

Data-Driven Decentralized Algorithm for Wind Farm Control with Population-Games Assistance

Julian Barreiro-Gomez ¹, Carlos Ocampo-Martinez ^{2,*}, Fernando D. Bianchi ³ and Nicanor Quijano ⁴

¹ Learning & Game Theory Laboratory, New York University Abu Dhabi, Saadiyat Campus, PO Box 129188, Abu Dhabi, 129188 UAE; jbarreiro@nyu.edu

² Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain; carlos.ocampo@upc.edu

³ Instituto Balseiro and CONICET, Av. Bustillo 9500, San Carlos de Bariloche 8400, Argentina; fernando.bianchi@ib.edu.ar

⁴ Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1A No 18A-10, Bogotá 111711, Colombia; nquijano@uniandes.edu.co

* Correspondence: jbarreiro@nyu.edu

Abstract: In wind farms, the interaction between turbines that operate close by experience some problems in terms of their power generation. Wakes caused by upstream turbines are mainly responsible of these interactions, and the phenomena involved in this case is complex especially when the number of turbines is high. In order to deal with these issues, there is a need to develop control strategies that maximize the energy captured from a wind farm. In this work, an algorithm that uses multiple estimated gradients based on measurements that are classified by using a simple distributed population-games-based algorithm is proposed. The update in the decision variables is computed by making a superposition of the estimated gradients together with the classification of the measurements. In order to maximize the energy captured and maintain the individual power generation, several constraints are considered in the proposed algorithm. Basically, the proposed control scheme reduces the communications needed, which increases the reliability of the wind farm operation. The control scheme is validated in simulation in a benchmark corresponding to the Horns Rev wind farm.

Keywords: data-driven control strategy; wind turbines

1. Introduction

Nowadays, it is quite strange to find wind turbines operating isolated into a geographical scheme. Conversely, they are conveniently arranged in groups known as wind farms, which inject power into the electrical grid in a way and magnitude comparable to non-renewable energy sources.

Initially, control strategies designed for wind farms were mainly based on aggregate models, which represent those arrays as a large equivalent wind turbine [1–4]. The drawback of this modeling approach relies on the lack of interactions among turbines caused by the wakes. A turbine located in the path of wakes produced by up-stream turbines suffers from the reduction of arriving wind speed and will probably be exposed to a more turbulent air flow [5]. Consequently, using a unique control signal for all the turbines into the farm will not result as effective as computing a single signal for each turbine depending on its operating conditions. In particular, when the main objective is to maximize the energy capture, control strategies aimed at reducing the wake effects have shown promising results [6].

Control strategies considering non-aggregate models have been an active research topic in the last years (see [7] and reference therein). One of their main difficulties is the consideration of complex dynamics from all the physical phenomena involved in the turbine interactions and the wake effects. High-fidelity models can only be used to validate strategies, and simplification must be used to obtain control-oriented models in order to use model-based control techniques, see for instance [8,9]. For this reason, some reported approaches propose the use of data-driven control techniques. Moreover, since reliability is also an important feature to take into account, control strategies should also consider the reduction of the communication channel usage. Among others, the use of safe experimentation dynamics to design a distributed control strategy aimed at maximizing the energy capture considering local information and the total power amount produced by the wind farm is reported in [2]. In this proposed approach, control variables are randomly perturbed in order to optimize such total power. The resultant value of these variables should be updated only in case the amount of the total generated power was improved. Another viewpoint is reported in [10], where the control strategy is fed with local information only and its convergence is improved by adding information about the objective function gradient, no matter whether the resultant optimum was local. In [11], a Bayesian ascent algorithm is proposed including two different phases, i.e., a learning and an optimization phase. More recently, Ciri et al. [12] have applied extremum-seeking control to improve the power production by looking for the optimal gain of the torque controller at each individual wind turbine.

Thus, the implementation of a population-dynamics approach with time-varying set of strategies in a gradient-estimation algorithm, altogether as a data-driven decentralized control scheme, is the main contribution of this paper. The scheme comprises of a single controller per each wind turbine. The proposed approach is inspired in a population game [13], where the strategies are aligned to the domain-space coordinates of an unknown function to be maximized. Hence, the gradient estimation is performed from a unique (or multiple) measurement(s) such as in [14,15], corresponding to the population game strategies. Finally, the set of strategies are updated in function of the gradients estimates and the entire routine is run again. Furthermore, the proposed algorithm allows also to establish set-points for the desired power generation. Moreover, different from other works related to evolutionary dynamics, e.g., [16–19], where the evolutionary games work as the optimization algorithm, this paper proposes to take advantage of this game-theoretical approach to assist a heuristic algorithm.

One of the main features of the proposed approach is the use of local information only, i.e., each turbine only requires information about its own control signal apart of the total power amount from the whole wind farm (following the approach reported in [2]). However, different from [2], the proposed control algorithm avoids the random testing towards obtaining improvements in the total generated power. Instead, gradient estimation is used [10,20], but highlighting significant differences:

- the algorithm proposed in this paper uses historical information of the system evolution to compute multiple directions of the gradient estimations in a decentralized fashion and for every single wind turbine, i.e., it is not necessary to share information among turbines, and
- the proposed approach produces global solutions due to the availability of the total generated power amount.

Moreover, different from [2,20,21], the individual saturation of the power generation in the turbines is considered, and it is shown that the proposed data-driven algorithm can handle with this situation maximizing the total power. Preliminary results related to this work were reported in [21]. Similarly as in [11,12], the key aspect in the algorithm is given by the appropriate computation of an ascent direction.

This paper is structured as follows. In Section 2, the problem of controlling wind farms and the wake modeling are briefly presented. Preliminary concepts used in the proposed algorithm design are stated in Section 3, while Section 4 describes in-depth the proposed algorithm. Section 5 presents and explains the data-driven decentralized control scheme for wind farms. Next, in Section 6, the main

results obtained are presented and discussed, which have been obtained with a case study of 80 wind turbines maximizing the total power generation by using the axial induction factors. Finally, some conclusions are drawn in Section 7.

Notation

Calligraphy letters are used to denote the sets, e.g., \mathcal{S} . The column vectors are denoted with bold font, e.g., \mathbf{y} . Every sub-index refers to elements corresponding to strategies in a population, e.g., $s_{i,k}$ refers to a vector associated to the i th strategy in a time instant k . Besides, $|\cdot|$ denotes the cardinality of a set. Hence, $\mathbb{R}_{\geq 0}$ denotes the set of all the non-negative real numbers and $\mathbb{R}_{> 0}$ denotes the set of all the strictly positive real numbers. Finally, both continuous-time and discrete-time frameworks are used throughout this paper denoted by k and t , respectively. Moreover, discrete time is denoted as a sub-index, e.g., \mathcal{S}_k , whereas continuous time is denoted as an argument, i.e., $p_i(t)$.

2. Problem Statement

The power generated by a turbine $i \in \mathcal{W}$ is given by

$$P_i(a_i, V_i) = \frac{1}{2} \rho A C_P(a_i) V_i^3, \quad (1)$$

being ρ the air density, A the area swept by the rotor, V_i the wind speed experienced by the turbine, and $\mathcal{W} = \{1, \dots, m\}$ a set indexing the turbine into the farm. The power coefficient C_P depends on the axial induction factor a_i according to

$$C_P(a_i) = 4 a_i (1 - a_i)^2. \quad (2)$$

In wind farm control applications, the axial induction factor a_i is used as control signal to regulate the power generated by each turbine. The maximum individual power production is achieved when $a_i = 1/3$.

In case of wind turbines within a farm, the wind speed faced by each turbine is a combination of the free-stream speed and the wakes produced by nearby turbines. Both the wind farm layout and the predominant wind speed direction determine the interactions caused by the wake effect. An illustration of the wake produced down-stream in a simple layout of m turbines aligned with the free-stream wind speed V_∞ can be seen in Figure 1.

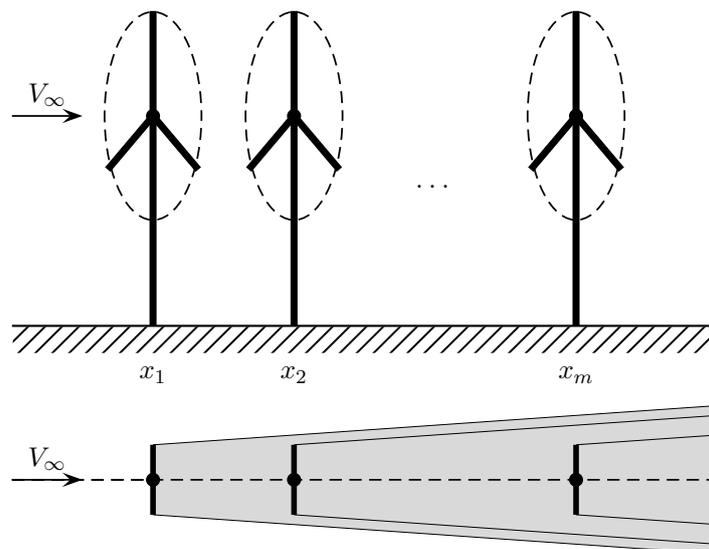


Figure 1. Example of wind farm layout and the corresponding wake effect.

The speed deficit caused by the wake effect can be modeled as proposed in [5]. When all turbines are aligned with V_∞ , the wind speed faced by the turbine $i \in \mathcal{W}$ is given by

$$V_i = V_\infty \left(1 - 2 \sqrt{\sum_{j \in \mathcal{W}: x_j < x_i} (a_j c_{ji})^2} \right), \quad (3)$$

where

$$c_{ji} = (D_j / (D_j + 2\beta(x_i - x_j)))^2,$$

with D_j the diameter of the wind rotor, x_j the position the turbine and β a coefficient indicating the expansion of the wake.

The control objective considered in this paper is to maximize the total generated power [2,20], i.e.,

$$\max_{\mathbf{a}} P_T(\mathbf{a}) = \sum_{i \in \mathcal{W}} P_i(\mathbf{a}). \quad (4)$$

This objective consists in finding the vector of axial induction factors $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_m]^\top \in \mathbb{R}^m$ that maximizes the total power P_T produced by the entire wind farm. In general, a greedy strategy, in which every turbine works at $a_i = 1/3$, does not achieve the optimal value due to the wind speed deficit caused by the wake effects. In order to maximize the total energy capture, the most upstream turbines must reduce the generation. Thus, the wind speed deficit faced by downstream turbines is lower and the total power is higher. As the quantification of the wake effect is quite complex and depending on uncertain parameters, it is advisable the use of control algorithms not relying on wake modeling. This is the purpose of the algorithm proposed in this paper.

3. Preliminary Concepts

The proposed strategy uses a data set, which is composed by two main elements: (i) the estimation of the gradient; and (ii) the computation of the relevance of each datum. Figure 2 presents the main steps of the proposed heuristic algorithm, which are explained in detail next along Sections 3 and 4.

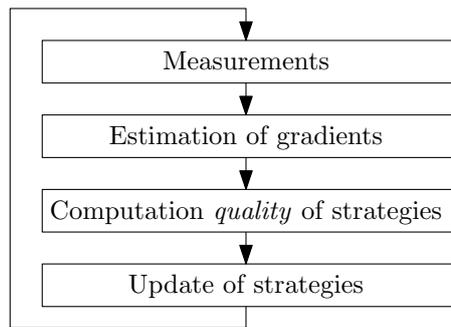


Figure 2. General steps of the heuristic proposed approach.

3.1. Gradient Estimation

Consider an unknown function $f(\mathbf{y})$, where $\mathbf{y} \in \mathbb{R}^m$, i.e., $f: \mathbb{R}^m \rightarrow \mathbb{R}$. The objective is to find the appropriate vector \mathbf{y} such that f is maximized. For that, a data-driven algorithm based on gradient estimation (which uses a distributed gradient estimation proposed in [14,15]), and evolutionary game-theoretical concepts is proposed. In particular, several gradients are estimated and a qualification regarding each one of them is performed in order to determine a direction for the vector \mathbf{y} .

Let $\mathbf{c}, \mathbf{d} \in \mathbb{R}^m$ be two arbitrary vectors in the domain of the function f . It is assumed that the function values $f(\mathbf{c})$ and $f(\mathbf{d})$ are available through measurements, and that information is known in the coordinates \mathbf{c} and \mathbf{d} , respectively. Therefore, the estimation of an increasing rate and direction over the function f between the vectors \mathbf{c} and \mathbf{d} is given by

$$g(\mathbf{c}, \mathbf{d}) = (f(\mathbf{d}) - f(\mathbf{c})) \frac{\mathbf{d} - \mathbf{c}}{|f(\mathbf{d}) - f(\mathbf{c})|}. \quad (5)$$

$$g_i(f(\mathbf{c}), f(\mathbf{d}), c_i, d_i) = (f(\mathbf{d}) - f(\mathbf{c})) \frac{d_i - c_i}{|f(\mathbf{d}) - f(\mathbf{c})|} \quad (6)$$

be the i th component of $g(\mathbf{c}, \mathbf{d})$, i.e., $g = [g_1 \ \dots \ g_m]^\top$.

The function $g : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ is a modification of the original one presented in [14]. Basically, there is a need in this problem to capture the information regarding the increasing directions, and that is why the magnitude has been slightly modified.

Remark 1. *In order to estimate the gradients for the unknown function $f(\mathbf{y})$, it is necessary*

1. *to be able to capture measurements of the unknown function f , and*
2. *to know the correspondence of the measurement with the element in the domain of f , i.e., for a measurement $f(\mathbf{d})$ the element \mathbf{d} in the domain of f is known.*

3.2. Population-Game Role

The proposed algorithm uses multiple estimated gradients based on measurements. The decision variables are updated by computing a superposition of the gradients. Thus, an appropriate increasing direction for the unknown function is identified in order to maximize it. One key step within the algorithm is to classify the multiple measurements according to their quality (those elements in the domain with higher-value measurements are considered to be of better quality, see Section 4). Such classification is performed by using a population-games-based algorithm whose setting changes every iteration. Next, the evolutionary dynamics and population games background is introduced.

Consider a population with a finite and large number of rational agents. Within the population, there are n available strategies every discrete time $k \in \mathbb{Z}_{\geq 0}$ (with a sampling time given by $\tau \geq t^f$) associated to a coordinate. At time instant k and during a fixed period of time denoted by τ , agents have the possibility to choose among n available strategies. Let $\mathcal{I} = \{1, \dots, n\}$ be the set of indexes corresponding to the n available strategies, and $\mathcal{S}_k = \{s_k^1, \dots, s_k^n\}$ the set of strategies, where $s_k^i \in \mathbb{R}^m$, for all $i \in \mathcal{I}$, and $\mathcal{S}_k \subset \mathbb{R}^m$. It should be noticed that the set of strategies \mathcal{S}_k varies along the discrete time with a sampling time given by τ .

At each discrete-time step k , a strategic interaction in continuous time occurs for a period of time equivalent to τ (sampling time). In the previously mentioned interaction and at time instant k , the scalar value $p_i(t) \in \mathbb{R}_{\geq 0}$, with $0 \leq t \leq \tau$, corresponds to the proportion of agents that are selecting the strategy $s_k^i \in \mathcal{S}_k$.

All the proportions of agents selecting the different strategies in the population compose a strategic distribution or a population state, which is denoted by $\mathbf{p}(t) \in \mathbb{R}^n$, $0 \leq t \leq \tau$. The set of all possible population states evolves in a simplex denoted by $\Delta = \{\mathbf{p}(t) \in \mathbb{R}_{\geq 0}^n : \sum_{i \in \mathcal{I}} p_i(t) = 1\}$. Agents have incentives to select among available strategies in the population according to a fitness function given by $f_i(p_i(t)) = (f(s_k^i) - \pi)p_i(t)$, $0 \leq t \leq \tau$, for all $i \in \mathcal{I}$, where $\pi \in \mathbb{R}_{> 0}$ is an upper bound such that $\pi > f(s_k^i)$, for all $i \in \mathcal{I}$. The fitness function for the whole population is given by $f(\mathbf{p}) = [f_1(p_1) \ \dots \ f_n(p_n)]^\top$. Notice that the constant value π affects all the fitness functions to ensure that $f_i(p_i(t))$ is decreasing with respect to $p_i(t)$. Then, it is ensured that the population game is stable according to Definition 1 (adapted from [13]).

Definition 1. *A population game $f : \Delta \rightarrow \mathbb{R}^n$ is a stable game if $\mathbf{z}^\top Df \mathbf{z} \leq 0$, for all $\mathbf{z} \in \Delta_T$, $\mathbf{p} \in \Delta$, where Δ_T is the tangent space of the simplex given by $\Delta_T = \{\mathbf{z} \in \mathbb{R}^n : \sum_{i \in \mathcal{I}} z_i = 0\}$.*

The objective within the population is to achieve a Nash equilibrium that is denoted by $\mathbf{p}^* \in \Delta$. Formally, the set of Nash equilibrium is defined next.

Definition 2. Let $\mathbf{p}^* \in \Delta$ in the population be a Nash equilibrium if each used strategy entails the maximum benefit for the proportion of agents selecting it, i.e., the set $\{\mathbf{p}^* \in \Delta : p_i^* > 0 \Rightarrow f_i(\mathbf{p}^*) \geq f_j(\mathbf{p}^*)\}$, for all $i, j \in \mathcal{I}$, corresponds to the Nash equilibria.

Additionally, suppose that the possible interaction among agents choosing different strategies is given by an undirected non-complete communication graph denoted by $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where \mathcal{I} is the set of vertices representing the strategies and $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{I}\}$ is the set of edges or links determining possible communication and information sharing among strategies. The set of neighbors of the node $i \in \mathcal{I}$ is given by $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$. It should be clear that each strategy estimates a gradient based only on information from its neighbors.

For the proposed algorithm, the proportion of agents $p_i(\tau)$ represents a quality assigned to the strategy s_k^i evaluated at time τ , which is a tuning parameter of the proposed algorithm. In other words, the proportion provides information about how well the strategy s_k^i maximizes the function f with respect to the other available strategies $\mathcal{S}_k \setminus \{s_k^i\}$ at time instant $k \in \mathbb{Z}_{\geq 0}$. Afterwards, the set of strategies \mathcal{S}_k has an update based on gradient estimations over the function f , and the different qualities for all the strategies are given by $\mathbf{p} \in \mathbb{R}^n$.

The Nash equilibrium for the population game f is obtained by solving the maximization of the potential function using the distributed population dynamics presented in [18]. In this case, the distributed projection dynamics (DPD) are chosen, and are defined as

$$\dot{\mathbf{p}} = \mathbf{L}f(\mathbf{p}).$$

Here, \mathbf{L} is the Laplacian matrix corresponding to the connected graph \mathcal{G} . Clearly, another distributed population dynamics could have been used (distributed replicator dynamics, distributed Smith dynamics). If the sampling time τ is big enough, then $\mathbf{p}(\tau) = \mathbf{p}^* \in \Delta$ is the Nash equilibrium of the game. Otherwise, the quality of strategies at time $k \in \mathbb{Z}_{\geq 0}$ corresponds to a transitory trend of the proportion of agents in the population.

Next, in Section 4 the combination of these two concepts is shown in order to derive a data-driven optimization algorithm.

4. Algorithms According to Information Availability

The proposed approach can be implemented in different manners depending on the availability of information. These approaches are selected depending on the information scheme of the application.

4.1. Using Multiple Measurements at Each Iteration

Consider the smallest population games involving only two strategies $\mathcal{S}_k = \{s_k^1, s_k^2\}$, and with an associated communication graph \mathcal{G} as introduced in Section 3. The initial condition for the strategic set \mathcal{S}_0 consists of two arbitrary strategies denoted by $s_0^1 \in \mathbb{R}^m$ and $s_0^2 \in \mathbb{R}^m$. Hence, the initial population state is arbitrarily selected in the relative interior of the simplex set, i.e., $\mathbf{p}(0) \in \text{int}\Delta$. Furthermore, the fitness functions corresponding to the strategies are given by $f_1(p_1(t)) = (f(s_k^1) - \pi)p_1(t)$ and $f_2(p_2(t)) = (f(s_k^2) - \pi)p_2(t)$, being $\pi \in \mathbb{R}_{>0}$ an upper bound for the fitness functions such that $\pi > f(s_k^i)$, for all $i \in \mathcal{I}$.

Remark 2. The value $f(s_k^i)$ is measurable in the corresponding node $i \in \mathcal{I}$, for all $k \in \mathbb{Z}_{\geq 0}$.

Due to the fact that both strategies $s_k^1 \in \mathcal{S}_k$ and $s_k^2 \in \mathcal{S}_k$ have information about each other, then the gradients $g(s_k^1, s_k^2)$ and $g(s_k^2, s_k^1)$ can be computed in the nodes $1 \in \mathcal{I}$, and $2 \in \mathcal{I}$, respectively. Therefore, agents have the required information to make a decision within the population.

The update of the strategic set \mathcal{S}_k is performed as follows. For instance, assume that $f(s_k^2) > f(s_k^1)$, then the strategy s_k^1 evolves towards the strategy s_k^2 in order to be closer in the domain. In contrast, s_k^2 is updated in the opposite direction of the strategy s_k^1 , getting farther from it in the domain.

Thus, according to the strategic update procedure, those strategies with better quality (i.e., whose image in the function $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is better) have a smoother change along the time, whereas those with lower values suffer bigger changes. Then, the update rate for the strategy s_k^i , for all $i \in \mathcal{I}$ is denoted by

$$\theta_{i,k} = (1 - p_i(\tau))\gamma, \quad (7)$$

where $\gamma \in \mathbb{R}_{>0}$ is a common tunable parameter for all the strategies, which determines the update rate according to (7), for all $i \in \mathcal{I}$. For instance, consider the case involving two strategies $i, j \in \mathcal{I}$, and without loss of generality, let $f(s_k^i) > f(s_k^j)$. Therefore, it is expected that $p_i(\tau) > p_j(\tau)$, i.e., the strategy s_k^i has a better quality than the strategy s_k^j . Consequently, a bigger update rate is assigned to the strategy s_k^j , i.e., it is obtained that $\theta_i(\tau) < \theta_j(\tau)$. In addition, it is proposed to take into account an exploration term that allows to identify potentially better quality strategies in the domain of the function f . The aforementioned exploration parameter is applied within the strategic update by means of a random value $\delta \in [-\varepsilon, \varepsilon]^m$, where $\varepsilon \in \mathbb{R}_{>0}$. Hence, the strategic update is performed as follows:

$$s_{k+1}^i = s_k^i + |\mathcal{N}_i|^{-1} \theta_{i,k} \sum_{j \in \mathcal{N}_i} g(s_k^i, s_k^j) + \delta, \quad \forall i \in \mathcal{I}.$$

In order to illustrate how the strategic update is performed according to a superposition of gradients, Figure 4 shows an example for a function f in the \mathbb{R}^2 domain. Figure 4a shows a population case involving four strategies whose information interaction is determined by a star graph communication, i.e., $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I} = \{1, \dots, 4\}$, and $\mathcal{E} = \{(1,2), (1,3), (1,4)\}$. For the example presented in Figure 4a, $f(s_k^1) > f(s_k^3)$, $f(s_k^2) > f(s_k^1)$ and $f(s_k^4) > f(s_k^1)$, where $s_k^1, s_k^2, s_k^3, s_k^4 \in \mathbb{R}^2$, and $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. The superposition of the three gradients can be seen at the node 1 $\in \mathcal{I}$.

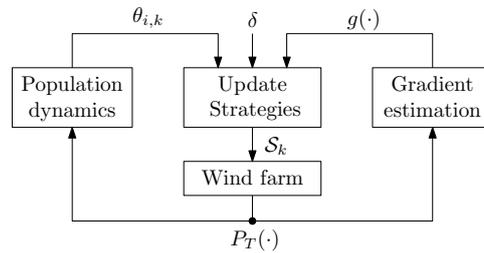


Figure 3. General scheme for the gradient-estimation-based algorithm with population-games assistance.

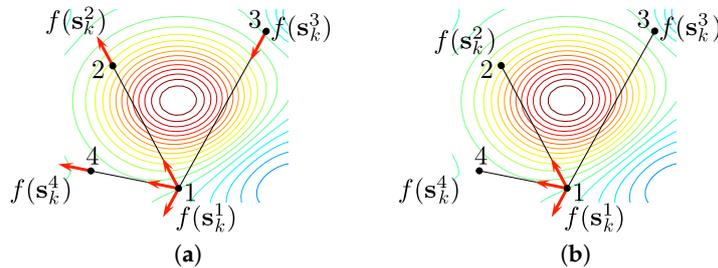


Figure 4. Example gradient estimation with four strategies $\mathcal{S}(k) = \{s_k^1, s_k^2, s_k^3, s_k^4\}$, i.e., $n = 4$, and $f : \mathbb{R}^2 \mapsto \mathbb{R}$, i.e., $m = 2$. Vectors illustrate the direction for the strategies update and the superposition of influences over strategy with index 1. (a) Various available measurements every iteration, (b) one available measurement every iteration.

Notice that the proposed approach requires the availability of n measurements corresponding to n different images of the function f , which should be captured at the same time instant $k \in \mathbb{Z}_{\geq 0}$, i.e., it is necessary to capture the measurements corresponding to the values $f(s_k^i)$, for all $i \in \mathcal{I}$. Figure 3 presents a general scheme for the algorithm. Nevertheless, a subtle modification can be applied to the algorithm such that less information is required at each time instant as discussed in the next section.

4.2. Using a Single Measurement at Each Iteration

Consider the same population game that has been introduced in Section 3, i.e., there is a set of n strategies at each time instant $k \in \mathbb{Z}_{\geq 0}$ denoted by $\mathcal{S}_k = \{s_k^1, \dots, s_k^n\}$. However, different from Section 3, there are $n - 1$ preserved strategies within the set \mathcal{S}_k in the strategy update ($n - 1$ strategies do not change at each time instant), and only one of them is modified. Formally, it is said that $\{\mathcal{S}_{k+1}\} \setminus \{\mathcal{S}_k\} = \{s_{k+1}^1\}$, which means that there is only one new strategy comparing the strategic sets \mathcal{S}_{k+1} and \mathcal{S}_k .

It is quite important to highlight that even though there is only one new datum at each iteration, the estimation of all the gradients is different.

To initialize the algorithm, it is assumed that the measurements of the values corresponding to the function f for the n initial strategies are known, i.e., $f(s_0^1), \dots, f(s_0^n)$ are initially known. Notice that, the condition to initialize the algorithm can be ensured within at least a time $n\tau$, i.e., it is possible to capture the n required initial measurements in n iterations with sampling time τ .

Although there is only one new measurement at each time instant, all strategies can be updated. Therefore, the information limitation is treated by means of the following algorithm:

$$\begin{aligned} s_{k+1}^1 &= s_k^1 + |\mathcal{N}_1|^{-1} \theta_{1,k} \sum_{j \in \mathcal{I}} g(s_k^1, s_k^j) + \delta, \\ s_{k+1}^i &= s_k^{i-1}, \quad i \in \mathcal{I} \setminus \{1\}. \end{aligned} \quad (8)$$

To illustrate the procedure in Equation (8), Figure 4b presents an example with only one available measurement at each time instant, i.e., the new measurable information is given by $f(s_k^1)$. Note that, at the next time instant $k + 1$, the values for the function $f(s_{k+1}^2) = f(s_k^1)$, $f(s_{k+1}^3) = f(s_k^2)$, and $f(s_{k+1}^4) = f(s_k^3)$ are already known by the algorithm.

5. Data-Driven Decentralized Control of Wind Farms

The power generated by the wind turbine $i \in \mathcal{W}$, denoted by $P_i(\mathbf{a})$ with $\mathbf{a} = [a_1 \ \dots \ a_m]^\top \in \mathbb{R}^m$, depends on the behavior of other turbines as shown in Equation (3). Therefore, it is obtained that the unknown function f in Equation (5) corresponds to the total power function $P_T = \sum_{i \in \mathcal{W}} P_i$ in Equation (4). On the other hand, each strategy belonging to the domain of the unknown function is given by $s_k^i = \mathbf{a}_k^i \in \mathbb{R}^n$. More specifically, $\mathbf{a}_k^i = [a_1^i \ \dots \ a_n^i]^\top$. Therefore, for this specific control problem, there is only one available measurement at each time instant $k \in \mathbb{Z}_{\geq 0}$, i.e., the only available information is the total generated power for the current established axial induction factors. In consequence, the appropriate approach to deal with this data-drive control problem is the one introduced in Section 4.2. The control scheme corresponds to a decentralized architecture since individual decisions are made at each wind turbine without requiring communication with other turbines.

Figure 5 shows the traditional decentralized control scheme, where the information about the total generation power $P_T(\mathbf{a})$ is provided to each wind turbine $i \in \mathcal{W}$. In addition, each turbine also has information about its current axial coefficient. Then, the algorithm in Equation (8) can be written for each turbine as follows:

$$\begin{aligned} a_{j,k+1}^1 &= a_{j,k}^1 + \varphi_k, \\ a_{j,k+1}^i &= a_{j,k}^{i-1}, \quad i \in \mathcal{I} \setminus \{1\}, \end{aligned} \quad (9)$$

where

$$\varphi_k = \frac{\theta_{1,k}}{|\mathcal{N}_1|} \sum_{\ell \in \mathcal{N}_1} g_i(P_T(\mathbf{a}_k^1), P_T(\mathbf{a}_k^\ell), a_{j,k}^\ell, a_{j,k}^1) + \delta_i,$$

for all $j \in \mathcal{W}$. Notice that in the algorithm in Equation (9), each turbine has the required information since $P_T(\mathbf{a}_k^1)$ is measured and $P_T(\mathbf{a}_k^\ell)$ has been stored for all $\ell \in \mathcal{N}_1$.

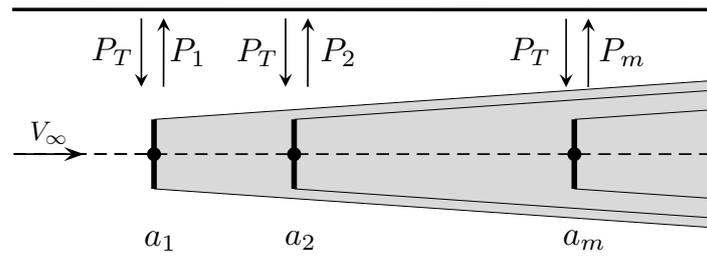


Figure 5. Typical decentralized control scheme. Each wind turbine has information about the total generated power and its own axial induction factor.

Now suppose that it is necessary to limit the individual power generation under a value P_i^s , i.e., the admissible power generation for each wind turbine is

$$P_i(a_i, V_i) = \min \left(P_i^s, \frac{1}{2} \rho A c_p(a_i) V_i^3 \right), \forall i \in \mathcal{W}.$$

Therefore, it is necessary to verify the admissibility of axial coefficient a_i if $P_i(a_i, V_i) > P_i^s$. This procedure is performed by computing Equations (1) and (2) in a decentralized manner to find the new admissible value for a_i , i.e., an axial coefficient a_i^s such that $P_i(a_i^s, V_i) = P_i^s$ is established.

6. Case Study and Simulation Results

The effectiveness of the proposed control algorithm was shown by using the Horns Rev wind farm, which has 80 turbines of 2 MW each, and with rotor diameter of 80 m. This farm is arranged in a 8×10 layout with a separation among turbines of five rotor diameters (400 m) (see Figure 8). The model parameters were chosen as $\beta = 0.075$ and $\rho = 1.225 \text{ kg/m}^3$.

The proposed decentralized algorithm considered three data at each iteration (one from a current measurement, and two from historical data), that means $n = 3$, being $\tau = 0.01 \text{ s}$, $\varepsilon = 0.001$ and $\gamma = 1$ the the tuning parameters selected. Besides, the communication graphs considered for each algorithm were $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I} = \{1, 2, 3\}$, and $\mathcal{E} = \{(1, 2), (1, 3)\}$. The decentralized scheme was composed of 80 different algorithms as in Equation (9), i.e., each turbine had its individual algorithm. Two scenarios were tested for four different wind speed directions, i.e., 0° , 15° , 30° , and 45° :

- Scenario 1: the free-stream wind speed was below the rated value and all wind turbines are working in maximization of the energy capture.
- Scenario 2: the free-stream wind speed was above the rated value and some turbines are working in power limitation (at 2 MW).

It is important to state that, when the wind speed faced by a turbine was above the rated value, the axial coefficient (for the corresponding turbine) was imposed by the turbine controller and not by the wind farm control. The control algorithm then computed the axial induction factors for the rest of wind turbines in order to generate maximum power for the wind speed conditions.

Figure 6a shows the total power the wind farm produces related to scenario 1 considering four different wind speed directions. Dashed lines indicates the total power using a greedy control strategy ($a_i = 1/3 \forall i$) for the four directions, respectively. The proposed algorithm was capable of increasing the generation with respect to the greedy case in the four directions, although it was more marked in 0° and 45° . This was a consequence of the wind farm layout, in which the air flow disturbance caused by up-stream turbines is more notorious. In Figure 6b,c, it can be seen that the power and the axial induction factors for wind turbines 1–10 (in case of wind speed direction 0°). These figures show that the algorithm reduces the axial coefficients in the up-stream wind speed and increases the one in last turbines. This produces an increase of the wind power in the last turbines and the total power generated by the entire wind farm.

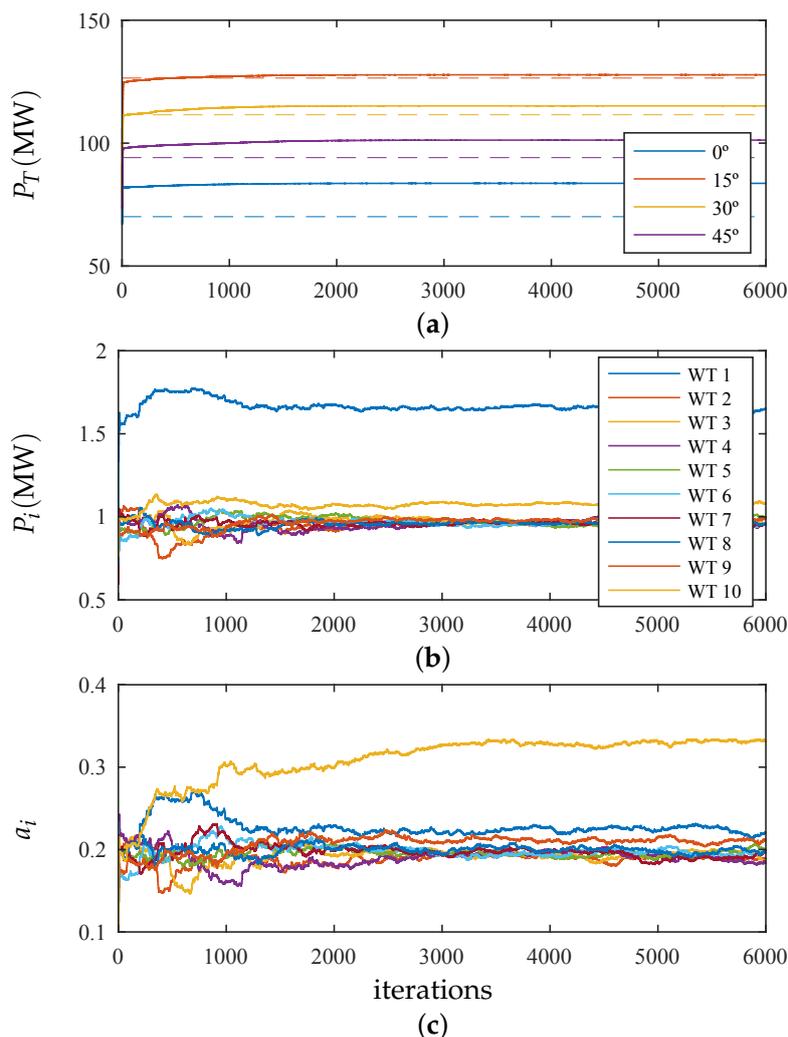


Figure 6. (a) Total powers for scenario 1 (free-stream wind speed of 10 m/s) for four wind speed directions. (b) Power generated by wind turbines 1–10 and with wind direction of 45° . (c) Axial coefficients for wind turbines 1–10 and with wind direction of 45° .

Results corresponding to scenario 2 are shown in Figure 7. In this scenario, the free-stream wind speed was above rated value and therefore the total power production should be close to rated values (160 MW). It can be observed in Figure 7 that this is the case for the wind speed directions of 15° , 30° and 45° , but not for 0° . In this last case, the wake effects induce the reduction the wind speed faced by the set of last wind turbines, causing a total power lower than 160 MW. Nevertheless, observing the powers and axial induction factors for wind turbines 1–10, it can be seen that the proposed algorithm in this situation still seeks to maximize the power generation by reducing the generation in the first turbines and increasing it in the last ones.

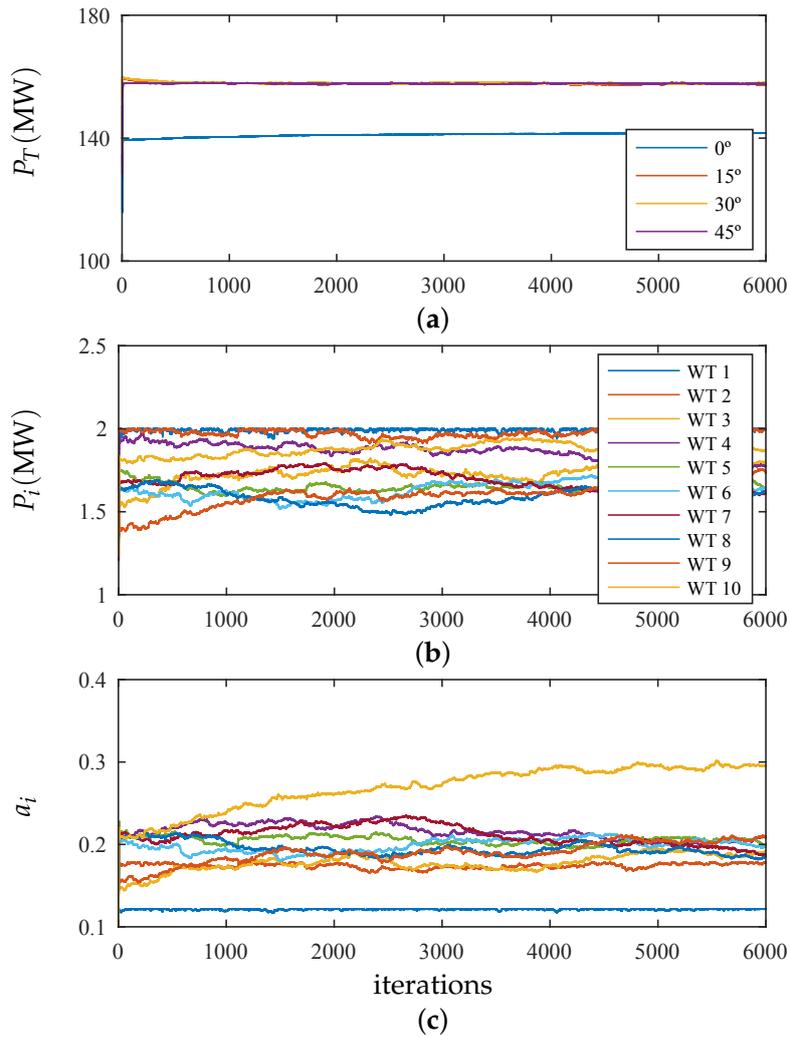


Figure 7. (a) Total powers for scenario 2 (free-stream wind speed of 12 m/s) for four wind speed directions. (b) Power generated by wind turbines 1–10 and with wind direction of 45° . (c) Axial coefficients for wind turbines 1–10 and with wind direction of 45° .

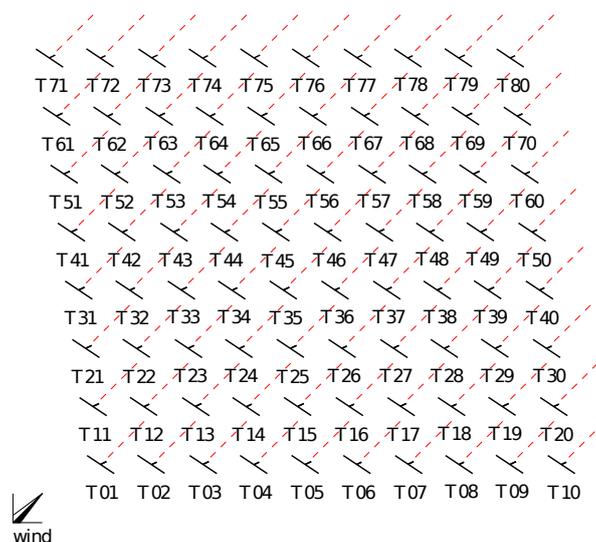


Figure 8. Horns Rev wind farm of 80 turbines facing a main wind speed with 45° direction.

7. Conclusions

A data-driven control for wind farms under a decentralized scheme has been presented. In the proposed algorithm, multiple gradient estimations are utilized in order to improve and speed up the maximization of the total generated power. The control is organized in a decentralized scheme, in which m algorithms produce one control variable for each wind turbine. In spite of the limited communications among agents, the proposed control is able to converge to the global solution. The control strategy was evaluated by simulation in the the well known Horns Rev wind farm layout under low and high wind speed conditions (in the latter the control actions saturate). In both cases, the algorithm was capable of increasing the total power compared with the greedy case and for different wind directions (see Figures 6 and 7).

Even though it has been presented for this specific application, this algorithm may be useful in other engineering problems in which data-driven approaches (i.e., only measured information is available) and maximization of a given objective are required. Moreover, it has been shown the versatility of the algorithm to consider any parameter as the decision variable within the optimization problem. Besides, dealing with the variance minimization of the power generation remains an appealing open problem. Finally, since the proposed algorithm uses estimated gradients based on measurements for known parameter values, i.e., in the maximization of an unknown function f , it is known that the measurement $f(a)$ corresponds to the variable a . In this regard, other decision variables could be easily considered by following the same procedure.

Author Contributions: All the authors have equally contributed in the research presented in this manuscript as well as in its preparation.

Funding: This work has been partially funded by the Spanish projects SCAV (Ref. DPI2017-88403-R) and DEOCS (Ref. DPI2016-76493-C3-3-R) and the Colombian project ISAGEN Solución Energética Piloto La Guajira. J. Barreiro-Gomez gratefully acknowledges support from U.S. Air Force Office of Scientific Research under grant number FA9550-17-1-0259.

References

1. Marden, J.; Ruben, S.; Pao, L.Y. Surveying Game Theoretic Approaches for Wind Farm Optimization. In Proceedings of the 50th AIAA Aerospace Sciences Meeting, Nashville, TN, USA, 9–12 January 2012; pp. 1–10.
2. Marden, J.R.; Ruben, S.D.; Pao, L.Y. A Model-Free Approach to Wind Farm Control Using Game Theoretic Methods. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1207–1214, doi:10.1109/TCST.2013.2257780.
3. Buccafusca, L.; Beck, C.; Dullerud, G. Modeling and Maximizing Power in Wind Turbine Arrays. In Proceedings of the IEEE Conference on Control Technology and Applications (CCTA), Kamuela, HI, USA, 27–30 August 2017; pp. 773–778.
4. Zhong, S.; Wang, X. Decentralized Model-Free Wind Farm Control via Discrete Adaptive Filtering Methods. *IEEE Trans. Smart Grid* **2018**, *9*, 2529–2540.
5. Jensen, N.O. *A Note on Wind Generator Interaction*; Technical Report; Riso National Laboratory: Roskilde, Denmark, 1983.
6. Fleming, P.; Lee, S.; Churchfield, M.; Scholbrock, A.; Michalakes, J.; Johnson, K.; Moriarty, P. The SOWFA Super-Controller: A High-Fidelity Tool for Evaluating Wind Plant Control Approaches. In Proceedings of the EWEA Annual Meeting, Vienna, Austria, 4–7 February 2013.
7. Boersma, S.; Doekemeijer, B.; Gebraad, P.; Fleming, P.; Annoni, J.; Scholbrock, A.; Frederik, J.; van Wingerden, J.W. A tutorial on control-oriented modeling and control of wind farms. In Proceedings of the American Control Conference, Seattle, WA, USA, 24–26 May 2017, doi:10.23919/acc.2017.7962923.
8. Annoni, J.; Gebraad, P.M.O.; Scholbrock, A.K.; Fleming, P.A.; van Wingerden, J. Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model. *Wind Energy* **2016**, *19*, 1135–1150, doi:10.1002/we.1891.

9. Vali, M.; van Wingerden, J.; Boersma, S.; Petrović, V.; Kühn, M. A predictive control framework for optimal energy extraction of wind farms. *Journal of Physics.: Conference Series*, **2016**, *753*, 5, doi:10.1088/1742-6596/753/5/052013.
10. Gebraad, P.M.O.; van Wingerden, J.W. Maximum power-point tracking control for wind farms. *Wind Energy* **2015**, *18*, 429–447.
11. Park, J.; Law, K.H. Bayesian Ascent: A Data-Driven Optimization Scheme for Real-Time Control With Application to Wind Farm Power Maximization. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 1655–1668, doi:10.1109/tcst.2015.2508007.
12. Ciri, U.; Rotea, M.; Santoni, C.; Leonardi, S. Large-eddy simulations with extremum-seeking control for individual wind turbine power optimization. *Wind Energy* **2017**, *20*, 1617–1634, doi:10.1002/we.2112.
13. Sandholm, W.H. *Population Games and Evolutionary Dynamics*; MIT Press: Cambridge, MA, USA, 2010.
14. Simić, S.N.; Sastry, S. Distributed gradient estimation using random sensor networks. In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Memorandum UCB/ERL M02/25, Atlanta, GA, USA, 28 September 2002.
15. Simić, S.N.; Sastry, S. Distributed Environmental Monitoring Using Random Sensor Networks. In *Information Processing in Sensor Networks: Second International Workshop*; Zhao, F.; Guibas, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 582–592.
16. Tembine, H.; Altman, E.; El-Azouzi, R.; Hayel, Y. Evolutionary games in wireless networks. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2010**, *40*, 634–646.
17. Quijano, N.; Ocampo-Martinez, C.; Barreiro-Gomez, J.; Obando, G.; Pantoja, A.; Mojica-Nava, E. The role of population games and evolutionary dynamics in distributed control systems. *Control Syst. Mag.* **2017**, *37*, 70–97.
18. Barreiro-Gomez, J.; Obando, G.; Quijano, N. Distributed Population Dynamics: Optimization and Control Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 304–314.
19. Barreiro-Gomez, J.; Ocampo-Martinez, C.; Quijano, N. Time-varying Partitioning for Predictive Control Design: Density-Games Approach. *J. Process Control* **2019**, *75*, 1–14, doi:10.1016/j.jprocont.2018.12.011.
20. van Dam, F.C.; Gebraad, P.M.O.; van Wingerden, J. A Maximum Power Point Tracking Approach for Wind Farm Control. In Proceedings of the Science of Making Torque from Wind, Oldenburg, Germany, 9–11 October 2012.
21. Barreiro-Gomez, J.; Ocampo-Martinez, C.; Bianchi, F.; Quijano, N. Model-free control for wind farms using a gradient estimation-based algorithm. In Proceedings of the European Control Conference (ECC), Linz, Austria, 15–17 July 2015; pp. 1516–1521.