Model-free Control for Wind Farms using a Gradient Estimation-based Algorithm

J. Barreiro-Gomez, C. Ocampo-Martinez, F. Bianchi, and N. Quijano

Abstract— **Wind turbines working close to other turbines experience interactions that affect the power production. These interactions arise as a consequence of wakes caused by upstream wind turbines. In order to achieve a more effective and precise control of the power generated by wind farms, the control strategy must consider these interactions. However, the phenomena involved in wake effects are complex especially in cases of large number of turbines. This paper presents the implementation of a gradient estimation-based algorithm as a model-free control for two different control schemes aimed to maximize the energy capture of a wind farm. One control is centralized, leaving to a supervisor the task of command computation and the other topology is decentralized, distributing the performing generation among wind turbines. This latter scheme aims to increase the reliability of the wind farm operation by reducing the communications needed to fulfill the objective of maximizing energy capture. Both control schemes are evaluated by simulation in the case of three-turbine wind farm.**

I. Introduction

Nowadays, wind turbines rarely operate geographically isolated. Instead, they are commonly gathered in groups called wind farms, which inject into the electrical power grid values comparable to conventional power sources. As a consequence of the high penetration levels of wind power plants, it becomes crucial to control wind farms in order to avoid undesirable effects in power systems and also to provide ancillary services. Initially, these control strategies were based on aggregate models representing wind farms as a large equivalent wind turbine. However, this approach does not take into account the interactions among turbines caused by wakes. A turbine located in the path of wakes produced by close turbines reduces the wind speed and is also exposed to a more turbulent air flow. As a result, a common command signal for all the wind turbines in the farm may not result so effective than a control strategy using the particular operating conditions of each turbine.

Control strategies considering non-aggregated models have been investigated recently. The main difficulty in this line is the complexity of the phenomena involved in the turbine interactions and wakes. For this reason, most of the results seek to use model-free control techniques. Reliability is also an important point to consider so the strategies also aim at reducing the use of communication channels. Marden et al. [5] use the safe experimentation dynamics to propose a distributed control to maximize the energy capture by employing local information and the value of the total power produced by the wind farm. In this game-theoretical approach, the total power is optimized by applying random perturbations to the control variables. These control variables are updated if an improvement of the total generated power is obtained, and are not updated otherwise. An improvement to this strategy is presented in [1], where the proposed control approach only uses local information and achieves a faster convergence with algorithms exploiting estimation of the gradient of the objective function, although the optimum is not global. Other works like [3], [9] use distributed model predictive control aiming to reduce loads and improving energy capture. The convergence is faster but the control design requires a wind farm model.

The main contribution of this paper is the implementation of a gradient-estimation-based algorithm as a model-free control approach for two different control schemes. The topologies comprise a centralized topology with a unique algorithm, and a typical decentralized control scheme with a controller per each wind turbine. Besides, the proposed algorithm is inspired in a population game [7], whose set of strategies corresponds to different coordinates in the domain of an unknown function that is desired to maximize. Then, gradients are estimated based on a unique (or multiple) measurement(s) as in [8], corresponding to the strategies in the population game. Finally, an update of the set of strategies depending on the estimated gradients is made, and the routine repeats.

It is important to highlight that the proposed algorithm in this paper requires local information (i.e., each turbine requires information about its own control signal) and the value of the total power produced by the wind farm as in [5]. However, different from [5], the proposed algorithm does not make random experimentation to seek an improvement in the generated power. In contrast, the proposed algorithm uses gradient estimation as in [1]. Nevertheless, there are two differences between the algorithm in [1] and the proposed algorithm: *i)* the proposed approach uses multiple directions of gradient estimations for each wind turbine based on stored information, instead of only one gradient estimation per each wind turbine; and *ii)* the proposed algorithm converges to the global solution since it disposes of information about the

J. Barreiro-Gomez and C. Ocampo-Martinez are with the Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain. {jbarreiro,cocampo}@iri.upc.edu

F. Bianchi is with Catalonia Institute for Energy Research, IREC, Jardins de les Dones de Negre 1, 08930 Sant Adrià de Besòs, Barcelona, Spain fbianchi@irec.cat

J. Barreiro-Gomez and N. Quijano are with Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1^{A}No 18A-10, Bogotá, Colombia {j.barreiro135, nquijano}@uniandes.edu.co.

J. Barreiro-Gomez is supported by Colciencias and by Agència de Gestió d'Ajust Universitaris i de Recerca AGAUR.

total generated power.

The remainder of this paper is organized as follows. Section II presents the problem of controlling wind farms and wake modeling. Section III introduces the proposed algorithm based on gradient estimation and population dynamics. Section IV shows a centralized scheme, and the decentralized scheme. Afterwards, Section V presents results of a case study with three wind turbines in a path topology for the two discussed control schemes. Finally, concluding remarks are presented in Section VI.

Notation: Calligraphy letters are used to denote 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., $\mathbf{s}_i(k)$ refers to a vector associated to strategy i in a discrete time k. The cardinality of a set is denoted as $|\cdot|$, and the sign function is denoted by $\text{sgn}(\cdot)$. Finally, $\mathbb{R}_{\geq 0}$ denotes the set of all the non-negative real number and $\mathbb{R}_{>0}$ denotes the set of all positive real numbers.

II. Problem statement

Consider a wind farm composed by m wind turbines and let $\mathcal{W} \ = \{1,...,m\}$ be the set of turbines. The power generated by a turbine $i \in \mathcal{W}$ is given by

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

where ρ is the air density in kg/m^3, A is the area swept by the wind rotor in m^2, and V_i is the wind speed experienced by the turbine in m/s. In energy capture maximization, the power coefficient C_P can be considered as a function of the axial induction a_i according to

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

For the purpose of wind farm control, the variable a_i is considered as the control input of the wind turbine. The maximum power extraction is achieved when $a_i = 1/3$.

In a wind farm, the energy captured by each turbine depends not only on the freestream wind speed but also on the interactions among turbines caused by the wake effects. These interactions depend on the geographical distribution of the turbines in the farm and on the wind direction. Figure 1 illustrates a simple wind farm layout of m turbines and the wakes generated by other downstream turbines when the freestream wind speed V_∞ is perfectly aligned with all the turbines. The wake effects on each turbine are quite complex, but it can be obtained a proper estimation with rather simple models such as the proposed in [2]. In the case that V_∞ is aligned with all the turbines, the wind speed experience by the turbine i can be computed as

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

where $c_{ji} = (D_j/(D_j + 2\beta(x_i - x_j)))^2$, with D_j as the diameter of the wind rotor in m, x_j the position the turbine, and β a roughness coefficient that defines the slope of the wake when passing through a turbine.

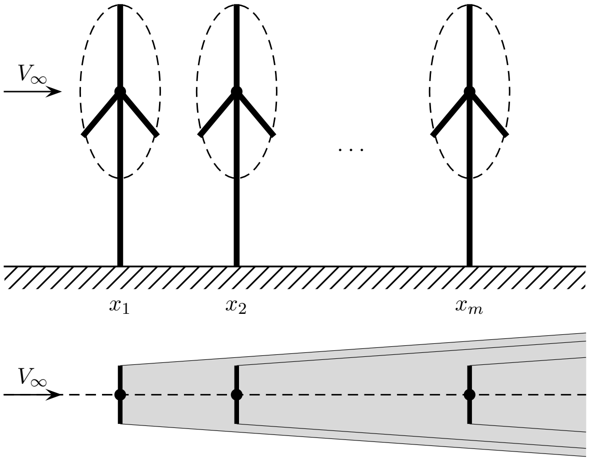

Fig. 1. Illustration of a simple wind farm layout and the corresponding wake effect.

The control objective is to maximize the total power captured by the wind farm

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

The idea is to find the vector of axial coefficients $\mathbf{a} = [a_1, a_2, \ldots, a_m]^\top$ in order to maximize the total power P_T produced by the wind farm. As a consequence of the wake effects, the maximum may not be reached at $a_i = 1/3$ for all $i \in \mathcal{W}$ since turbines located upstream reduce the wind speed experienced by the rotor of downstream turbines.

Due to the complexity of the expressions and the uncertainty on the model parameters, it is desirable that the considered control algorithms do not depend on both the wake and turbine models. This is the purpose of the algorithm proposed in the next section.

III. Algorithm

The problem that the algorithm solves is to maximize an unknown function $f(\mathbf{y})$, where $\mathbf{y} \in \mathbb{R}^m$, and $f : \mathbb{R}^m \mapsto \mathbb{R}$. This section presents the different components of the proposed algorithm. The algorithm is developed based on a population game that is solved at each iteration, and by using the distributed gradient estimation proposed in [8]. Consider a function with the mapping $f : \mathbb{R}^m \mapsto \mathbb{R}$. Then, consider two coordinates in the domain of f, i.e., $\mathbf{c}, \mathbf{d} \ \in \ \mathbb{R}^m$. It is assumed that the measurements $f(\mathbf{c})$ and $f(\mathbf{d})$ are available in the coordinates $\mathbf{c}$ and $\mathbf{d}$, respectively. The estimation of an increasing rate and direction over the function f between the points $\mathbf{c}$ and $\mathbf{d}$ is given by

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

Notice that the function G has a subtle modification from [8] regarding the magnitud of the gradient. This has been made since the requirement from the function G in the proposed algorithm is mainly the information about an increasing direction.

A. Population game approach

Consider a population with a finite and large number of agents, which are elements in the population able to make decisions. There are n available strategies in the population every discrete time $k \in \mathbb{Z}_{\geq 0}$ (with a sampling time given by $\Delta t \geq \tau$) associated to a coordinate, then it is defined a set of indices for the n strategies given by $\mathcal{I} = \{1, ..., n\}$. Each agent selects a strategy from the set of strategies denoted by $\mathcal{S}(k) = \{\mathbf{s}_1(k), ..., \mathbf{s}_n(k)\}$, where $\mathbf{s}_i(k) \in \mathbb{R}^m$, for all $i \in \mathcal{I}$, and $\mathcal{S}(k) \subset \mathbb{R}^m$. The scalar value $p_i(t) \in \mathbb{R}_{\geq 0}, 0 \leq t \leq \tau$, where t is the continuous time, is the proportion of agents that are selecting the strategy $\mathbf{s}_i(k) \in \mathcal{S}(k)$. All the proportion of agents selecting the different strategies form a strategic distribution or population state denoted by $\mathbf{p}(t) \in \mathbb{R}^n, 0 \leq t \leq \tau$.

Moreover, the set of all possible population states is given by a simplex denoted by $\Delta = \{\mathbf{p}(t) \in \mathbb{R}^n_{\geq 0} : \sum_{i\in\mathcal{I}} p_i(t) = 1, 0 \leq t \leq \tau\}$, and the set of population states in the interior of the simplex is denoted by $\text{int}\Delta = \{\mathbf{p}(t) \in \mathbb{R}^n_{>0} : \sum_{i\in\mathcal{I}} p_i(t) = 1, 0 \leq t \leq \tau\}$. Agents have incentives to switch among available coordinates in the population according to a fitness function given by $F_i(p_i(t)) = (f(\mathbf{s}_i(k)) - \pi)p_i(t), 0 \leq t \leq \tau$, for all $i \in \mathcal{I}$, where $\pi \in \mathbb{R}_{>0}$ is an upper bound such that $\pi > f(\mathbf{s}_i(k))$, for all $i \in \mathcal{I}$. This constant value affects all the fitness functions to ensure that F_i is decreasing with respect to $p_i(t)$. Then, it is ensured that the population game is stable as defined next [7].

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

Additionally, suppose that the possible interaction among agents choosing different coordinates is given by an undirected non-complete communication graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, where $\mathcal{V}_c = \mathcal{I}$ is the set of vertices or nodes representing the players, and $\mathcal{E}_c = \{(i, j) : i, j \in \mathcal{V}_c\}$ is the set of edges or links determining possible communication and information sharing among strategies. The set of neighbors of the node $i \in \mathcal{V}_c$ is given by $\mathcal{N}_{c_i} = \{j : (i, j) \in \mathcal{E}_c\}$.

In the proposed algorithm, the proportion of agents $p_i(\tau)$ represents a quality assigned to the strategy $\mathbf{s}_i(k)$ evaluated in a time τ, i.e., how good the strategy maximizes the function f with respect to the other available strategies $\mathcal{S}(k)$ in the discrete time $k \in \mathbb{Z}_{\geq 0}$. 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 $\mathbf{p} \in \mathbb{R}^n$.

Remark 1: It is proposed to solve the population game by using the distributed replicator dynamics [6]. Then, if the sampling time τ is big enough, then $\mathbf{p}^*(\tau) \in \Delta$ is the Nash equilibrium of the game. Otherwise, the quality of strategies at time $k \in \mathbb{Z}_{\geq 0}$ corresponds to a transitory tendency of the proportion of agents in the population. Furthermore, τ is one of the tuning parameters in the proposed algorithm. ♢

B. Using multiple measurements at each iteration

Suppose the simplest population of two strategies $\mathcal{S}(k) = \{\mathbf{s}_1(k), \mathbf{s}_2(k)\}$ forming a communication graph[1] $\mathcal{G}_c$. For the initial condition, the population game has two arbitrary strategies at discrete time $k = 0$ given by $\mathbf{s}_1(0) \in \mathbb{R}^m$ and $\mathbf{s}_2(0) \in \mathbb{R}^m$, respectively; and an arbitrary initial population state $\mathbf{p}(0) \in \text{int}\Delta$. The associated fitness functions are given by $F_1(p_1(t)) = (f(\mathbf{s}_1(k)) - \pi)p_1(t)$ and $F_2(p_2(t)) = (f(\mathbf{s}_2(k)) - \pi)p_2(t)$.

Remark 2: It is assumed that at each discrete time $k \in \mathbb{Z}_{\geq 0}$, the evaluation of the unknown function $f(\mathbf{s}_i(k))$ is measured in the corresponding node $i \in \mathcal{V}_c$ and for all $i \in \mathcal{I}$. ♢

Since strategies with indices 1 and 2 dispose of information each other, then the gradients $G(\mathbf{s}_1(k), \mathbf{s}_2(k))$ at node 1 and $G(\mathbf{s}_2(k), \mathbf{s}_1(k))$ at node 2 over f, can be determined and agents can make decisions within the population. The update of strategies in the population game is made as follows: if $f(\mathbf{s}_2(k)) > f(\mathbf{s}_1(k))$, then strategy $\mathbf{s}_1(k)$ changes to another strategy in direction of player $\mathbf{s}_2(k)$ to get closer to it, whereas strategy $\mathbf{s}_2(k)$ changes in opposite direction of $\mathbf{s}_1(k)$ getting farther, and avoiding a decreasing on the fitness function that is offered by agents in the next iteration.

In the updating strategy process, strategies with more quality have less change than those with less quality. More change in the update is assigned to those with less associated quality. The factor of change assigned to the updating of the strategy $\mathbf{s}_i(k), i \in \mathcal{I}$ is denoted by

$$\theta_i(k) = (1 - p_i(\tau))\gamma,$$

where γ is a common tuning factor for all strategies determining the rate of change of $\theta_i(k)$, for all $i \in \mathcal{I}$. For two strategies with indices $i, j \in \mathcal{I}$ suppose that $f(\mathbf{s}_i(k)) > f(\mathbf{s}_j(k))$, then it is expected that $p_i(\tau) > p_j(\tau)$. Consequently, more change factor is assigned to the updating of strategy $\mathbf{s}_j(k)$ than to the updating $\mathbf{s}_i(k)$, i.e., $\theta_i(\tau) < \theta_j(\tau)$. Furthermore, it is proposed to consider another parameter given by an exploration factor over the function f in the strategy updating process given by a random value $\delta \in [-\epsilon \quad \epsilon]^m$, $\epsilon \in \mathbb{R}$. Finally, the updating of strategies is given by

$$\mathbf{s}_i(k+1) = \mathbf{s}_i(k) + \frac{\theta_i(k)}{|\mathcal{N}_{ci}|} \sum_{j\in\mathcal{N}_{ci}} G(i, j) + \delta, \;\; \forall i \in \mathcal{I}.$$

Figure 2a) shows a case of four strategies forming a star communication graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, where $\mathcal{V}_c = \{1, \ldots, 4\}$, and $\mathcal{E}_c = \{(1, 2), (1, 3), (1, 4)\}$. For the example presented in Figure 2a), $f(\mathbf{s}_1(k)) > f(\mathbf{s}_3(k))$, $f(\mathbf{s}_2(k)) > f(\mathbf{s}_1(k))$ and $f(\mathbf{s}_4(k)) > f(\mathbf{s}_1(k))$, where $\mathbf{s}_1(k), \mathbf{s}_2(k), \mathbf{s}_3(k), \mathbf{s}_4(k) \in \mathbb{R}^2$, and $f : \mathbb{R}^2 \mapsto \mathbb{R}$. It can be seen the superposition of influences of strategies with indices $2, 3$, and 4 over the updating of strategy with index 1.

[1] In this algorithm, the communication graph is undirected, connected, and non-complete. The estimation of the gradients is made over this graph with partial information, i.e., each strategy estimates a gradient based only on information from its neighbors.

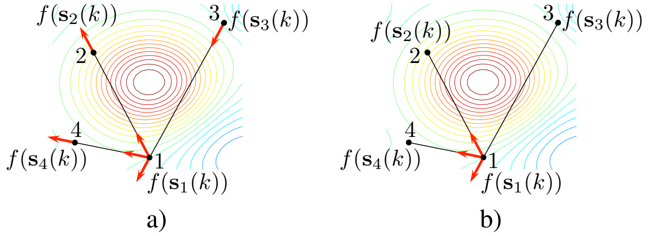

Fig. 2. Example of gradient estimation with four strategies $\mathcal{S}(k) = \{\mathbf{s}_1(k), \mathbf{s}_2(k), \mathbf{s}_3(k), \mathbf{s}_4(k)\}$ $(n = 4)$ and $f : \mathbb{R}^2 \mapsto \mathbb{R}$ $(m = 2)$. Vectors illustrate the direction for the strategies update and the superposition of influences over strategy with index 1. a) Various available measurements at each iteration, b) One available measurement at each iteration.

Notice that this approach is suitable in cases where there are n available measurements of the function at the same $k \in \mathbb{Z}_{\geq 0}$, i.e., at each iteration it is necessary to have the values for the function corresponding to the strategies in the population game. However, modifications over the algorithm can be made to solve maximization problems in cases with less available information at each iteration as shown in the next section.

C. Using one measurement at each iteration

In this case, consider the population game introduced in Section III-A. There is a set of n strategies at $k \in \mathbb{Z}_{\geq 0}$ given by $\mathcal{S}(k) = \{\mathbf{s}_1(k), ..., \mathbf{s}_n(k)\}$. It is important to clarify that in this approach the undirected communication graph $\mathcal{G}_c$ must be a complete graph[2]. Moreover, an algorithmic connected undirected non-complete graph is introduced, denoted by $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$, which is used by the algorithm to make the gradient estimation.

In order to initialize the algorithm, suppose that, at $k = 0$, the measurements corresponding to the unknown function evaluated for the n strategies, i.e., $f(\mathbf{s}_1(0)), ..., f(\mathbf{s}_n(0))$ are known. Notice that since it is necessary to have n measurements to initialize the algorithm, and that each measurement is taken every time determined by τ, then the initial condition for the algorithm is guaranteed at least in a time given by $n\tau$.

Remark 3: In this case, there is only one available measurement at each $k \in \mathbb{Z}_{\geq 0}$ given by $f(\mathbf{s}_1(k))$, and it is assumed that this measurement is made at $k \in \mathbb{Z}_{\geq 0}$ by the node $1 \in \mathcal{V}_c$ for the strategy with index $1 \in \mathcal{I}$. ◇

In the algorithm, all the strategies are updated every iteration. However, in the algorithm only one new measurement appears at each iteration. The modification of the algorithm to treat the constraint of available information at each iteration is as follows:

$$\begin{aligned}\mathbf{s}_1(k+1) =& \mathbf{s}_1(k) + \frac{\theta_1(k)}{|\mathcal{N}_{a_1}|} \sum_{j \in \mathcal{N}_{a_1}} G(1, j) + \delta, \\ \mathbf{s}_i(k+1) =& \mathbf{s}_{i-1}(k), \;\; i = \{2, ..., n\}.\end{aligned}$$

Figure 2b) shows an example with only one available measurement at each iteration given by $f(\mathbf{s}_1(k))$. Notice that at the next iteration $k + 1$, the evaluation of functions is performed as $f(\mathbf{s}_2(k + 1)) = f(\mathbf{s}_1(k))$, $f(\mathbf{s}_3(k + 1)) = f(\mathbf{s}_2(k))$, and $f(\mathbf{s}_4(k+1)) = f(\mathbf{s}_3(k))$, which is information already stored by the algorithm.

IV. Model-free Control of wind farms

For the defined set of wind turbines $\mathcal{W}$, there is a topology represented by an undirected connected graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where $\mathcal{V}_t = \mathcal{W}$ is the set of nodes representing the wind turbines, and $\mathcal{E}_t$ describes the possible interaction and information sharing. In general, the graph describing the system tolopogy is determined according to the geographical position of the turbines. As mentioned before, each wind turbine $i \in \mathcal{W}$ generates a power denoted by $P_i(\mathbf{a}), \mathbf{a} = [a_1 \quad \cdots \quad a_m]^\top \in \mathbb{R}^m$ that depends on the behavior of the rest of the wind turbines (since the behavior of other wind turbines modifies the wind as shown in (3)).

Notice that in this control problem, it is only possible to make one measurement at each time $k \in \mathbb{Z}_{\geq 0}$ (i.e., there is only information about the total generated power for the current axial coefficients). Consequently, the suitable update of strategies to solve the control problem is the one introduced in Section III-C. In order to illustrate the performance of the control approach with this algorithm, two different control schemes are shown. First, it is assumed that each wind turbine has total information of the whole wind farm. Then, a centralized control scheme is presented. Afterwards, it is assumed that each wind turbine has partial information about the wind farm. Then, a decentralized control scheme is proposed.

A. Centralized control scheme

Figure 3 shows the centralized control scheme. In this scheme, each wind turbine sends the information of the generated power to a central coordinator. Consequently, such coordinator has full information about the wind farm, where an algorithm computes the new axial induction factors. Finally, the coordinator sends the computed axial coefficients back to all wind turbines.

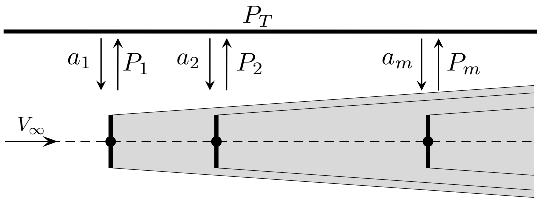

Fig. 3. Centralized control scheme. Each wind turbine sends its generated power to a central algorithm that optimizes the total power.

The optimization problem for the centralized algorithm is given by

$$\max \;\; P_T(\mathbf{s}) = \sum_{j \in \mathcal{W}} P_j(\mathbf{a}),$$

where $\mathbf{s} = \mathbf{a} \in \mathbb{R}^m$ is considered a strategy as presented in Section III, and $P_T : \mathbb{R}^m \mapsto \mathbb{R}$.

[2] Although the information sharing among the strategies must be guaranteed by a complete graph, this algorithm can be implemented for decentralized control solutions as in the control design proposed in this paper.

B. Decentralized control scheme

This control scheme, shown in Figure 4, is the typical one. It consists of a scheme in which each wind turbine $i \in \mathcal{W}$ disposes of the information about the total generated power P_T and its own axial induction factor a_i.

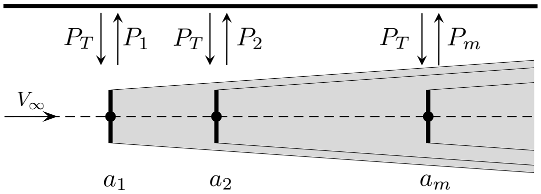

Fig. 4. Typical decentralized control scheme. Each wind turbine has information about the total generated power and its own axial induction factor.

In order to solve this control problem, it is proposed to have m algorithms computing in parallel, where each algorithm is associated to a partitioning of the whole optimization-based system. This scheme is decentralized since there is not information sharing or communication among the local algorithms. Let $\mathcal{P} = \{1, ..., m\}$ be the set of partitions in this decentralized control scheme (there are as many partitions as wind turbines in the wind farm, since each turbine has an algorithm associated). The algorithm at the wind turbine is in charge of solving the maximization power problem by considering all the axial induction factor of other wind turbines as exogenous parameters, i.e.,

$$\begin{aligned}\max\ P_T^p(s) =& P_T(s, a_{-i})\\ =& P_i(\mathbf{a}) + \sum_{j\in\mathcal{W}, j\neq i} P_j(\mathbf{a}),\end{aligned}$$

where P_T^p is the objective function of the partition $p \in \mathcal{P}$, a_{-i} represents all the axial induction factors except a_i, which are considered exogenous signals. The strategy for the population game is $s = a_i \in \mathbb{R}$ and $P_T^p : \mathbb{R} \mapsto \mathbb{R}$.

Notice that the algorithm assumes that the total power is represented by a function that only depends on the argument $a_i, i \in \mathcal{W}$, i.e., the algorithm assumes that other axial induction factors are exogenous signals. Moreover, the measurement over P_T^p is equal to the measurement over P_T.

V. Case study and simulation results

In order to illustrate the proposed control approaches, a wind farm with three turbines of 2 MW and 80 m of rotor diameter aligned with the constant freestream wind direction forming a path graph is controlled. The distance among turbines is five times the rotor diameter (400 m). For this scheme, the total generated power is given by

$$P_T(\mathbf{a}) = \frac{1}{2}\rho A\left(C_p(a_1)V_\infty^3 + C_p(a_2)V_2(\mathbf{a})^3 + C_p(a_3)V_3(\mathbf{a})^3\right),$$

where, from (3),

$$\begin{aligned}V_2(\mathbf{a}) =& V_\infty\left(1 - 2a_1c_{12}\right),\\ V_3(\mathbf{a}) =& V_\infty\left(1 - 2\sqrt{a_1^2c_{13}^2 + a_2^2c_{23}^2}\right).\end{aligned}$$

The model parameters have been set as $\beta = 0.075$ and $\rho = 1.225$ kg/m^3. For this simple example, the optimal axial coefficients can be analytically computed considering the model, and its result is $\mathbf{a}^* = [0.208\ 0.232\ 0.333]^\top$ [4].

Regarding the algorithm parameters corresponding to the presented simulation results, the centralized and decentralized approaches consider four data each iteration (one data for a current measurement, and three from stored data), i.e., $n = 4$, and the tuning parameters are selected to be $\tau = 0.01$s, $\epsilon = 0.8/k$, and $\gamma = 0.4$. Moreover, the communication graph within the each algorithm is given by $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, where $\mathcal{V}_c = \{1, 2, 3, 4\}$, and $\mathcal{E}_c = \{(1, 2), (1, 3), (1, 4)\}$. Finally, in order to initialize each algorithm, it is necessary to know the total generated power for m different initial strategies (composed by arbitrary coordinates in the domain of P_T). Initial conditions for the centralized approach are given by the following arbitrary strategies:

- $\mathbf{s}_1(0) = [0.1 \quad 0.3 \quad 0.2]^\top$,
- $\mathbf{s}_2(0) = [0.7 \quad 0.3 \quad 0.5]^\top$,
- $\mathbf{s}_3(0) = [0.2 \quad 0.2 \quad 0.2]^\top$,
- $\mathbf{s}_4(0) = [0.1 \quad 0.1 \quad 0.1]^\top$.

The decentralized scheme is composed by three different algorithms denoted by *A1*, *A2*, and *A3*, and their initial conditions are given by the following arbitrary strategies[3]:

- *A1:* $s_1(0) = 0.1$, $s_2(0) = 0.7$, $s_3(0) = 0.2$, and $s_4(0) = 0.1$,
- *A2:* $s_1(0) = 0.3$, $s_2(0) = 0.3$, $s_3(0) = 0.2$, and $s_4(0) = 0.1$,
- *A3:* $s_1(0) = 0.2$, $s_2(0) = 0.5$, $s_3(0) = 0.2$, and $s_4(0) = 0.1$.

Figure 5 presents the results obtained using the two proposed control schemes in Section IV, i.e., the centralized, and decentralized schemes. In the top plots, it can be observed the total power generated by the wind farm (Fig. 5a) and the power generated and axial coefficients for the three turbines (Fig. 5b and c, respectively). Clearly, for the freestream wind speed perfectly aligned with the three turbines, to set the axial coefficient of each individual turbine at the optimal value does not lead to maximum power production (3.77 MW). It can be seen than in less than 300 iterations the centralized algorithm is able to find the optimum value $\mathbf{a}^*$ corresponding to the maximum power production. As expected for this particular wind farm layout and wind speed direction, the last turbine operates at maximum efficiency and the other with less capacity in order the maximize the energy capture.

In the bottom plots of Fig. 5, it can be observed the results using the decentralized control scheme. In this case, the algorithm needs more iterations to reach the optimal axial induction coefficients. This is reasonable since each decentralized algorithm do not have information about the axial coefficient of other wind turbines. Even though, the algorithm is capable of finding the optimal solution rather fast. It is also worth of noticing that even with only local information about the axial induction factors, the decentralized control is able to achieve the same maximum power production obtained with the centralized scheme. Another advantage is the higher

[3] Notice that these initial conditions are the same for both the centralized and decentralized control approaches in order to compare their performance.

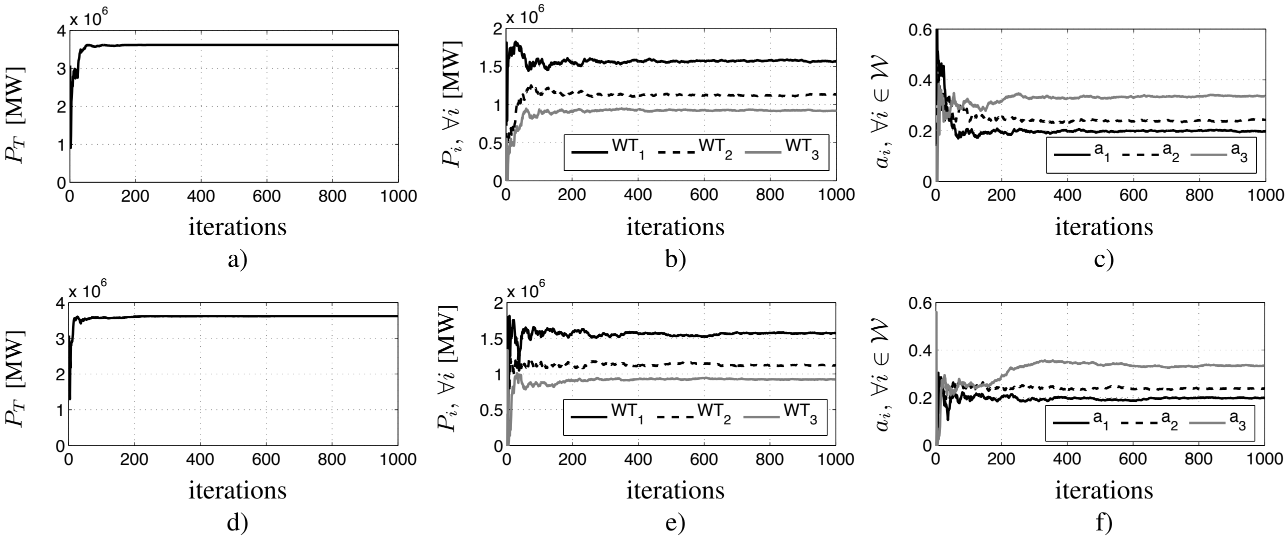

Fig. 5. Simulation results for the three wind turbine case study for the centralized scheme (Figures a), b), and c)), and for the decentralized scheme (Figures d), e) and f)). Results show the total power (Figures a), and d)), power for each wind turbines (Figures b), and e)), and the axial induction factors for the three wind turbines (Figures c), and f)).

reliability of the decentralized scheme given that it only uses partial information about the axial induction factors.

VI. Conclusions and further work

A model-free control for wind farms under centralized and decentralized control schemes has been proposed. Main differences with other algorithms in the literature have been pointed out. It has been shown that the proposed algorithm uses multiple gradient estimations to decide an improvement direction to increase the total generated power. Furthermore, it has been shown that the same convergence to the global solution is obtained for both schemes.

The centralized control scheme is composed by one coordinator that computes the m control variables and that sends them back to all the m wind turbines, whereas the decentralized control scheme is composed by m algorithms, each one of them computing one control variable for each wind turbine. In this regard, the algorithm in the centralized scheme has a higher computational burden than each one of the m algorithms in the decentralized approach (i.e., an algorithm with m decision variables with respect to an algorithm with one decision variable).

As further work, it is proposed to study more and different topologies. Moreover, it is challenging to solve the maximization power problem just by using partial information about the total generated power, i.e., a distributed control topology. For that, adaptations over the proposed algorithm can be made. On the other hand, this paper has presented an application of the algorithm in the maximization power of a wind farm. However, this algorithm is suitable for the solution of other model-free engineering problems in which a function is wanted to be maximized (e.g., water problems, transport problems, etc.), and where the only available information is obtained by measurements. Finally, due to the fact that it is possible to obtain the same power coefficient $C_p(a_i)$ for two different values of the axial coefficient a_i when $C_p(a_i) \neq 1/3$, then it is possible to constrain the values of the axial coefficients $0 \leq a_i \leq 1/3$, for all $i \in \mathcal{W}$ in the proposed algorithm. Furthermore, it is proposed to develop control strategies by using the proposed algorithm with a different control action, e.g., a reference of power generation at each turbine. In this regard, constraints of nominal power at each turbine may also be considered within the algorithm.

VII. Acknowledgements

This work has been partially supported by the projects ECOCIS (Ref. DPI2013-48243-C2-1-R), EFFINET (Ref. FP7-ICT-2011-8-31855), and ALTERNAR, Acuerdo 005, 07/19/13, CTeI-SGR-Nariño, Colombia.

References

[1] P. M. O. Gebraad and J. W. Van Wingerden. Maximum Power-point Tracking Control for Wind Farms. *Wind Energy*, 18(3):429–447, 2015.

[2] N. O. Jensen. A note on wind generator interaction. Technical report, Risø Institution, 1983.

[3] D. Madjidian. Dynamic power coordination for load reduction in dispatchable wind power plants. In *Proc. of the European Control Conference*, pages 3554–3559, 2013.

[4] J. Marden, S. Ruben, and L. Y. Pao. Surveying Game Theoretic Approaches for Wind Farm Optimization. In *Proc. of the 50th AIAA Aerospace Sciences Meeting*, pages 1–10, 2012.

[5] J. R. Marden, S. D. Ruben, and L. Y. Pao. A Model-Free Approach to Wind Farm Control Using Game Theoretic Methods. *IEEE Transactions on Control Systems Technology*, 21(4):1207–1214, 2013.

[6] A. Pantoja and N. Quijano. Distributed optimization using population dynamics with a local replicator equation. In *Proc. of the Conference on Decision and Control*, pages 3790–3795, 2012.

[7] W. H. Sandholm. *Population games and evolutionary dynamics*. Cambridge, Mass. MIT Press, 2010.

[8] S. N. Simic and S. Sastry. Distributed gradient estimation using random sensor networks. In *1st ACM Intern. Workshop on Wireless Sensor Networks and Applications (WSNA), Memorandum UCB/ERL M02/25*, 2002.

[9] V. Spudić, C. Conte, M. Baotić, and M. Morari. Cooperative distributed model predictive control for wind farms. *Optimal Control Applications and Methods*, 2014. (in press).