A Model-Free Approach to Wind Farm Control Using Game Theoretic Methods

Jason R. Marden, *Member, IEEE*, Shalom D. Ruben, *Member, IEEE*, and Lucy Y. Pao, *Fellow, IEEE*

***Abstract*— This brief explores the applicability of recent results in game theory and cooperative control to the problem of optimizing energy production in wind farms. One such result is a model-free control strategy that is completely decentralized and leads to efficient system behavior in virtually any distributed system. We demonstrate that this learning rule can provably maximize energy production in wind farms without explicitly modeling the aerodynamic interaction amongst the turbines.**

***Index Terms*— Cooperative systems, networked control systems, wind farms.**

I. Introduction

WIND energy is widely becoming recognized as one of the most cost-efficient sources of renewable energy. Accordingly, expectations for wind energy are at unprecedented levels. In the US, a goal has been set for wind energy capacity to meet 20% of the country's electrical energy demands by 2030 [2]. One of the keys to realizing this goal in a cost-efficient manner is to utilize existing wind farms in a more efficient manner through improved control algorithms.

Most research on the control of wind turbines has focused on the single-turbine setting [3], [4]. The control of an array of turbines in a wind farm is more challenging than controlling a single turbine because of the aerodynamic interactions amongst the turbines, which render most of these single-turbine control algorithms highly inefficient for optimizing power capture in wind farms [5]–[7]. The potential for improving performance, both in terms of increasing power capture as well as mitigating loads across the wind farm, has led to new research efforts in coordinating the control of arrays of wind turbines [8]–[14]. One approach for dealing with these aerodynamic interactions is to develop wake models for use in the distributed control algorithms [15]–[20]. However, the variable and chaotic nature of wind makes such a task incredibly challenging. An alternative approach, and the goal of this brief, is to develop an online control algorithm where each turbine adjusts its own axial induction factor in response to local information, such as the individual turbine's power generation, local wind conditions, or minimal information regarding neighboring turbines. The axial induction factor is a measure of the decrease in axial air velocity through the turbine and is related to the power the wind turbine extracts from the wind. Here, the goal is to develop a control algorithm that permits the set of turbines to reach a desirable set of axial induction factors that lead to good system level behavior, e.g., power maximization or load minimization, without the need for explicitly modeling the wind.

In this brief, we investigate the applicability of recent results in game theory and cooperative control for optimizing energy production in wind farms [21]–[23]. The field of game theory provides a framework for analyzing systems comprised of enmeshed decision makers. In wind farms, the decision makers represent the individual turbines, and the enmeshment follows from the fact that the decision of one turbine impacts the wind conditions and potential power generation by other turbines. The game theoretic framework is broad enough to model several phenomena that are relevant to wind farms including multiple and heterogeneous decision makers (e.g., turbines with variations in blade size), limited information in decision making (e.g., each turbine has limited information regarding the environment), environmental uncertainties (e.g., variability in wind conditions), and more.

One of the fundamental challenges associated with developing control strategies for the individual turbines in a wind farm is dealing with the following informational constraints.

1) Each turbine does not have access to the functional form of the power generated by the wind farm. This is because the aerodynamic interaction between the turbines is poorly understood.
2) Each turbine may not have access to the choices of other turbines. This is because of the lack of a suitable communication system.

Accordingly, the applicability of some of the common approaches to distributed optimization, e.g., subgradient methods [24], [25], are inapplicable because of these informational constraints. Recent research focuses on the use of genetic algorithms for wind farm optimization [26]. However, genetic algorithms do not typically provide any guarantees on convergence times or the quality of the solution.

The focus of this brief is on recent developments in game theoretic control for multiagent systems on the problem of wind farm optimization under the aforementioned informational constraints. We start by defining the wind farm model in Section II and a tractable example in Section III. Next, in Section IV we present two game theoretic distributed learning algorithms that can be utilized in distributed controllers to provide convergence to the collection of axial induction factors that optimize the power production in a wind farm. Both learning algorithms are model-free, which means that they do

Manuscript received January 8, 2012; revised February 27, 2013; accepted March 26, 2013. Manuscript received in final form April 4, 2013. Date of publication May 13, 2013; date of current version June 14, 2013. This work was supported in part by AFOSR under Grant FA9550-09-1-0538, the Center for Research and Education in Wind, and NSF under Grant CMMI-0700877. A preliminary version of this work appeared in [1]. Recommended by Associate Editor L. Fagiano.

J. R. Marden and L. Y. Pao are with the Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, CO 80309 USA (e-mail: jason.marden@colorado.edu; pao@colorado.edu).

S. D. Ruben is with the Department of Mechanical Engineering, University of Colorado, Boulder, CO 80309 USA (e-mail: shalom@colorado.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2013.2257780

not require a characterization of the aerodynamic interaction between the turbines to provide the desired convergence. The difference between the two algorithms centers on the amount of information available to the individual turbines. The first distributed learning algorithm, termed safe experimentation dynamics (SED) [27], requires each of the individual turbines to have knowledge regarding the total power produced in the wind farm. The second distributed learning algorithm, termed payoff-based distributed learning for Pareto optimality (PDLPO) [28], requires each of the individual turbines to only have knowledge regarding the power produced by the turbine itself and limited information regarding the behavior of neighboring turbines.

Lastly, in Section V we present several illustrations. In Section V-A, we focus on a simple three-turbine example and demonstrate that the proposed algorithms lead to a 7% increase in power produced by the wind farm when compared to the locally optimal controllers. Here, locally optimal controllers are the well-studied single-turbine controllers that seek to achieve an axial induction factor of $1/3$. In Section V-B, we provide simulation results on a more complex 80-turbine wind farm which replicates the layout of Horns Rev, an actual commercial offshore wind farm that is located in the North Sea off the coast of Denmark. Here, the proposed algorithms lead to a 25% increase in power produced by the wind farm when compared to the locally optimal controllers.

This brief presents an initial study into the applicability of game theoretic methods for wind farm control. To make the resulting analysis tractable, we consider fixed exogenous wind conditions and steady-state conditions with regard to the aerodynamic interaction between the turbines. Furthermore, we also assume the existence of control strategies for the individual wind turbines that can stabilize any axial induction factor between 0 and $1/3$.[1] While these assumptions are not necessarily realistic for actual wind farms, they serve to demonstrate the potential of game theoretic methods for multiturbine coordination for providing provable guarantees on system performance. Furthermore, it is important to point out that the presented algorithms were not altered for application to wind farm control; hence, there is a significant opportunity to improve upon the presented results by fine-tuning such algorithms for the problem of wind farm optimization. Lastly, it is worth noting that there are alternative game theoretic distributed learning algorithms that can accommodate relaxations in these assumptions while still providing provable guarantees on the system performance [27], [30], e.g., provide provable guarantees under varying wind conditions.

II. Wind Farm Model

We consider a wind farm consisting of n wind turbines denoted by the set $N = \{1, 2, \ldots, n\}$. For simplicity in initially developing and exploring our approach, we assume uniform wind of constant speed U_∞ and constant direction. Furthermore, we assume throughout that all wind turbines are oriented so that the turbine axes are parallel to the wind direction. Each turbine $i \in N$ is characterized by the diameter D_i of the disk generated by the turbine blades and a 2-D location (x_i, r_i) relative to a common vertex, where x_i is the distance from this vertex in the wind direction and r_i is the distance from this vertex in the orthogonal direction. We represent joint axial induction factors by the tuple $a = (a_1, a_2, \ldots, a_n)$ where a_i denotes the axial induction factor of turbine i. The set of admissible axial induction factors for turbine i is given by the set $\mathcal{A}_i = \{a_i : 0 \leq a_i \leq 0.5\}$, and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is the set of admissible joint axial induction factors. We will frequently express a joint axial induction factor $a \in \mathcal{A}$ by $a = (a_i, a_{-i})$, where $a_{-i} = (a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ represents the collection of axial induction factors of all wind turbines other than turbine i.

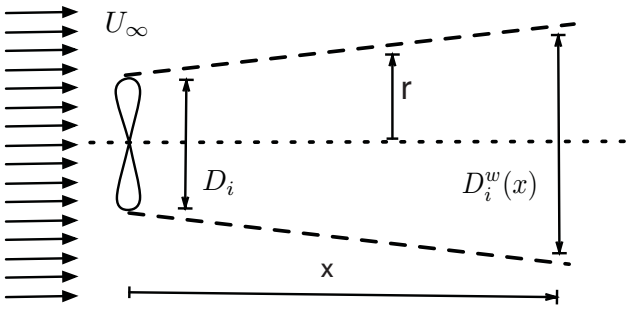


Fig. 1. Parameters for single-turbine wake model.

A. Wake Model

A wake model seeks to characterize the wake resulting from a single turbine. The well-studied Park model is one of the most prevalent wake models studied in the existing literature [26], [31]–[34]. Consider the situation highlighted in Fig. 1, where turbine $i \in N$ is the only turbine and let $V_i(x, r; a_i)$ represent the velocity profile of the wake generated by turbine i relative to the vertex $(x_i = 0, r_i = 0)$. According to the Park model, the velocity profile takes the form

$$V_i(x, r; a_i) = U_\infty \left(1 - \delta V_i(x, r; a_i)\right) \tag{1}$$

where $\delta V_i(x, r; a_i)$ represents the fractional deficit of the velocity at the point (x, r) downstream of turbine i

$$\delta V_i(x, r; a_i) = \begin{cases} 2a_i \left(\frac{D_i}{D_i + 2kx}\right)^2, & \text{for any } \; r \leq \frac{D_i + 2kx}{2} \\ 0, & \text{for any } r > \frac{D_i + 2kx}{2} \end{cases} \tag{2}$$

where k is a roughness coefficient.[2] The two dominant traits of this model are: 1) the velocity profile is constant along the radial direction of a wake, i.e., $V_i(x, r; a_i) = V_i(x, r'; a_i)$ for all $r, r' \leq (D_i + 2kx)/2$ and 2) the velocity approaches U_∞ at large distances from the turbine. According to (2), the diameter of the wake of turbine i at a distance x downstream is given by $D_i^w(x) = D_i + 2kx$.

[1] Empirical evidence on actual wind turbines suggests that it is possible to achieve axial induction factors different from $1/3$; however, to our knowledge there is only one initial study into the stability of such schemes [29].

[2] The roughness coefficient defines the slope at which the wake expands out from the turbine. Roughness coefficients have been found empirically for many different environments, e.g., $k = 0.075$ for farmlands and $k = 0.04$ for offshore locations [32].

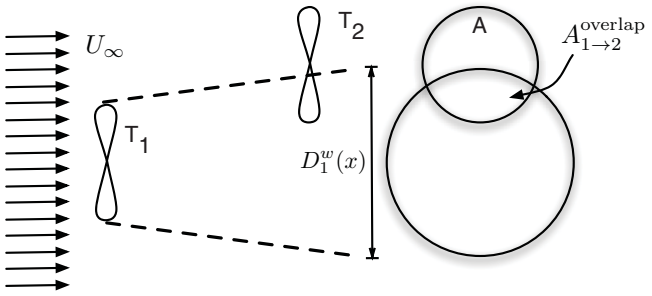


Fig. 2. Two-turbine example depicted above where the wind seen at turbine 2 is nonuniform. Using (4), the aggregate velocity deficit at turbine 2 is $\delta V_2(a) = 2a_1 \left(D_1/D_1 + 2k(x_2 - x_1)\right)^2 \left(A_{1\to 2}^{\text{overlap}}/A_2\right)$. Note that, if the wake of turbine 1 completely encompasses turbine 2, i.e., $A_{1\to 2}^{\text{overlap}} = A_2$, then we recover (2) as expected.

B. Wake Interaction Model

A core modeling challenge associated with the multiturbine setting is characterizing how overlapping wakes interact with one another. A common approach in the existing literature is that of momentum balance [31]. Rather than deriving an entire velocity profile, we use an aggregate wind velocity seen by each turbine $i \in N$, which we will represent by $V_1(a), \ldots, V_n(a)$, respectively. For any wind turbine $i \in N$, the aggregate wind velocity is given by

$$V_i(a) = U_\infty(1 - \delta V_i(a)) \tag{3}$$

where the aggregate velocity deficit seen by turbine i is

$$\delta V_i(a) = 2\sqrt{\sum_{j\in N: x_j < x_i} \left(a_j \left(\frac{D_j}{D_j + 2k(x_i - x_j)}\right)^2 \frac{A_{j\to i}^{\text{overlap}}}{A_i}\right)^2} \tag{4}$$

where A_i is the area of the disk generated by the blades of turbine i and $A_{j\to i}^{\text{overlap}}$ is the area of the overlap between the wake generated by turbine j and the disk generated by the blades of turbine i. See Fig. 2 for an illustration with two turbines. Accordingly, (3) takes the form

$$V_i(a) = U_\infty \left(1 - 2\sqrt{\sum_{j\in N: x_j < x_i} \left(a_j c_{ji}\right)^2}\right) \tag{5}$$

where

$$c_{ji} = \left(\frac{D_j}{D_j + 2k(x_i - x_j)}\right)^2 \frac{A_{j\to i}^{\text{overlap}}}{A_i}. \tag{6}$$

C. Power Model

We represent the control parameters of a wind turbine by the turbine's axial induction factor which represents the fractional decrease in wind velocity between the free stream conditions and those seen at the rotor plane. We parameterize our wind model by the axial induction factors as opposed to more traditional control parameters, e.g., tip-speed ratio and pitch angle, to provide a more compact representation of the wind farm model studied in this brief. More specifically, the power generated by turbine i is characterized by [35], [36]

$$P_i(a) = \frac{1}{2}\rho A_i C_P(a_i) V_i(a)^3 \tag{7}$$

where ρ is the density of air and $C_P(a_i)$ is the power efficiency coefficient which takes on the form

$$C_P(a_i) = 4a_i(1 - a_i)^2. \tag{8}$$

The total power generated in the wind farm is simply

$$P(a) = \sum_{i\in N} P_i(a).$$

III. Motivating Example

This brief focuses on developing wind turbine control strategies for maximizing power capture in a wind farm.[3] More specifically, we focus on the attainment of the optimal joint axial induction factor

$$a^{\text{opt}} \in \arg\max_{a\in\mathcal{A}} \; P(a).$$

It is important to note that most existing control strategies for wind turbines are geared at stabilizing an axial induction factor of $1/3$. The reason for this stems from (7) and (8), where we know that for any $a_{-i} \in \mathcal{A}_{-i} = \prod_{j\neq i} \mathcal{A}_j$ we have

$$1/3 = \arg\max_{a_i\in\mathcal{A}_i} \; P_i\left(a_i, a_{-i}\right).$$

We refer to this locally optimal control strategy as greedy, and we let $a^{\text{greedy}} = \{1/3, \ldots, 1/3\}$. Does this greedy control policy efficiently extend to the wind farm setting? More formally, how does the power production associated with a^{greedy} compare with the power production associated with a^{opt}?

To shed light on this question, we focus on the three-turbine wind farm illustrated in Fig. 3. In this setting, the power produced by the wind farm, $P(a)$, takes on the form

$$\frac{1}{2}\rho A \left(C_p(a_1)U_\infty^3 + C_p(a_2)V_2(a)^3 + C_p(a_3)V_3(a)^3\right) \tag{9}$$

where U_∞ is the upwind velocity and $A = A_1 = A_2 = A_3$. Solving for $V_2(a)$ and $V_3(a)$ using (5), we obtain

$$V_2(a) = U_\infty(1 - 2a_1c_{12}) \tag{10}$$

$$V_3(a) = U_\infty\left(1 - 2\sqrt{(a_1c_{13})^2 + (a_2c_{23})^2}\right). \tag{11}$$

Numerically optimizing (9) over the set of admissible joint axial induction factors $\mathcal{A}$ gives us

$$\{a_1^{\text{opt}}, a_2^{\text{opt}}, a_3^{\text{opt}}\} = \{0.232, 0.208, 0.333\}. \tag{12}$$

It is straightforward to verify that $a_3^{\text{opt}} = 1/3$, as there are no further turbines downstream. For this simple setting, we can attain a_1^{opt} and a_2^{opt} through an exhaustive search over the set $\mathcal{A}_1 \times \mathcal{A}_2$; however, it is important to note that such an approach is not tractable in general.[4] The efficiency of the locally optimal axial induction factors is

$$\frac{P(a^{\text{greedy}})}{P(a^{\text{opt}})} = 0.9265 \tag{13}$$

[3]It is important to note that there are alternative objectives in wind farm control, e.g., mitigating loads [35]. While we will not explicitly formulate such objectives, the control techniques developed in this brief are applicable for these settings as well.

[4]Note that for the optimal joint axial induction factor, the first turbine's axial induction factor is higher than the second turbine's axial induction factor. This phenomenon is consistent with observations made in other work on wind farm optimization, e.g., [26].

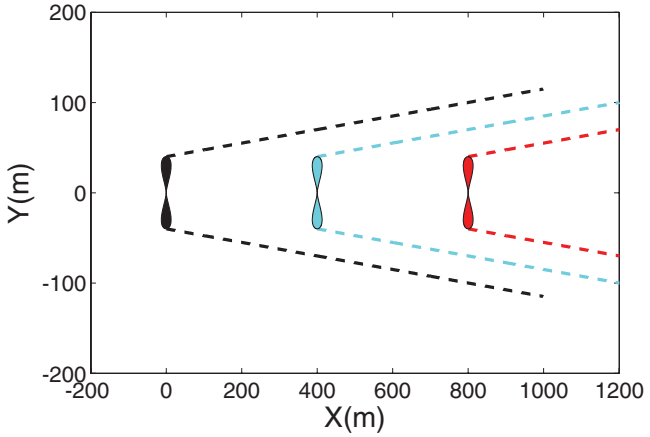


Fig. 3. Simple three-turbine wind farm considered in Section III. We denote the turbine on the left as turbine 1, the turbine in the middle as turbine 2, and the turbine on the right as turbine 3. Each turbine has a diameter of 80 m, and the turbines are spaced 400 m apart from one another. The roughness coefficient is $k = 0.075$. The wind is in the direction of the positive x-axis with a fixed upwind velocity U_∞. Note that according to (5), (7), and (8), the wind velocity U_∞ does not factor into the optimal profile of axial induction factors.

meaning that the efficiency loss is greater than 7%. The reason for this degradation is that the locally optimal axial induction factors do not account for the aerodynamic interactions between the turbines.

IV. Wind Farm Control

The previous section showed that control algorithms for wind turbines in the single-turbine setting do not efficiently extend to the multiturbine setting. This section explores approaches for developing control algorithms for this multiturbine setting, where the objective is to maximize the total power production in a wind farm given fixed exogenous wind conditions. The challenge with such an objective is dealing with the facts that: 1) the aerodynamic interaction between the turbines is not well characterized and 2) the information available to each of the wind turbines may be limited. Nonetheless, we will show that these limitations can be overcome to meet our desired objectives.

A. Preliminaries: Cooperative Control

In this section, we formulate the problem of wind farm optimization as a cooperative control problem. The forthcoming control designs establish an interaction framework that produces a sequence of joint axial induction factors $a(0)$, $a(1), \ldots$, where at each iteration $t \in \{0, 1, 2, \ldots\}$ the decision of each turbine $i \in N$ is chosen independently according to a local control law of the form

$$a_i(t) = \Pi_i(\text{Information available to turbine } i \text{ at iteration } t). \tag{14}$$

The control policy of turbine i, $\Pi_i(\cdot)$, designates how each turbine processes available information to formulate a decision at each iteration. We will refer to a turbine's axial induction factor as the turbine's action or decision. The goal is to design the local control policies $\{\Pi_i(\cdot)\}_{i\in N}$ within the desired informational constraints such that the collective behavior converges to a collection of axial induction factors a^{opt} that optimizes the total power production in the wind farm, i.e., $a^{\text{opt}} \in \arg\max_{a\in\mathcal{A}} P(a)$. Here, we focus on the design of turbine control policies that are model-free, i.e., control policies that do not rely on characterization of the aerodynamic interaction between the turbines.

B. Model-Free With Communication

For the full communication setting, we focus on the design of local turbine control policies in (14) of the form

$$a_i(t) = \Pi_i\left(\{a_i(\tau), P(a(\tau))\}_{\tau=0,1,\ldots,t-1}\right). \tag{15}$$

Accordingly, the decision of turbine i at any iteration $t > 0$ is able to depend on: 1) the axial induction factor of turbine i at any previous iteration $\tau \leq t-1$ and 2) the power produced by the wind farm at any previous iteration $\tau \leq t-1$. Note that turbine i does not have access to the axial induction factors of the other turbines at any iteration $\tau \leq t-1$, i.e., $a_{-i}(\tau)$, or the structural form of $P(\cdot)$.

We will now introduce SED presented in [27]. SED requires that the set of axial induction factors for each turbine be a discretized set. In SED, each turbine $i \in N$ possesses a local state variable that impacts on the turbine's control policy. We represent a turbine's state by the tuple $[\bar{a}_i, \bar{p}_i]$, where:

1) the benchmark action is $\bar{a}_i \in \mathcal{A}_i$ and
2) the benchmark power is $\bar{p}_i$, which is in the range of $P(\cdot)$.

SED proceeds as follows.

1) *Initialization:* At iteration $t = 0$, each turbine $i \in N$ randomly selects an axial induction factor $a_i(0) \in \mathcal{A}_i$. This will be initially set as the turbine's baseline action at iteration $t = 1$ and is denoted by $\bar{a}_i(1) = a_i(0)$. The turbine's baseline power at iteration $t = 1$ is given by $\bar{p}_i(1) = P(a(0))$.
2) *Action Selection:* At each subsequent iteration, each turbine selects the baseline action with probability $(1-\epsilon)$ or experiments with a new random action with probability ϵ

$$a_i(t) = \begin{cases} \bar{a}_i(t) & \text{with probability } (1-\epsilon), \\ \text{RAND} & \text{with probability } \epsilon \end{cases} \tag{16}$$

where $\epsilon > 0$ will be referred to as the turbine's exploration rate and RAND represents that $a_i(t)$ is chosen randomly according to a uniform distribution over the set $\mathcal{A}_i$.[5]
3) *State Update:* Each turbine $i \in N$ updates the baseline action according to

$$\bar{a}_i(t+1) = \begin{cases} a_i(t), & P(a(t)) > \bar{p}_i(t) \\ \bar{a}_i(t), & P(a(t)) \leq \bar{p}_i(t) \end{cases}$$

[5]In general, uniformity is not necessary to provide the asymptotic guarantees given in Theorem 1 [27].

and the baseline power according to

$$\bar{p}_i(t+1) = \max\left\{P(a(t)),\, \bar{p}_i(t)\right\}.$$

This step is performed whether or not Step 2) is involved in experimentation.

4) Return to Step 2) and repeat.

This learning algorithm is called SED since $P(\bar{a}(t))$ is nondecreasing with respect to the iterations, i.e., the power generated by the wind farm when using the baseline action is nondecreasing. We now state the following characterization on the limiting behavior associated with the SED [27].

Theorem 1: Suppose all turbines use SED as highlighted above. Given any probability $p < 1$, if the exploration rate $\epsilon > 0$ is sufficiently small, then for all sufficiently large iterations t, $a(t) \in \arg\max_{a\in\mathcal{A}} P(a)$ with at least probability p.

There are several important properties regarding the applicability of Theorem 1 to wind farm optimization. First, Theorem 1 guarantees that the average power produced in the wind farm converges to the optimal power that could be produced given the current wind conditions

$$\lim_{\epsilon\to 0^+}\lim_{t\to\infty}\left(\frac{1}{t}\sum_{\tau=0}^{t-1} P(a(\tau))\right) = \max_{a\in\mathcal{A}} P(a). \tag{17}$$

Second, note that there is no underlying dependence on a given wind model and wake interaction model. Accordingly, the above characterization holds for any setting which is of fundamental importance since developing accurate wind models, and wake interaction models is an active research area [8]–[12], [14]. Third, it is important to highlight that the characterization provided in Theorem 1 refers to probabilistic convergence as opposed to almost sure convergence. This means that the joint axial induction factors will not converge to the optimal axial induction factors. Rather, the individual wind turbines will spend most of the time using the optimal axial induction factors. The reason for this is that the individual turbines do not have access to the structural form of $P(a)$. Therefore, the turbines perpetually probe the system, albeit with small probability, to gain information. Lastly, there are extensions of the presented algorithm for scenarios with nondeterministic power production functions $P(\cdot)$ [27]. This nondeterministic case may yield better performance in wind farm settings with nonstatic wind conditions.

C. Model-Free With Limited Communication

For the limited communication setting, we focus on the design of local turbine control policies in (14) of the form

$$a_i(t) = \Pi_i\left(\left\{\left\{a_j(\tau)\right\}_{j\in N_i}, P_i\left(a(\tau)\right)\right\}_{\tau=0,1,\ldots,t-1}\right) \tag{18}$$

where $N_i \subseteq N$ is the neighbor set of turbine i. We assume throughout that $i \in N_i$ for all turbines $i \in N$. Accordingly, the decision of turbine i at any iteration $t > 0$ is able to depend on: 1) the axial induction factor of any turbine $j \in N_i$ at any previous iteration $\tau \le t-1$ and 2) the power produced by turbine i at any previous iteration $\tau \le t-1$. In this setting, turbine i no longer has access to information pertaining to the power produced by all other turbines at any iteration $\tau \le t-1$, i.e., $\{P_j(a(\tau))\}_{j\neq i}$, and thus the aforementioned SED are no longer admissible.

We now present the algorithm PDLPO introduced in [28]. As with the SED, each turbine $i \in N$ possesses a local state variable which impacts the turbine's control policy. Now, at each point in time a turbine's state is represented by the triple $[\vec{a}_i, \bar{p}_i, m_i]$ defined as follows.

1) The benchmark action vector is $\vec{a}_i \in \prod_{j\in N_i}\mathcal{A}_j$.
2) The benchmark power is $\bar{p}_i$, which is in the range of $P_i(\cdot)$.
3) The mood is m_i, which can take on two values, content (C) and discontent (D).

The learning algorithm produces a sequence of action profiles $a(1), \ldots, a(t)$, where the behavior of turbine i at each iteration $t = 1, 2, \ldots$, is conditioned on turbine i's underlying benchmark power $\bar{p}_i(t)$, benchmark action $\bar{a}_i(t)$, and mood $m_i(t) \in \{C, D\}$.

We divide the dynamics into two parts: the turbine dynamics and the state dynamics. To simplify the forthcoming presentation, we will present the algorithm under the assumption that $1 > P_i(a) \ge 0$ for all $a \in \mathcal{A}$.

Turbine Dynamics: Fix an experimentation rate $\epsilon > 0$ and constant $c \ge n$.[6] Let $\bar{x}_i(t) = [\vec{a}_i, \bar{p}_i, m_i]$ be the current state of turbine i at iteration t.

1) *Content* ($m_i = C$)*:* In this state, the turbine chooses an axial induction factor at iteration t, $a_i(t)$, according to the following probability distribution:

$$\Pr\left[a_i(t) = a_i | \bar{x}_i(t)\right] = \begin{cases} \frac{\epsilon^c}{|\mathcal{A}_i|-1}, & \text{for } a_i \neq \vec{a}_i^i, \\ 1-\epsilon^c, & \text{for } a_i = \vec{a}_i^i \end{cases} \tag{19}$$

where $|\mathcal{A}_i|$ represents the cardinality of the set $\mathcal{A}_i$, and $\vec{a}_i^i$ represents the action of turbine i in the vector $\vec{a}_i$.

2) *Discontent* ($m_i = D$)*:* In this state, turbine i chooses an action a_i according to the following probability distribution

$$\Pr\left[a_i(t) = a_i | \bar{x}_i(t)\right] = \frac{1}{|\mathcal{A}_i|} \quad \text{for every} \quad a_i \in \mathcal{A}_i. \tag{20}$$

Note that the benchmark action and benchmark power production levels play no role in the turbine dynamics when the turbine is discontent.

State Dynamics: Let $a_{N_i} = \{a_j(t) : j \in N_i\}$ represent the actions of the neighbors of turbine i at time t, and $p_i = P_i(a(t))$ represent the power produced by turbine i given the profile $a(t)$. The state is updated as follows.

1) *Content* ($m_i = C$)*:* If $[a_{N_i}, p_i] = [\vec{a}_i, \bar{p}_i]$, where the equality means that the two expressions are equivalent component by component, the new state is determined by the transition

$$[\bar{a}_i, \bar{p}_i, C] \overset{[a_{N_i}, p_i]}{\longrightarrow} [\bar{a}_i, \bar{p}_i, C]. \tag{21}$$

[6]The forthcoming result only requires $c \ge \max_{a\in\mathcal{A}}\left(\sum_{i\in N} P_i(a)\right)$.

If $[a_{N_i}, p_i] \neq [\bar{a}_i, \bar{p}_i]$, the new state is determined by the transition

$$[\bar{a}_i, \bar{p}_i, C] \overset{[a_{N_i}, p_i]}{\longrightarrow} \begin{cases} [a_i, p_i, C] \text{ with prob } \epsilon^{1-p_i} \\ [a_i, p_i, D] \text{ with prob } 1-\epsilon^{1-p_i}. \end{cases}$$

2) *Discontent* ($m_i = D$)*:* If the selected action and power produced are $[a_i, p_i]$, the new state is determined by the transition

$$[\bar{a}_i, \bar{p}_i, D] \overset{[a_{N_i}, p_i]}{\longrightarrow} \begin{cases} [a_i, p_i, C] \text{ with prob } \epsilon^{1-p_i} \\ [a_i, p_i, D] \text{ with prob } 1-\epsilon^{1-p_i}. \end{cases}$$

We now state the following characterization on the limiting behavior associated with the algorithm PDLPO as provided in [28]. Before stating the theorem, we introduce the following assumption on the neighbor sets.

Assumption 1 (Interdependence of Neighbor Sets): The neighbor sets $\{N_i\}_{i \in N}$ are interdependent if the graph with nodes N and edges $\{i, j \in N : j \in N_i\}$ is connected.

Theorem 2: Suppose all turbines use the PDLPO as highlighted above and that the neighbor sets satisfy Assumption 1. Given any probability $p < 1$, if the exploration rate $\epsilon > 0$ is sufficiently small, then for all sufficiently large times t, $a(t) \in \arg\max_{a \in \mathcal{A}} P(a)$ with at least probability p.

There are several important properties regarding the applicability of Theorem 2 to the problem of wind farm optimization. As with the SED, we attain probabilistic convergence as opposed to almost sure convergence. However, we do so using control policies that rely only on local information since in this setting no turbine has access to the power generated by the wind farm. Accordingly, both the presented algorithms optimize total power production without requiring knowledge of the aerodynamic interactions between the turbines. The distinction between the two algorithms is the communication. In the SED, each turbine has access to information regarding the power generated by all turbines in the wind farm. This is in contrast to PDLPO, where each turbine only has access to information regarding the power generated by the turbine itself and the behavior of neighboring turbines.[7]

V. Simulation Results

We now present several illustrations of the distributed control algorithm presented in the previous section on the problem of wind farm optimization. We first focus on the simple three-turbine row farm example from Section III and will then proceed to a more involved 80-turbine wind farm example which replicates the layout of Horns Rev.

A. Three-Turbine Example

Recall the three-turbine wind farm illustrated in Fig. 3. As shown previously in (13), the efficiency loss associated with the locally optimal controllers was over 7% when compared with the globally optimal controllers. While this 7% was evaluated focusing on a specific wind model and wake interaction model, i.e., Park model with momentum balance, it seems realistic that alternative models, in addition to reality, would see similar deficiencies. Ultimately, it is imperative that the underlying control strategy accounts for the aerodynamic interaction between the turbines.

1) Full Communication: We simulated the SED learning algorithm on this three-turbine wind farm example. The model parameters are $k = 0.075$, $\rho = 1.225$ (kg/m^3), and $U_\infty = 8$ (m/s). The results are presented in Fig. 4(a) and (b) for the exploration rate parameter $\epsilon = 0.05$, discretized action sets of the form $\mathcal{A}_i = [0.1{:}\,0.01{:}\,0.33]$, and an initial profile of axial induction factors $a(0) = \{0.33, 0.33, 0.33\}$. The simulation results demonstrate that the power produced by the wind farm rapidly approaches the optimal power that could be produced by the wind farm for the given wind conditions. Furthermore, the axial induction factors quickly approach the optimal axial induction factors identified in (12). It is important to emphasize that this algorithm was effectively able to optimize system performance without exploiting any information regarding the underlying wind model or wake interaction model.

2) Limited Communication: In the full communication setting above, each wind turbine $i \in N$ has information regarding the power produced by the wind farm, i.e., $P(a) = \sum_{i \in N} P_i(a)$ for any joint axial induction factor $a \in \mathcal{A}$. We now focus on the PDLPO learning algorithm, which does not require such informational demands. Recall that the algorithm requires that the turbine power levels satisfy $1 > P_i(a) \geq 0$ for all $a \in \mathcal{A}$. To achieve this, we define a normalized power for each turbine

$$P_i^{\text{NORM}}(a) = \frac{0.915}{z}\left(P_i(a) - \min_{a \in \mathcal{A}} P_i(a)\right) \tag{22}$$

where

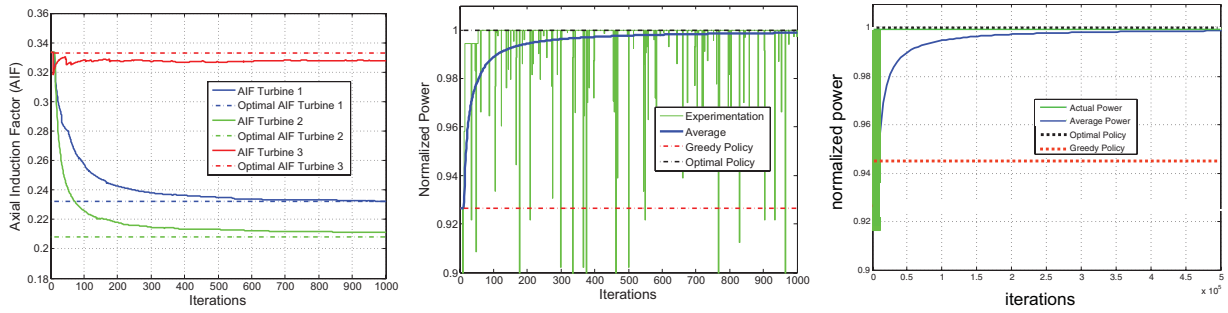
$$z = \max_{i \in N, a, a' \in \mathcal{A}} P_i(a) - P_i(a').$$

Note that this scaling ensures that the normalized power functions satisfy $1 > P_i^{\text{NORM}}(a) \geq 0$ for any $a \in \mathcal{A}$. Furthermore, the proposed normalizing does not change the optimal axial inductions factors

$$\arg\max_{a \in \mathcal{A}} \sum_{i \in N} P_i^{\text{NORM}}(a) = \arg\max_{a \in \mathcal{A}} \sum_{i \in N} P_i(a).$$

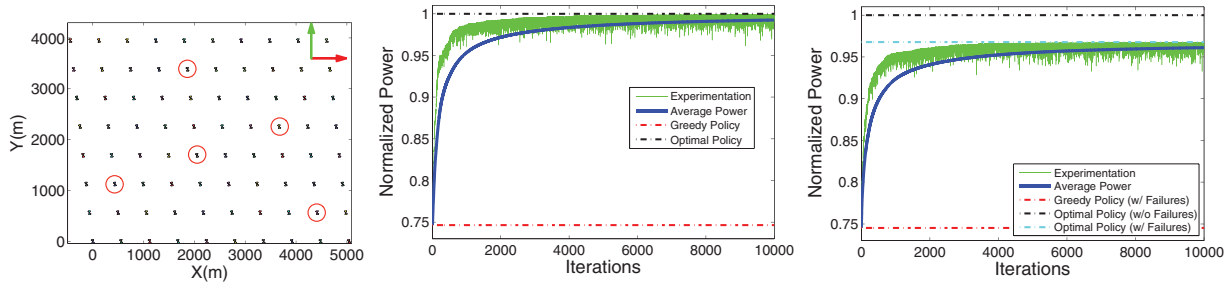
Fig. 4(c) shows the evolution of the power produced by the wind farm using the PDLPO learning algorithm. The model parameters are $k = 0.075$, $\rho = 1.225$ (kg/m^3), and $U_\infty = 8$ (m/s). The algorithm parameters were set as $c = 2.2$ and $\epsilon = 0.001$, the neighbor sets were set as $N_1 = \{1, 2\}$, $N_2 = \{1, 2, 3\}$, and $N_3 = \{2, 3\}$, the action sets were discretized as $\mathcal{A}_i = \{0.2, 0.32\}$, the initial profile of axial induction factors was $a(0) = \{0.2, 0.2, 0.2\}$, and each turbine was initially in the discontent state. Here, the greedy policy was chosen as $(0.32, 0.32, 0.32)$ while the optimal policy was selected over the discretized set $\mathcal{A}$ and was of the form

[7] It is important to note that the presented algorithm differs slightly from the presentation in [28]. The key differences between the two algorithms is that in [28] we have $N_i = \{i\}$ for all turbines $i \in N$. However, the turbines' power functions must satisfy some notion of interdependence. Whether this interdependence condition is satisfied or not in wind farms remains an open question and warrants further studies. However, by defining connected neighbor sets satisfying Assumption 1, this interdependence condition is immediately satisfied, irrespective of the structure of the turbines' power functions, and the proof set forth in [28] follows immediately.



(a) (b) (c)

Fig. 4. Simulation results on the three-turbine wind farm. (a) Evolution of axial induction factors and (b) evolution of normalized power provide simulation results of the full information (FI) SED algorithm. The simulation results demonstrate that the SED learning algorithm can optimize the power produced in the wind farm without an explicit model of the aerodynamic interaction between the turbines. (c) Evolution of normalized power provides simulation results for the limited-information (LI) PDLPO algorithm. The performance is not as good as the performance associated with the SED learning algorithm, which is to be expected as this learning algorithm places additional restrictions on the amount of information available to the individual turbines.



(a) (b) (c)

Fig. 5. Simulation results on an 80-turbine wind farm that replicating the Horns Rev wind farm located in the North Sea off the coast of Denmark [37] which is composed of Vestas V80 2-MW turbines, each with a diameter of 80 m. (a) Layout of the wind farm which is in an oblique rectangle with a turbine spacing of 7 turbine diameters, i.e., 560 m, in both the x- and y-directions. The positive x-direction represents due east, and the positive y-direction represents due north. (b) and (c) Simulation results of the SED. (b) Case when all turbines are operational. Here, the greedy policy produces around 74.6% of the potential wind power production. When using SED, after 1000 iterations, the average power produced is over 95% of the potential wind power. (c) Case when the circled turbines in (a) are not operational. Since 5 turbines are not operational, the optimal power production from the remaining 75 turbines is approximately 96.8% of optimal power production when all turbines are operational. When using SED, the average power produced by this wind farm quickly approaches the optimal power production possible. Note that while the system perpetually experiments, as expected, the average performance is not impacted by these experimentations. Here, we assumed that the failures were present at the start of the algorithm.

$a^{\text{opt}} = (0.2, 0.2, 0.32)$. While the simulation results demonstrate that the power produced by the wind farm approaches the optimal power, the time required to reach this configuration is far greater than that required by the SED learning algorithm. This is to be expected as the payoff-based learning algorithm placed additional restrictions on the amount of information available to the individual turbines.[8]

B. Horns Rev Example

This section presents a study on a more complex wind farm which replicates the 80-turbine Horns Rev wind farm in Denmark. The wind farm configuration is illustrated in Fig. 5(a). We simulated the SED learning algorithm on this 80-turbine wind farm when all turbines are operational and also when there are turbine failures. The model parameters were set as $k = 0.04$, $\rho = 1.225$ (kg/m^3), and $U_\infty = 8$ (m/s) in the eastward direction. Here, we used $k = 0.04$ to model offshore conditions. The exploration rate parameter is set as $\epsilon = 0.03$, with discretized action sets of the form $\mathcal{A}_i = [0{:}0.01{:}0.33]$, and initial axial induction factors of $a_i(0) = 0.33$ for each turbine $i \in N$. Fig. 5(b) and 5(c) present simulation results which demonstrate that the power produced by the wind farm quickly approaches the optimal power that could be produced by the wind farm for the given conditions. Fig. 5(b) focuses on the setting when all turbines are operational. Here, the degradation in performance associated with the greedy control policy when compared to the optimal control policy is over 25%. Fig. 5(c) focuses on the setting when the circled turbines in Fig. 5(a) are not operational. These results demonstrate that the proposed control algorithms are robust to operational failures.

[8]It is important to highlight that Fig. 4(c) is not necessarily indicative of true steady-state behavior with regard to the proposed dynamics. Rather, this figure provides an illustration of a specific run of the dynamics. This illustration highlights that the turbines initially start in a fully discontent state and then eventually settle on the optimal action profile. It is important to highlight that the turbines will eventually become discontent again and the process will repeat. The characterization in Theorem 2 provides the desired results when $\epsilon \to 0$. However, as $\epsilon \to 0$, the time required to reach steady state converges to ∞ which is obviously not desirable from a system-level perspective.

VI. Conclusion

This brief initiated a study on optimizing power production in wind farms using model-free algorithms. By model-free, we mean an algorithm that does not utilize a model of the aerodynamic interaction between the turbines. We presented two model-free distributed learning algorithms from the game

theoretic literature which could be employed to provably optimize power production in wind farms. We simulated both learning algorithms on two simplistic wind farm examples and observed efficiency gains in upwards of 25% when compared to the greedy algorithm. Nonetheless, it is important to highlight that the asymptotic guarantees associated with the presented algorithms placed strong assumptions on the wind farm conditions which are unrealistic. These include controlling the axial induction factor, invariant wind conditions, and a focus on purely steady-state behavior. It is imperative to understand how the presented algorithms, or variants thereof, extend to more realistic wind farm conditions. Such assessments can initially be conducted by simulations on more realistic wind farm models, e.g., [38]. A second opportunity for research is to identify the importance of aerodynamic models for wind farm optimization. While the presented algorithms did not require such models, this limitation ultimately hampered their performance. Accordingly, a key area of future research is assessing what level of model precision is necessary to see significant improvements in the behavior of such algorithms.

REFERENCES

[1] J. R. Marden, S. D. Ruben, and L. Y. Pao, "Surveying game theoretic approaches for wind farm optimization," in *Proc. AIAA Aerosp. Sci. Meeting*, Jan. 2012, pp. 1–10.

[2] *20% Wind Energy by 2030: Increasing Wind Energy's Contribution to U.S. Electricity Supply*, U.S. Dept. Energy, Washington, DC, USA, 2008.

[3] L. Pao and K. Johnson, "Control of wind turbines: Approaches, challenges, and recent developments," *IEEE Control Syst. Mag.*, vol. 31, no. 2, pp. 44–62, Apr. 2011.

[4] K. Johnson, J. W. van Wingerden, M. J. Balas, and D.-P. Molenaar, "Structural control of floating wind turbines," *Mechatronics*. vol. 21, no. 4, pp. 704–719, Jun. 2011.

[5] K. E. Johnson and N. Thomas, "Wind farm control: Addressing the aerodynamic interaction among wind turbines," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 2104–2109.

[6] R. J. Barthelmie and L. E. Jensen, "Evaluation of wind farm efficiency and wind turbine wakes at the Nysted offshore wind farm," *Wind Energy*, vol. 13, no. 6, pp. 573–586, 2010.

[7] M. Steinbuch, W. W. de Boer, O. H. Bosgra, S. Peters, and J. Ploeg, "Optimal control of wind power plants," *J. Wind Eng. Ind. Aerodyn.*, vol. 27, nos. 1–3, pp. 237–246, Jan. 1988.

[8] M. Kristalny and D. Madjidian, "Decentralized feedforward control of wind farms: Prospects and open problems," in *Proc. Joint IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 3464–3469.

[9] V. Spudic, M. Baotic, M. Jelavic, and N. Peric, "Hierarchical wind farm control for power/load optimization," in *Proc. Conf. Sci. Making Torque from Wind*, 2010, pp. 681–692.

[10] V. Spudic, M. Jelavic, and M. Baotic, "Wind turbine power control for coordinated control of wind farms," in *Proc. 18th Int. Conf. Process Control*, Jun. 2011, pp. 463–468.

[11] A. J. Brand, "A quasi-steady wind farm control model," in *Proc. Eur. Wind Energy Conf.*, Mar. 2011, pp. 1–40.

[12] M. Soleimanzadeh and R. Wisniewski, "Controller design for a wind farm, considering both power and load aspects," *Mechatronics*, vol. 21, no. 4, pp. 720–727, Jun. 2011.

[13] M. Soleimanzadeh, R. Wisniewski, and S. Kanev, "An optimization framework for load and power distribution in wind farms," *J. Wind Eng. Ind. Aerodyn.*, vols. 107–108, pp. 256–262, Aug.–Sep. 2012.

[14] D. Madjidian and A. Rantzer, "A stationary turbine interaction model for control of wind farms," in *Proc. 18th IFAC World Congr.*, Aug. 2011, pp. 1–6.

[15] O. Rathmann, S. Frandsen, and R. Barthelmie, "Wake modelling for intermediate and large wind farms," in *Proc. Eur. Wind Energy Conf.*, 2007, pp. 1–10.

[16] A. Duckworth and R. J. Barthelmie, "Investigation and validation of wind turbine wake models," *Wind Eng.*, vol. 32, no. 5, pp. 459–475, 2008.

[17] T. Knudsen, M. Soltani, and T. Bak, "Prediction models for wind speed at turbines in a farm with application to control," in *Proc. Euromech Colloq. Conf.*, Oct. 2009, pp. 583–590.

[18] S. Ivanell, J. N. Sorensen, R. F. Mikkelsen, and D. Henningson, "Analysis of numerically generated wake structures," *Wind Energy*, vol. 12, no. 1, pp. 63–80, 2009.

[19] N. Troldborg, J. N. Sorensen, and R. F. Mikkelsen, "Numerical simulations of wake characteristics of a wind turbine in uniform inflow," *Wind Energy*, vol. 13, no. 1, pp. 86–99, 2010.

[20] P. Torres, J.-W. van Wingerden, and M. Verhaegen, "Modeling of the flow in wind farms for total power optimization," in *Proc. IEEE Int. Conf. Control Autom.*, Dec. 2011, pp. 963–968.

[21] J. R. Marden, G. Arslan, and J. S. Shamma, "Connections between cooperative control and potential games," *IEEE Trans. Syst., Man Cybern., B, Cybern.*, vol. 39, no. 6, pp. 1393–1407, Dec. 2009.

[22] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game theoretical formulation," *ASME J. Dyn. Syst., Meas. Control*, vol. 129, no. 5, pp. 584–596, Sep. 2007.

[23] R. Gopalakrishnan, J. R. Marden, and A. Wierman, "An architectural view of game theoretic control," *ACM SIGMETRICS Perform. Evaluation Rev.*, vol. 38, no. 3, pp. 31–36, 2011.

[24] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[25] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2506–2517, Nov. 2009.

[26] A. Scholbrock, "Optimizing wind farm control strategies to minimize wake loss effects," M.S. thesis, Dept. Mech. Eng., Environ. Eng., Univ. Colorado Boulder, Boulder, CO, USA, Jul. 2011.

[27] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, "Payoff based dynamics for multi-player weakly acyclic games," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 373–396, Feb. 2009.

[28] J. R. Marden, H. P. Young, and L. Y. Pao. (2011). *Achieving Efficiency in Distributed Systems* [Online]. Available: http://ecee.colorado.edu/marden/publications/

[29] A. D. Buckspan, L. Y. Pao, J. P. Aho, and P. Fleming, "Stability analysis of a wind turbine active power control system," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 1–6.

[30] H. P. Young, *Strategic Learning and Its Limits*. London, U.K.: Oxford Univ. Press, 2005.

[31] I. Katic and N. O. Jensen, "A simple model for cluster efficiency," in *Proc. Eur. Wind Energy Conf.*, 1986, pp. 407–410.

[32] T. Sorensen, M. L. Thogersen, P. Nielsen, A. Grotzner, and S. Chun, "Adapting and calibration of existing wake models to meet the conditions inside offshore wind farms," EMD Int. A/S, Aalborg, Denmark, Rep. 5899, 2008.

[33] J. D. Grunnet, M. Soltani, T. Knudsen, M. Kragelund, and T. Bak, "Aeolus toolbox for dynamic wind farm model, simulation and control," in *Proc. Eur. Wind Energy Conf.*, 2010, pp. 3119–3129.

[34] T. Stovall, G. Pawlas, and P. Moriarty, "Wind farm wake simulations in OpenFOAM," in *Proc. AIAA Aerosp. Sci. Meeting*, Jan. 2010, p. 80309.

[35] J. F. Manwell, J. G. McGowan, and A. Rogers, *Wind Energy Explained: Theory, Design and Application*. New York, NY, USA: Wiley, 2002.

[36] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, *Wind Energy Handbook*. New York, NY, USA: Wiley, 2001.

[37] R. Barthelmie, S. Frandsen, K. Hansen, J. Schepers, K. Rados, W. Schlez, A. Neubert, L. Jensen, and S. Neckelmann, "Modelling the impact of wakes on power output at Nysted and Horns Rev," in *Proc. Eur. Wind Energy Conf.*, 2009, pp. 1–10.

[38] M. J. Churchfield and S. Lee. (2012, Mar.). *Nwtc Design Codes: Simulator for Offshore Wind Farm Applications (SOWFA)* [Online]. Available: http://wind.nrel.gov/designcodes/simulators/SOWFA/