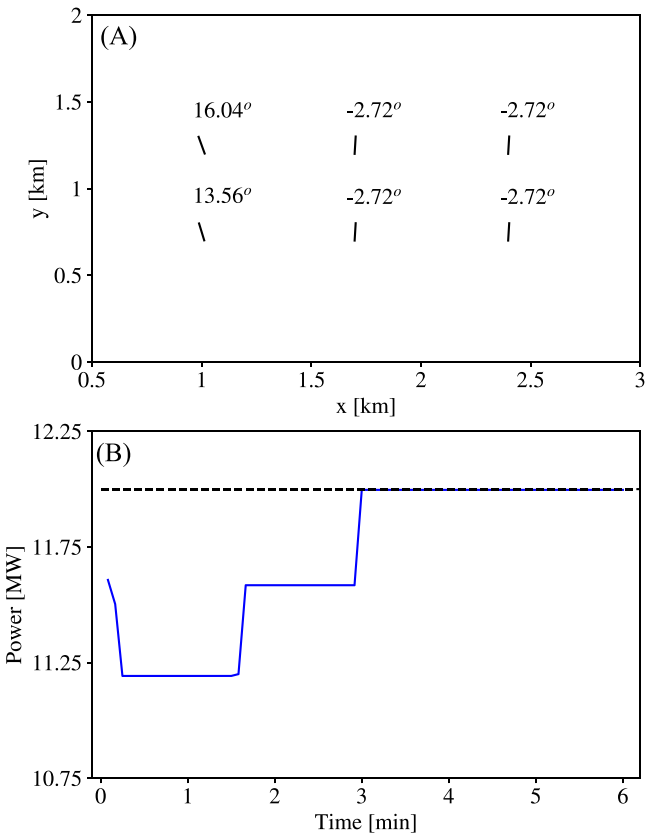

FIGURE 9 An example of a six-turbine wind farm optimized using a dynamic yaw response, where (A) shows the final yaw of turbines (wind farm is seen from above) and (B) shows the evolution of the power of the farm in time (blue line), as compared to the target power reference signal (black dashed line)

and subsequent increase reflects the interturbine travel distance that needs to be captured in modeling the dynamic response of the wind farm to the yawing turbines. This travel time plays an important role in determining the optimal control action and demonstrates that static yaw optimization that only computes the power gains in the final yaw configuration misses the initial power losses and thus may overestimate the benefits of yaw control, particularly under dynamically changing conditions.

We note that the results here demonstrate the effect of the transient behavior, but that improvements in performance are possible through changes to the control strategy. For example, computing a series of yaw angles over the time horizon rather than the single best yaw value over the full time horizon would likely improve performance. However, we do not do that here, as such a scheme would require the calculation of a full yaw trajectory for each turbine as a function of forward time. As the structure of the graph model prevents the use of gradient-based optimization at this time, we only calculate the best single yaw value overall for each turbine over the finite time horizon. Further refinements will be required to develop a full model predictive control framework for dynamic yaw.

In the next section, we discuss modifications to the setting of Figure 9 to consider additional relevant timescales for application to power tracking.

4.2 | Yaw control results

We now use the dynamic yaw model within a wind farm control scheme for farm-level power tracking. In this application, the controller uses the dynamic yaw model as both the wind farm plant surrogate as well as the control model. This farm has the same number of turbines and the same configuration as that shown in Figure 9A. The desired power signal is shown as the black dashed line in Figure 10. It includes a 9% power increase at 30 min.

Here, we make a couple of changes to the approach in the previous example to improve the system performance. Rather than assuming the yaw is applied instantaneously at the beginning of the time horizon, we incorporate the time corresponding to the physical action of yawing, including a yaw rate of 4° per minute. We also split the full time horizon into control intervals and recompute the yaw set points (update the control set points) at these intervals based on the current state of the system; that is, each optimization horizon does not encompass the whole control signal, but rather encompasses a subsection of the control signal at each evaluation. More specifically, the controller first computes optimal yaw over the full time horizon. Then, the turbines are yawed according to the set yaw rate until either they reach the desired yaw or it is time for another controller update. The current yaw of the turbines is then used as the initial conditions for the next update, and the process is repeated. From this process, one can infer that the turbines will not always reach the target yaw within one controller update period. In this example, the simulation time was 3600 s, the controller look-ahead time period was 300 s, and the controller was updated every 240 s. To lessen the inherent dependence of the nonlinear optimization problem on initial conditions, an ensemble of optimizations are run (we used 10 members in the ensemble), each with differing initial conditions that are perturbed randomly within ±2° from the current yaw. This ensemble of optimizations can be parallelized since each realization is independent of the others. Thus, the addition of an ensemble of optimizations will not extend the time it takes to reach the final control since they can be calculated in parallel. The final yaw control commands use the average of the ensemble results.

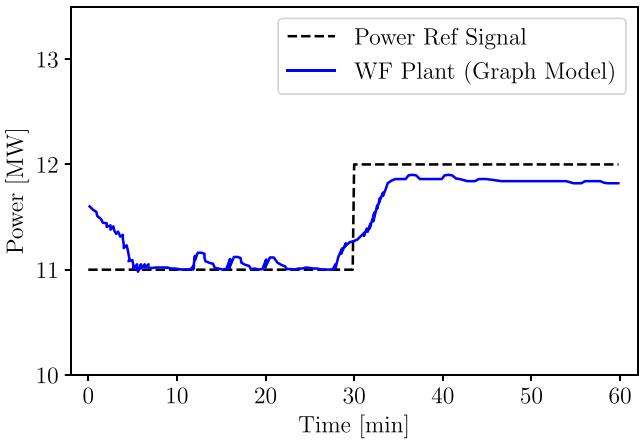

FIGURE 10 Wind farm power in an application of power tracking applied to a six-turbine wind farm where the dynamic yaw model is used as the wind farm plant as well as the control model. The figure shows the evolution of the power of the farm in time (blue line), as compared to the target power reference signal (black dashed line). The controlled signal has an RMSE of 0.24 MW from the reference signal.

Figure 10 shows the controlled response (in blue) alongside the desired trajectory (black dashed line). There are a few aspects of the control seen in Figure 10 worth remarking upon. The first is that the controller is able to curtail the power down to 11 MW from the initial power of 11.61 MW. Then, the yaw control is asked to navigate a step change in the middle of the control period. Following a step change is a somewhat unrealistic expectation for yaw control, due to the slower timescale of yawing. Nevertheless, we can see that the yaw control is able to anticipate the step change before it happens, due to the model-constrained optimal framework. Finally, we can see that even though the power output increases with the step change, it is unable to fully reach the final power reference value and seems to hover with a constant offset under the power reference signal. This could be a result of using the average of an ensemble of optimizations, or it could be that yawing simply cannot reach the demanded power increase of 8% under the assumed operating conditions.

5 | CONCLUSIONS

In this paper, we presented a new graph-based dynamic yaw model that is able to represent the transient response of a wind farm due to dynamic yaw actions of individual turbines at different times. The model incorporates a new analytical description of the wake deficit deformation due to yaw to represent the pairwise turbine interactions. The model also includes a time-delay formulation that accounts for the slower wake velocities behind turbines and the time delays for the propagation of velocity and yaw information through the farm. Real-time data are used to create a time-dependent and spatially-dependent velocity inflow that improves model performance. The model was validated both statically and dynamically using LES.

The dynamic yaw model was then implemented in a model-constrained optimal control structure to use yaw to control the power of a wind farm. The results showed that the controller was able to use yaw to achieve controlling a wind farm power output to follow a desired step change control signal. However, further work on the controller formulation could improve the performance, particularly in the response time of the control and the final value the controller achieves, which settles into an equilibrium state just below that of the reference signal.

The results shown here demonstrate that the proposed dynamic yaw model represents a computationally efficient approach to predict the time-dependent response including the transient behavior of a wind farm to changes in yaw that is suitable for control applications. The dynamic yaw model therefore provides a controls-based wind farm modeling approach that is able to represent a wind farm under changing conditions. Ongoing work aims to implement this model in conjunction with pitch control to provide additional control authority and enable faster control response. Future work to modify the model to enable gradient-based optimization or adapt an optimization scheme for a receding horizon approach is likely to improve the model performance.

ACKNOWLEDGEMENTS

G.M.S., C.M., and D.F.G. gratefully acknowledge funding support from the National Science Foundation (grant numbers DGE-1746891, CBET-1949778, and CMMI-1635430). This work was carried out at the Advanced Research Computing at Hopkins (ARCH) core facility (rockfish.jhu.edu), which is supported by the National Science Foundation (NSF) grant number OAC 1920103 and the Maryland Advanced Research Computing Center for computing resources.

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the US Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the US Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed in the paper do not necessarily represent the views of the DOE or the US Government. The US Government retains and the publisher, by accepting the paper for publication, acknowledges that the US Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for US Government purposes.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.2884.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Genevieve M. Starke https://orcid.org/0000-0001-9554-8455
Jennifer R. King https://orcid.org/0000-0001-6219-0098

REFERENCES

1. Shapiro CR, Gayme DF, Meneveau C. Modelling yawed wind turbine wakes: a lifting line approach. *J Fluid Mech*. 2018;841:R1.

2. Jiménez A, Crespo A, Migoya E. Application of a LES technique to characterize the wake deflection of a wind turbine in yaw. *Wind Energy*. 2010;13(6):559-572.
3. Bastankhah M, Porté-Agel F. Experimental and theoretical study of wind turbine wakes in yawed conditions. *J Fluid Mech*. 2016;806:506-541.
4. Howland MF, Lele SK, Dabiri JO. Wind farm power optimization through wake steering. *Proc Natl Acad Sci*. 2019;116(29):14495-14500. https://www.pnas.org/content/116/29/14495
5. Dou B, Qu T, Lei L, Zeng P. Optimization of wind turbine yaw angles in a wind farm using a three-dimensional yawed wake model. *Energy*. 2020;209:118415. https://www.sciencedirect.com/science/article/pii/S036054422031522X
6. Bastankhah M, Shapiro CR, Shamsoddin S, Gayme DF, Meneveau C. A vortex sheet based analytical model of the curled wake behind yawed wind turbines. *J Fluid Mech*. 2022;933:A2.
7. Wei D, Zhao W, Wan D, Xiao Q. A new method for simulating multiple wind turbine wakes under yawed conditions. *Ocean Eng*. 2021;239:109832.
8. Gebraad PMO, Teeuwisse FW, van Wingerden JW, Fleming PA, Ruben SD, Marden JR, Pao LY. Wind plant power optimization through yaw control using a parametric model for wake effects—a CFD simulation study. *Wind Energy*. 2016;19(1):95-114.
9. Fleming PA, Ning A, Gebraad PMO, Dykes K. Wind plant system engineering through optimization of layout and yaw control. *Wind Energy*. 2015;19(22).
10. Mühle F, Schottler J, Bartl J, et al. Blind test comparison on the wake behind a yawed wind turbine. *Wind Energy Sci*. 2018;3(2):883-903.
11. Bastankhah M, Porté-Agel F. Wind farm power optimization via yaw angle control: a wind tunnel study. *J Renew Sustain Ener*. 2019;11:23301.
12. Fleming P, Annoni J, Shah JJ, et al. Field test of wake steering at an offshore wind farm. *Wind Energy Sci*. 2017;2(1):229-239.
13. Fleming P, King J, Dykes K, et al. Initial results from a field campaign of wake steering applied at a commercial wind farm—part 1. *Wind Energy Sci*. 2019;4(2):273-285.
14. Howland MF, Quesada JB, Martínez JJP, Larranaga FP, Yadav N, Chawla JS, Sivaram V, Dabiri JO. Collective wind farm operation based on a predictive model increases utility-scale energy production. *Nature Energy*. 2022;7:1315-1338.
15. Bay CJ, Annoni J, Taylor T, Pao L, Johnson K. Active power control for wind farms using distributed model predictive control and nearest neighbor communication. In: American Control Conference; 2018.
16. Howland MF, Ghate AS, Lele SK, Dabiri JO. Optimal closed-loop wake steering—part 1: conventionally neutral atmospheric boundary layer conditions. *Wind Energy Sci*. 2020;5(4):1315-1338. https://wes.copernicus.org/articles/5/1315/2020/
17. Howland MF, Bossuyt J, Martínez-Tossas LA, Meyers J, Meneveau C. Wake structure in actuator disk models of wind turbines in yaw under uniform inflow conditions. *J Renew Sustain Energy*. 2016;8(4):43301.
18. Starke GM, Stanfel P, Meneveau C, Gayme DF, King J. Network based estimation of wind farm power and velocity data under changing wind direction. In: 2021 American Control Conference (ACC); 2021:1803-1810.
19. Annoni JR, Bay C, Johnson KE, Dall'Anese E, Quon EW, Kemper TW, Fleming PA. Wind direction estimation using SCADA data with consensus-based optimization. *Wind Energy Sci (Online)*. 2019;4(2):355-368.
20. Annoni J, Bay C, Johnson K, Fleming P. Short-term forecasting across a network for the autonomous wind farm. In: Proc. of the American Control Conf.; 2019:2837-2842.
21. Zaki A, Attia M, Hegazy D, Amin S. Comprehensive survey on dynamic graph models. *Int J Adv Comput Sci Appl*. 2016;7.
22. Shapiro CR, Meyers J, Meneveau C, Gayme DF. Dynamic wake modeling and state estimation for improved model-based receding horizon control of wind farms. In: American Controls Conference; 2017:709-716.
23. Starke GM, Meneveau C, King JR, Gayme DF. The Area Localized Coupled model for analytical mean flow prediction in arbitrary wind farm geometries. *J Renew Sustain Energy*. 2020;13:33305.
24. Shapiro CR, Ji C, Gayme DF. Real-time energy market arbitrage via aerodynamic energy storage in wind farms. In: Proc. of the American Control Conf.; 2020:4830-4835.
25. Farrell A, King J, Draxl C, Mudafort R, Hamilton N, Bay CJ, Fleming P, Simley E. Design and analysis of a wake model for spatially heterogeneous flow. *Wind Energy Sci*. 2021;6(3):737-758. https://wes.copernicus.org/articles/6/737/2021/
26. Bastankhah M, Porté-Agel F. A new analytical model for wind-turbine wakes. *Renew Energy*. 2014;70:116-123.
27. Johns Hopkins University. Lesgo: a parallel pseudo-spectral large-eddy simulation code. https://lesgo.me.jhu.edu; 2019.
28. Shapiro CR, Gayme DF, Meneveau C. Filtered actuator disks: theory and application to wind turbine models in large eddy simulation. *Wind Energy*. 2019;22:1414-1420.
29. Lignarolo LEM, Mehta D, Stevens RJAM, et al. Validation of four LES and a vortex model against stereo-PIV measurements in the near wake of an actuator disc and a wind turbine. *Renew Energy*. 2016;94:510-523.
30. Stevens RJAM, Martínez LA, Meneveau C. Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments. *Renew Energy*. 2018;116(Part A):470-478.
31. Martínez-Tossas LA, Churchfield MJ, Yilmaz AE, Sarlak H, Johnson PL, Sørensen JN, Meyers J, Meneveau C. Comparison of four large-eddy simulation research codes and effects of model coefficient and inflow turbulence in actuator-line-based wind turbine modeling. *J Renew Sustain Energy*. 2018;10(3):33301.
32. Stevens RJAM, Graham J, Meneveau C. A concurrent precursor inflow method for large eddy simulations and applications to finite length wind farms. *Renew Energy*. 2014;68:46-50.
33. Bou-Zeid E, Meneveau C, Parlange M. A scale-dependent lagrangian dynamic mod el for large eddy simulation of complex turbulent flows. *Phys Fluids*. 2005;17(2):25105.
34. Gebraad PMO, van Wingerden JW. A control-oriented dynamic model for wakes in wind plants. *J Phys: Confer Ser*. 2014;524:12186.
35. Brent RP. *Algorithms for Minimization Without Derivatives*, Prentice-Hall Series in Automatic Computation: Prentice-Hall; 1972.
36. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes: The Art of Scientific Computing*. 3rd ed: Cambridge University Press; 2007.
37. Van Rossum G, Drake FL. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.

How to cite this article: Starke GM, Meneveau C, King JR, Gayme DF. A dynamic model of wind turbine yaw for active farm control. *Wind Energy*. 2024;27(11):1302-1318. doi:10.1002/we.2884

APPENDIX A: THE ANALYTICAL CURLED WAKE MODEL INCLUDING YAW

The model is based on the Gaussian wake model[26] in which the wake velocity deficit is expressed as

$$\frac{\delta u}{U_\infty} = C(x)\exp\left[-\frac{(y-y_c)^2+(z-z_h)^2}{2\sigma^2}\right], \tag{A1}$$

where δu is the wake deficit, U_∞ is the freestream velocity, $C(x)$ is the initial velocity deficit, y_c is the y-location of the center of the wake, z_h is the turbine height, and σ is the width of the wake. If σ is only a function of x, one obtains the original Gaussian model.[3] However, yaw deforms the shape of the wake into a curled shape. As shown in Bastankhah et al,[6] the wake deformation can be represented by making the wake's "radius" dependent on the polar angle θ, that is, by using $\sigma(x,\theta)$, where θ is defined as

$$\theta = \arctan\left(\frac{z-z_h}{y-y_c}\right). \tag{A2}$$

The velocity deficit also depends on the spanwise position of the wake center, which is defined as

$$y_c = \frac{(\pi-1)|\hat{t}|^3 + 2\sqrt{3}\pi^2\hat{t}^2 + 48(\pi-1)^2|\hat{t}|}{2\pi(\pi-1)\hat{t}^2 + 4\sqrt{3}\pi^2|\hat{t}| + 96(\pi-1)^2}\widehat{sgn(t)} - \frac{2}{\pi}\frac{\hat{t}}{[(z+z_h)/\xi_0]^2-1}, \tag{A3}$$

where $\hat{t}$ is a dimensionless downstream travel time (it is a function of x) as specified by an approximate analytical solution provided in Bastankhah et al[6] and

$$\xi_0 \approx R\sqrt{A_*} \tag{A4}$$

is the initial wake radius based on the projected area of the yawed rotor with

$$A_* = \frac{1-a}{1-2a} = \frac{1+\sqrt{1-C_T\cos^2\gamma}}{2\sqrt{1-C_T\cos^2\gamma}}, \tag{A5}$$

where a is the turbine induction factor and is related to the disk velocity u_d by $u_d = U_\infty\cos\gamma\,(1-a)$. The induction factor is affected by the yaw of the turbine and relationship between the induction factor and the local thrust coefficient can be approximated[1] as

$$a = \frac{C_T'\cos^2(\gamma)}{4+C_T'\cos^2(\gamma)}. \tag{A6}$$

From this equation, the thrust coefficient can also be calculated based on the local thrust coefficient as

$$C_T = \frac{16C_T'}{(4+C_T'\cos^2(\gamma))^2}. \tag{A7}$$

This is the coefficient of thrust used in Equation (A5).

Once the wake streamwise centerline and the polar angle are known, we can calculate wake function as

$$\sigma(x,\theta) = k_w x + 0.4\xi(x,\theta), \tag{A8}$$

where the first term in the equation represents the standard linear wake expansion with coefficient k_w, and the second term represents the evolution of the wake shape. The function ξ is a complicated function that depends on the streamwise distance, the polar angle, the current yaw angle, the dimensionless time, and the rotation rate of the turbine blades. The function is obtained using a power-series expansion and solution to the vortex sheet transport equation. The solution is written as

$$\xi(x,\theta) = \xi_0\left(1 - \sum_{n=2}^{4}\left[c_n(\hat{t})\cos(n\theta) + d_n(\hat{t})\sin(n\theta)\right]\right), \tag{A9}$$

where expressions for the coefficients c_n and d_n in terms of the dimensionless time $\hat{t}$ (depending on x) are provided in Appendix B of Bastankhah et al.[6] These expressions include an empirical transition function that interpolates between different asymptotic behaviors for different regimes (near and far from the turbine).

Finally, to fully define this model, the initial wake deficit $C(x)$ is calculated using Equation (3) in the main text, for which the area $\tilde{\sigma}^2$ uses a simplified version of the original σ function, which avoids a dependence on θ by using the approximation for ξ defined in Equation (A4). It is evaluated as

$$\tilde{\sigma}^2(x) = (k_w x + 0.4\xi_0)(k_w x + 0.4\xi_0\cos\gamma), \tag{A10}$$

which uses Equation (A4), defined earlier. It includes the effect of yaw in modifying the wake width in the spanwise direction, but leaving the vertical width unchanged.

APPENDIX B: POWELL'S ALGORITHM

Powell's algorithm is a conjugate direction method that performs a minimization without calculating the derivatives. The algorithm is initialized using an initial point and an initial set of search vectors, represented by $s_1,\ldots,s_N$ for an N-dimensional problem. The method minimizes the function then iterates over the search directions, finding the minimum of the function in a line search over each search direction. The minima found in each bidirectional line search can be expressed as

$$x_0 + \alpha_1 s_1, x_0 + \sum_{i=1}^{2}\alpha_i s_i, \ldots, x_0 + \sum_{i=1}^{N}\alpha_i s_i, \tag{B1}$$

where x_0 is the initial point and α_i is the scalar that determines the minimum in the s_i line search direction. The next point is then expressed as

$$x_1 = x_0 + \sum_{i=1}^{N}\alpha_i s_i. \tag{B2}$$

A new direction is then added to the search directions, defined by $x_1 - x_0$, or $\sum_{i=1}^{N}\alpha_i s_i$, which can be thought of as the average direction moved during this iteration. A direction is also dropped from the search directions to keep the search directions of size N. In Powell's modified algorithm, which is what is used in Python, the direction that achieved the largest decrease in the bidirectional minimization is dropped, if it satisfies specific conditions. This is to decrease the chances of linear dependence in the search directions, since the search direction that caused the largest change will also be a significant part of the new search direction. The conditions for the search directions to be kept the same are if (1) the function shows no great improvement along the direction of greatest increase in the iteration, or (2) the direction of the greatest improvement does not account for a sufficiently large fraction of the new search direction, which would indicate that several directions are contributing to the minimization.

The algorithm is iterated until there is no significant improvement to the objective function.[35–37]