Distributed Power Optimization of Large Wind Farms Using ADMM for Real-Time Control

Zhiwei Xu, *Graduate Student Member, IEEE*, Bing Chu, Hua Geng, *Fellow, IEEE*, and Xiaohong Nian

***Abstract*—In a wind farm, the interactions between turbines caused by wakes can significantly reduce the power output of the wind farm. Cooperative control among the turbines has the potential to improve the power output. However, existing centralized power optimization methods are computationally expensive and does not scale well for large wind farms, limiting their practical use in real-time control for time-varying wind conditions and turbine configuration (with adding or maintaining of turbines). To address this problem, this paper proposes a fully distributed power optimization method for wind farms using alternating direction method of multipliers (ADMM). The proposed method allows the wind farm power output to be optimized in fully distributed manner with turbine-to-turbine message passing over a mesh network, guarantees the implemented control actions satisfy the control constraints of all turbines, and provably converges to a stationary point of the wind farm power optimization problem. Simulation results demonstrate that the proposed method can significantly reduce the computation time with hardly sacrificing the power gain compared with centralized method and thus is computationally efficient for real-time power optimization of large wind farms.**

***Index Terms*—Large wind farms, power optimization, nonconvex optimization, distributed optimization, ADMM.**

I. Introduction

In recent years, wind power, as one of the most important and prominent renewable resources, has experienced a rapid development to address the challenges caused by climate change, environmental pollution and increasing electricity demands [1]–[3]. According to the Global Wind Energy Council Report [4], there has been more than 93GW new wind power installation in 2020, which is an increase of over 53% compared to that of 2019. The global cumulative wind power capacity has risen up to 743GW with a growth of more than 14% compared to 2019. With this promising trend, advanced wind farm control policies are becoming increasingly important to develop more profitable wind farms [5], [6]. There are various wind farm control problems [7], examples of which include power maximization [8], active power control [9], [10], voltage and frequency control [11]. In this paper, we aim to maximize the power out of wind farm considering wake interactions among the turbines.

This work was supported by National Natural Science Foundation of China (NSFC) under Grant U2166601, Grant U2066602 and Grant 52061635102. *(Corresponding author: Hua Geng.)*

Z. Xu and H. Geng are with the Department of Automation, Tsinghua University, and Beijing National Research Center for Information Science and Technology, Beijing 100084, China (e-mail: xuzw18@mails.tsinghua.edu.cn; genghua@tsinghua.edu.cn).

B. Chu is with the School of Electronics and Computer Science, University of Southampton, Southampton SO171BJ, U.K. (e-mail: B.Chu@soton.ac.uk).

X. Nian is with the School of Automation, Central South University, Changsha 410075, China (e-mail: xhnian@csu.edu.cn).

In a wind farm, the turbines are often placed together to reduce installation, operation and maintenance costs [12]. However, this can also create strong aerodynamic interactions between turbines due to the existences of wakes, through which the upstream turbines could significantly decrease the wind speed and power output of downstream turbines and thus affect the power generation performance of the whole wind farm [13]. In practice, greedy policy is widely applied, where each turbine optimizes its own power output by neglecting the impact on downstream turbines. It often leads to suboptimal power output of whole wind farm [8], [14], which can cause over 30% power loss under some worst case scenarios [13].

Cooperative control between turbines considering the wake interactions shows great potential to improve the power generation efficiency of wind farm [15], which increases the power output of the wind farm by operating upstream turbines suboptimally and allowing for higher power output at downstream turbines. Various cooperative control policies have been proposed for wind farm power optimization, mainly including data-driven methods and model-based methods.

Data-driven methods maximize the power output of wind farm with only control inputs and some measurement data obtained by interacting with the wind farm. A decentralized safe experimentation dynamic (SED) method is proposed using game theory in [8], which can optimize the power output of wind farm without explicitly modeling the wake interactions between turbines. Then in [16], a distributed zeroth-order feedback optimization algorithm is developed using two-point gradient estimators, which shows quicker convergence speed than the SED method due to the efficient estimation of gradient information. However, the above two methods only consider the invariant wind conditions. In [17], a distributed simultaneous perturbation approach (D-SPA) is developed, and in [18], two discrete adaptive filtering algorithms (DAFA) are proposed to optimize the power output of wind farm. Note that the D-SPA and DAFA accommodate slowly changing wind conditions. To deal with more complex wind conditions, a hierarchical stochastic projected simplex method is proposed in [19], which can quickly improve the power output of wind farm due to the full use of the learned knowledge. It is worth mentioning that the aforementioned methods do not require a power generation model of wind farm. Therefore, they are suitable for the wind farms that cannot be modelled easily, such as wind farms sited in coteau or highland area. However, a large number of measurement data are commonly required by data-driven methods to achieve an optimum, often resulting in slow convergence speed and likely lower power efficiency.

The model-based optimization methods usually have faster

convergence speed compared to data-driven methods and can improve the power output of wind farms in a short time thanks to the use of power generation models, providing a better choice for the wind farms whose power generation models can be easily obtained. Most existing model-based optimization methods are implemented in a centralized manner. For example in [20], sequential convex programming is used to maximize the power output of wind farm. And in [21], [22], the particle swarm optimization algorithm is applied and the results show that there is a noticeable increase in the amount of wind farm power output. For small wind farms, the centralized optimization methods can commonly be performed in real time and thus have the ability to adapt to changing wind and turbine conditions, e.g. individual turbines are unavailable due to maintenance or new turbines are installed. However, as modern wind farms tend to be large-scale, the above centralized methods can face serious difficulties for real-time control of large wind farms due to the more complex nature of the power generation models (in particular, its high dimensionality), which require significant computation load.

To address the above limitations, some model-based distributed optimization methods have been proposed for the large wind farm power optimization. In [23], the power output of the wind farm is maximized among the subsets using alternating direction method of multipliers (ADMM), where all turbines are grouped into smaller subsets based on wake interactions between turbines. This is extended in [24] using a combination of ADMM and reinforcement learning to maximize the power output of wind farm among the divided subsets. A model-based distributed power optimization approach with convergence guarantee is presented using proximal primal-dual gradient method [25], which allows for turbines to be optimized in parallel using local information. The simulation results show that the above distributed optimization methods are computationally efficient and can achieve real-time power optimization of large wind farms.

Note however that none of the above model-based distributed optimization methods consider the control constraints of turbines and thus cannot guarantee the feasibility of the resulting control action. The infeasible control action cannot be carried out by real turbines and even likely damages turbines, thus increasing maintenance cost of whole wind farm, which are not desired in practice. Though a saturation operation on the control action can be applied before applying it to the turbines, this ad-hoc approach would generally lead to suboptimal wind farm power generation performance. Note that a non-centralized predictive control strategy is proposed in [9] for wind farm active power control, which considers the turbine power limits and reduces the computational burden. However, the strategy has difficulties in maximizing the power output of the wind farm due to the underlying nonlinear wind farm power generation model.

In this paper, we develop a ADMM-based distributed power optimization method considering turbine control constraints for large wind farms that can be modelled to efficiently achieve real-time power optimization. The main contributions of this paper are as follows: (a) The control constraints handling in wind farm power optimization problem, that have great practice importance but have not been considered in the literatures, have been considered in this paper by formulating the resulting problem as a constrained nonconvex general form consensus optimization problem. (b) A distributed optimization method for constrained nonconvex general form consensus optimization problem is proposed using ADMM, further extending the results in [26], [27]. A rigorous convergence proof of the resulting method is given. (c) The proposed method is applied to the wind farm power optimization problem with control constraints as an application example. The design ensures implemented control action feasible for real turbines and it can be shown that the method can provably identify a stationary point of the wind farm power optimization problem under moderate assumptions—note this is nontrivial due to the nonconvex nature of the power optimization problem. Meanwhile, it allows for turbines to be optimized in fully parallel fashion via turbine-to-turbine message passing over a mesh network and thus guarantees high computation efficiency, scalability and reliability.

The rest of this paper is organized as follows. In Section II, the power optimization problem of wind farm is formulated. In Section III, an ADMM-based distributed power optimization method is proposed for large wind farms to achieve real-time optimization. The convergence properties of the proposed method are analyzed in Section IV. Simulation results are then given to demonstrate the performance of the proposed method in Section V. Finally, Section VI presents the conclusion and possible directions for future research.

II. Wind Farm Power Optimization Problem

In this section, the power generation model of wind farm is introduced, and then the power optimization problem is given.

A. Power Model

A wind farm with n wind turbines is considered. Let $\mathcal{N} = \{1, 2, \cdots, n\}$ represent the set of all turbines. Suppose that the blade disk planes of all turbines are perpendicular to the wind direction. The control action of turbine $i \in \mathcal{N}$ is then selected as its axial induction factor (AIF) u_i. The AIF is a measure of the wind velocity decrease over turbine rotor plane and can be controlled by the blade pitch and generator torque of the turbine [28], which provides a relatively simple expression and is widely used in wind farm control. The feasible domain of the AIF u_i is denoted as $\mathcal{U}_i = \{u_i \,|u_{i,min} \leq u_i \leq u_{i,max}\}$, where the $u_{i,min}$ and $u_{i,max}$ are respectively the lower bound and upper bound of the u_i, $i \in \mathcal{N}$. The joint AIF of all turbines is represented by the tuple $\boldsymbol{u} = (u_1, \cdots, u_n)$, whose admissible set is denoted as $\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_n$ and $\times$ is the Cartesian product.

The aggregate wind velocity $V_i(\{u_j\}_{j\in\mathcal{N}_i})$ at an arbitrary turbine $i \in \mathcal{N}$ can be given by

$$V_i(\{u_j\}_{j\in\mathcal{N}_i}) = V_\infty \left(1 - \delta V_i(\{u_j\}_{j\in\mathcal{N}_i})\right), \tag{1}$$

where $\mathcal{N}_i$ denotes the set of upstream turbines that are coupled with turbine i through wakes, V_∞ denotes the free-stream wind speed, $\delta V_i(\{u_j\}_{j\in\mathcal{N}_i})$ is the wind speed deficit at turbine i

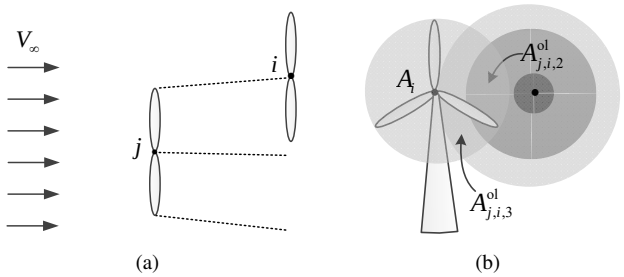


Fig. 1. Wake interaction example. (a) Wake interaction. (b) Cut-through at downstream turbine i.

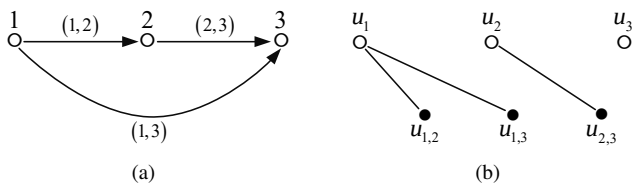


Fig. 2. Three-turbine wind farm example. (a) Wake interaction graph, where nodes represent turbines and directed edges represent interactions among the turbines via wakes. (b) Consensus network, where nodes represent optimization variables and undirected edges represent consensus constraints.

quantifying the reduction of the wind speed in wakes. The power generated by turbine i can be represented as

$$P_i(u_i;\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i}) = K_{p,i}(u_i)V_i(\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i})^3, \tag{2}$$

where $K_{p,i}(u_i) = \frac{1}{2}\rho_a A_i C_{p,i}(u_i)$, ρ_a denotes air density, A_i is the disk area generated by the blade of turbine i, $C_{p,i}(u_i)$ is the power coefficient defined by

$$C_{p,i}(u_i) = 4k_l u_i(1-u_i)^2, \tag{3}$$

and k_l is loss factor.

The goal of the wake interaction modelling is to identify the wind speed deficit $\delta V_i(\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i})$ in (1). The FLOw Redirection and Induction in Steady-state (FLORIS) model [29] is used in this paper to represent the wake interactions among the turbines due to its wide application in wind farm control. According to the FLORIS model, the wind speed deficit $\delta V_i(\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i})$ can be described as follows:

$$\delta V_i\left(\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i}\right) = 2\sqrt{\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left(u_j\sum_{q=1}^{3}c_{j,q}\left(d_i(\theta)\right)a_{j,i,q}\right)^2}, \tag{4}$$

where

$$a_{j,i,q} = \min\left(A^{\text{ol}(\theta)}_{j,i,q}/A_i, 1\right),$$

$c_{j,q}$ is the local wake decay coefficient of turbine j in q_{th} wake zone, $d_i(\theta)$ denotes the distance of turbine i from a common vertex along the wind direction θ, $A^{\text{ol}}_{j,i,q}$ is the overlapping area between the q_{th} wake zone of turbine j and the disk area A_i of turbine i and is related with wind direction θ. Consider the wake interaction examples shown in Fig. 1. In Fig. 1(a), the direction that the arrow points to represents wind direction and between the top and bottom dotted lines denotes the wake area of turbine j, where turbine i is affected by the wake of turbine j due to non-zero overlapping area. Fig. 1(b) shows the cut-through at downstream turbine i, where it can be observed that $A^{\text{ol}}_{j,i,2}$ and $A^{\text{ol}}_{j,i,3}$ are not equal to 0. The details for FLORIS model can be found in [29].

B. Problem Formulation

The power output of wind farm can be expressed as

$$P(\boldsymbol{u}) = \sum\nolimits_{i=1}^{n} P_i(u_i;\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i}), \tag{5}$$

which is the sum of the power generated by all turbines. Then the power optimization problem of wind farm can be formulated as follows:

$$\begin{aligned} &\text{maximize} && P(\boldsymbol{u}) \\ &\text{subject to} && \boldsymbol{u}\in\boldsymbol{\mathcal{U}}, \end{aligned} \tag{6}$$

Note that solving this optimization problem is not trivial as it is a nonconvex optimization problem (due to the nature of the power generation model described above) with bound constraint and is high-dimensional for large wind farms.

Remark 1: From (2) and (3), it can be easily found that the choice of $u_i = 1/3$ for turbine $i \in \boldsymbol{\mathcal{N}}$ results in maximum power output (of turbine i) and is thus called greedy policy. However, as mentioned earlier, the greedy policy might not be optimal for problem (6) in maximizing the total power output of wind farm due to the wake interactions among the turbines.

III. ADMM-based Distributed Power Optimization for Large Wind Farms

In this section, the wind farm power optimization problem is equivalently formulated as a constrained nonconvex general form consensus optimization problem, for which a distributed optimization method is proposed using ADMM. A fully distributed power optimization method is then developed for large wind farms considering control constraints by applying the proposed method.

A. Wind Farm Power Optimization as a General Form Consensus Problem

To achieve the consensus formulation of wind farm power optimization problem, we first need to characterize the wake interactions among the turbines by considering graph notations. In a wind farm, the upstream turbines along the wind direction likely decrease the wind speed and power output of downstream turbines but not necessarily vice versa. Therefore in this paper, a directed graph $\boldsymbol{\mathcal{G}}=(\boldsymbol{\mathcal{N}},\boldsymbol{\mathcal{E}})$ is used to capture the interactions between turbines, where $\boldsymbol{\mathcal{N}} = \{1,\cdots,n\}$ denotes the set of all turbines as mentioned earlier and $\boldsymbol{\mathcal{E}} \subseteq \boldsymbol{\mathcal{N}}\times\boldsymbol{\mathcal{N}}$ represents a set of directed edges. An edge $(j,i)\in\boldsymbol{\mathcal{E}}$ implies that the upstream turbine j reduces the wind speed and power output of the downstream turbine i via its wake. Recall the definition of $\boldsymbol{\mathcal{N}}_i$ and we have $(j,i)\in\boldsymbol{\mathcal{E}}$ for $\forall j\in\boldsymbol{\mathcal{N}}_i$, $i\in\boldsymbol{\mathcal{N}}$. Note that once $(j,i)\in\boldsymbol{\mathcal{E}}$, $(i,j)\notin\boldsymbol{\mathcal{E}}$. Let $\bar{\boldsymbol{\mathcal{N}}}_i$ denote the set of downstream turbines that the turbine i affects by its wake. A three-turbine wind farm example is shown in Fig. 2(a), where wind direction θ points horizontally from left to right, $\boldsymbol{\mathcal{N}} = \{1,2,3\}$ and $\boldsymbol{\mathcal{E}} = \{(1,2),(1,3),(2,3)\}$; Turbine 1 is not interfered by any other turbines, turbine 2 is impacted

by its upstream turbine 1 along θ, and turbine 3 is affected by turbines 1 and 2, i.e., $\boldsymbol{\mathcal{N}}_1=\varnothing$, $\boldsymbol{\mathcal{N}}_2=\{1\}$ and $\boldsymbol{\mathcal{N}}_3=\{1,2\}$; On the other hand, the turbine 1 affects turbines 2 and 3 by its wake, turbine 2 affects turbine 3, and turbine 3 does not impact any other turbines, i.e., $\bar{\boldsymbol{\mathcal{N}}}_1=\{2,3\}$, $\bar{\boldsymbol{\mathcal{N}}}_2=\{3\}$, $\bar{\boldsymbol{\mathcal{N}}}_3=\varnothing$.

Let

$$f_i(u_i;\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i})=-P_i(u_i;\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i}). \tag{7}$$

Based on (5), the wind farm power optimization problem (6) is equivalent to

$$\begin{aligned}&\text{minimize} \quad \sum\nolimits_{i=1}^{n} f_i(u_i;\{u_j\}_{j\in\boldsymbol{\mathcal{N}}_i})\\ &\text{subject to} \quad \boldsymbol{u}\in\boldsymbol{\mathcal{U}}.\end{aligned} \tag{8}$$

According to graph $\boldsymbol{\mathcal{G}}=(\boldsymbol{\mathcal{N}},\boldsymbol{\mathcal{E}})$, the auxiliary optimization variables $\{u_{j,i}\}_{j\in\boldsymbol{\mathcal{N}}_i}$ are introduced for turbine i, where $u_{j,i}$ denotes a copy of the control action u_j of turbine j and is stored locally at turbine i. Then the problem (8) can be rewritten as a consensus optimization problem as follows:

$$\begin{aligned}&\text{minimize} \quad \sum\nolimits_{i=1}^{n} f_i(u_i;\{u_{j,i}\}_{j\in\boldsymbol{\mathcal{N}}_i})\\ &\text{subject to} \quad u_{j,i}=u_j, \quad j\in\boldsymbol{\mathcal{N}}_i, \quad i\in\boldsymbol{\mathcal{N}},\\ &\qquad\qquad \boldsymbol{u}\in\boldsymbol{\mathcal{U}}.\end{aligned} \tag{9}$$

where $u_{j,i}=u_j$ is the consensus constraint of ensuring that the optimization variable $u_{j,i}$ of turbine i is consistent with the control action u_j of the upstream turbine $j\in\boldsymbol{\mathcal{N}}_i$. There are $|\boldsymbol{\mathcal{E}}|$ consensus constraints in (9), where $|\boldsymbol{\mathcal{E}}|$ denotes the cardinality of the set $\boldsymbol{\mathcal{E}}$. In the given three-turbine wind farm example (see Fig. 2(a)), there are three consensus constraints, i.e. $u_{1,2}=u_1$, $u_{1,3}=u_1$, $u_{2,3}=u_2$ (see Fig. 2(b)).

The local variable $\boldsymbol{x}_i=(u_i,\{u_{j,i}\}_{j\in\boldsymbol{\mathcal{N}}_i})$ is defined for turbine $i\in\boldsymbol{\mathcal{N}}$. Let $n_i=|\boldsymbol{x}_i|$. Note that the $\boldsymbol{x}_i$ is composed of the control action u_i of turbine i and the introduced auxiliary optimization variables $\{u_{j,i}\}_{j\in\boldsymbol{\mathcal{N}}_i}$ for turbine i. Then $n_i=1+|\boldsymbol{\mathcal{N}}_i|$. The objective function in (9) can be denoted as

$$\sum\nolimits_{i=1}^{n} f_i(\boldsymbol{x}_i). \tag{10}$$

In fact, each local variable $\boldsymbol{x}_i$ consists of a selection of the components of global variable $\boldsymbol{u}$ (i.e., joint AIF of all turbines). This means that each component of local variable $\boldsymbol{x}_i$ corresponds to one element of global variable $\boldsymbol{u}$. Define $h=\mathcal{Z}(i,l)$ as a mapping from local variable index into global variable index, which implies that the l_{th} component of the local variable $\boldsymbol{x}_i$ corresponds to the h_{th} component of global variable $\boldsymbol{u}$. Then in (9), the consensus constraints between local variables and global variable can be rewritten as

$$(\boldsymbol{x}_i)_l=\boldsymbol{u}_{\mathcal{Z}(i,l)}, \quad l=1,\cdots,n_i, \quad i\in\boldsymbol{\mathcal{N}}. \tag{11}$$

In the three-turbine wind farm example shown by Fig. 2, $\boldsymbol{x}_1=(u_1)$, $\boldsymbol{x}_2=(u_2,u_{1,2})$, $\boldsymbol{x}_3=(u_3,u_{1,3},u_{2,3})$, $1=\mathcal{Z}(1,1)$, $2=\mathcal{Z}(2,1)$, $2=\mathcal{Z}(3,3)$. Define $\tilde{\boldsymbol{u}}_i$ by $(\tilde{\boldsymbol{u}}_i)_l=\boldsymbol{u}_{\mathcal{Z}(i,l)}$, $l=1,\cdots,n_i$, $i\in\boldsymbol{\mathcal{N}}$. Intuitively, the $\tilde{\boldsymbol{u}}_i$ is made up of the components of $\boldsymbol{u}$ that $\boldsymbol{x}_i$ corresponds to. Then the consensus constraints (11) can be further rewritten as

$$\boldsymbol{x}_i=\tilde{\boldsymbol{u}}_i, \quad i\in\boldsymbol{\mathcal{N}}. \tag{12}$$

Hence, the problem (9) can now be equivalently written into

$$\begin{aligned}&\text{minimize} \quad \sum\nolimits_{i=1}^{n} f_i(\boldsymbol{x}_i)\\ &\text{subject to} \quad \boldsymbol{x}_i=\tilde{\boldsymbol{u}}_i, \quad i\in\boldsymbol{\mathcal{N}},\\ &\qquad\qquad \boldsymbol{u}\in\boldsymbol{\mathcal{U}}.\end{aligned} \tag{13}$$

Remark 2: The (13) shows that the wind farm power optimization problem (6) can be formulated as a constrained general form consensus optimization problem, where each local variable only contains a small number of the components of global variable. However, it is nontrivial to solve the problem (13) since this is a high-dimensional nonconvex optimization problem for large wind farms which is often computationally expensive but real-time control is required in practice to adapt to time-varying wind conditions and turbine configuration.

Remark 3: Note that the wind farm power optimization problem is formulated in (6) considering the control constraints of all turbines. In the equivalent formulation (13) of problem (6), the control constraints are also considered. This is different from the designs in [23]–[25], where the control constraints that are often crucially important in applications are not considered. However, the consideration of the constraints increases the solving difficulty of the problem and the complexity of algorithm's convergence analysis.

B. ADMM-based Distributed Optimization for Constrained Nonconvex General Form Consensus Problem

The above problem (13) can be seen as a special case of the following more general nonconvex general form consensus optimization problem which we will consider first:

$$\begin{aligned}&\text{minimize} \quad \sum\nolimits_{i=1}^{n} f_i(\boldsymbol{x}_i)+h(\boldsymbol{u})\\ &\text{subject to} \quad \boldsymbol{x}_i=\tilde{\boldsymbol{u}}_i, \quad i\in\boldsymbol{\mathcal{N}},\\ &\qquad\qquad \boldsymbol{u}\in\boldsymbol{\mathcal{U}},\end{aligned} \tag{14}$$

where $f_i:\mathbb{R}^{n_i}\mapsto\mathbb{R}$ is a smooth and possibly nonconvex function, $h:\mathbb{R}^{n}\mapsto\mathbb{R}$ is a convex function, $n_i\le n$, $\boldsymbol{u}=(u_1,\cdots,u_n)$ is defined as global variable, $\tilde{\boldsymbol{u}}_i$ consists of the components of $\boldsymbol{u}$ that $\boldsymbol{x}_i$ corresponds to, $\boldsymbol{\mathcal{N}}=\{1,2,\cdots,n\}$, $\boldsymbol{\mathcal{U}}$ is a closed convex set.

The augmented Lagrangian function is defined for (14) by

$$\mathcal{L}_\rho(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda})=\sum\nolimits_{i=1}^{n}\mathcal{L}_i(\boldsymbol{x}_i,\tilde{\boldsymbol{u}}_i,\boldsymbol{\lambda}_i)+h(\boldsymbol{u}), \tag{15}$$

where $\boldsymbol{x}=(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)$, $\boldsymbol{\lambda}=(\boldsymbol{\lambda}_1,\cdots,\boldsymbol{\lambda}_n)$ is dual variable,

$$\mathcal{L}_i(\boldsymbol{x}_i,\tilde{\boldsymbol{u}}_i,\boldsymbol{\lambda}_i)=f_i(\boldsymbol{x}_i)+\boldsymbol{\lambda}_i^T(\boldsymbol{x}_i-\tilde{\boldsymbol{u}}_i)+\frac{\rho}{2}\|\boldsymbol{x}_i-\tilde{\boldsymbol{u}}_i\|^2, \tag{16}$$

and $\rho>0$ is penalty parameter.

The problem (14) can be solved iteratively using ADMM derived directly from (15) with the following steps:

$$\boldsymbol{u}^{k+1}=\underset{\boldsymbol{u}\in\boldsymbol{\mathcal{U}}}{\operatorname{argmin}}\,\mathcal{L}_\rho\left(\boldsymbol{x}^k,\boldsymbol{u},\boldsymbol{\lambda}^k\right), \tag{17}$$

$$\boldsymbol{x}^{k+1}=\underset{\boldsymbol{x}}{\operatorname{argmin}}\,\mathcal{L}_\rho\left(\boldsymbol{x},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^k\right), \tag{18}$$

$$\boldsymbol{\lambda}^{k+1}=\boldsymbol{\lambda}^k+\rho\left(\boldsymbol{x}^{k+1}-\tilde{\boldsymbol{u}}^{k+1}\right), \tag{19}$$

where $\tilde{\boldsymbol{u}}^{k+1}=\left(\tilde{\boldsymbol{u}}_1^{k+1},\cdots,\tilde{\boldsymbol{u}}_n^{k+1}\right)$. Note that the augmented Lagrangian function $\mathcal{L}_\rho(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda})$ in (15) is fully separable

Algorithm 1: ADMM-based Distributed Power Optimization Method for Large Wind Farms

Initialization: $\boldsymbol{x}_i^0$, $\boldsymbol{\lambda}_i^0$, ρ

For $k = 0, 1, \cdots$

Step 1: Local communication

communicate with neighboring turbine $j \in \bar{\boldsymbol{\mathcal{N}}}_i$ and obtain $\boldsymbol{x}_j^k$, $\boldsymbol{\lambda}_j^k$

Step 2: Action update

update u_i by (22) and obtain u_i^{k+1}

Step 3: Local communication

communicate with neighboring turbine $j \in \boldsymbol{\mathcal{N}}_i$ and obtain u_j^{k+1}, and then form $\tilde{\boldsymbol{u}}_i^{k+1}$

Step 4: Local variable update

update $\boldsymbol{x}_i$ by (20) and obtain $\boldsymbol{x}_i^{k+1}$

Step 5: Dual variable update

update $\boldsymbol{\lambda}_i$ by (21) and obtain $\boldsymbol{\lambda}_i^{k+1}$

in all components of $\boldsymbol{x}$ and $\boldsymbol{\lambda}$. Then in (18) and (19), the update steps of $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ can fully decouple across their own components, namely:

$$\boldsymbol{x}_i^{k+1} = \underset{\boldsymbol{x}_i}{\operatorname{argmin}} \mathcal{L}_i\left(\boldsymbol{x}_i, \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k\right), \tag{20}$$

$$\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho\left(\boldsymbol{x}_i^{k+1} - \tilde{\boldsymbol{u}}_i^{k+1}\right), \tag{21}$$

Therefore, the problem (14) can be solved by applying distributed ADMM composed of (17), (20) and (21).

C. ADMM-based Distributed Power Optimization

As mentioned earlier, the equivalent formulation (13) of wind farm power optimization problem is a special case of problem (14) that $h(\boldsymbol{u}) = 0$. Then the (13) can be solved directly using the resulting ADMM consisting of (17), (20) and (21). As $h(\boldsymbol{u}) = 0$ in (13), the augmented Lagrangian function $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda})$ is fully separable in all components of $\boldsymbol{u}$ and thus the update step of $\boldsymbol{u}$ in (17) can fully decouple across its own components, namely:

$$u_i^{k+1} = \prod\nolimits_{\boldsymbol{\mathcal{U}}_i}\left(\frac{\sum_{\mathcal{Z}(j,l)=i}\left(\left(\boldsymbol{x}_j^k\right)_l + (1/\rho)(\boldsymbol{\lambda}_j^k)_l\right)}{\sum_{\mathcal{Z}(j,l)=i} 1}\right), \tag{22}$$

where $j \in \bar{\boldsymbol{\mathcal{N}}}_i \bigcup \{i\}$, $l{=}1, \cdots, n_j$, $i \in \boldsymbol{\mathcal{N}}$. Therefore, a distributed power optimization algorithm, combination of (22), (20) and (21), is proposed for large wind farms. It is further tabulated as shown in Algorithm 1.

It is worth mentioning that in Algorithm 1, the turbine i can calculate locally u_i^{k+1} by (22) upon receiving $\boldsymbol{x}_j^k$ and $\boldsymbol{\lambda}_j^k$ from all neighboring turbines $j \in \bar{\boldsymbol{\mathcal{N}}}_i$. It can also compute locally $\boldsymbol{x}_i^{k+1}$ by (20) after obtaining u_j^{k+1} from all neighboring turbines $j \in \boldsymbol{\mathcal{N}}_i$ and forming $\tilde{\boldsymbol{u}}_i^{k+1}$. Meanwhile, the calculation of $\boldsymbol{\lambda}_i^{k+1}$ can be independently carried out at turbine i. Therefore, the proposed Algorithm 1 is fully distributed, and All turbines can run this algorithm in parallel manner by communicating with their neighboring turbines.

Remark 4: Recall that $\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{U}}_1 \times \cdots \times \boldsymbol{\mathcal{U}}_n$. The constraint $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}$ in (13) is equal to $u_i \in \boldsymbol{\mathcal{U}}_i$ for any turbine $i \in \boldsymbol{\mathcal{N}}$. In (22), the u_i^{k+1} is found by projecting the average value of the entries of $\boldsymbol{x}_j^k + (1/\rho)(\boldsymbol{\lambda}_j^k)$ that correspond to the global index i onto the feasible domain $\boldsymbol{\mathcal{U}}_i$ of u_i, where the projection operator $\Pi_{\boldsymbol{\mathcal{U}}_i}(\bullet)$ is used to handle the control constraint $u_i \in \boldsymbol{\mathcal{U}}_i$ of turbine i. This implies that the developed Algorithm 1 for the problem (13) can guarantee the feasibility of the optimal control action for real turbines.

IV. Convergence and Implementation of the Proposed Method

In this section, the convergence property of the proposed method is analyzed and then its implementation is illustrated.

A. Convergence Analysis

While the convergence of ADMM can be shown for convex problems, a rigorous proof for nonconvex problem is challenging. Recently, promising progress has been made on this including the results in [26], [27]. In [26], for the nonconvex global variable consensus problem, the convergence of asynchronous ADMM is analyzed. The convergence is further analyzed in [27] for the unconstrained nonconvex general form consensus optimization problem. This paper further extends these results and considers constrained nonconvex general form consensus optimization problem. Convergence using ADMM for this problem is analyzed in detail and the wind farm power optimization problem is studied as an application example.

Assumption 1: ∇f_i is Lipschitz continuous with positive constant $L > 0$, such that for any $\boldsymbol{x}_i'$, $\boldsymbol{x}_i''$,

$$\left\|\nabla f_i\left(\boldsymbol{x}_i'\right) - \nabla f_i\left(\boldsymbol{x}_i''\right)\right\| \le L\left\|\boldsymbol{x}_i' - \boldsymbol{x}_i''\right\|. \tag{23}$$

Assumption 2: The penality parameter ρ is chosen large enough such that

1) The $\mathcal{L}_i\left(\boldsymbol{x}_i, \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k\right)$ of problem (20) is strongly convex with modulus γ_i, i.e. for any $\boldsymbol{x}_i'$, $\boldsymbol{x}_i''$,

$$\begin{aligned}&\mathcal{L}_i\left(\boldsymbol{x}_i', \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k\right) - \mathcal{L}_i\left(\boldsymbol{x}_i'', \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k\right)\\ \le&\nabla_{\boldsymbol{x}_i}\mathcal{L}_i\left(\boldsymbol{x}_i', \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k\right)^T\left(\boldsymbol{x}_i' - \boldsymbol{x}_i''\right) - \frac{\gamma_i}{2}\left\|\boldsymbol{x}_i' - \boldsymbol{x}_i''\right\|^2.\end{aligned} \tag{24}$$

2) $\rho\gamma_i \ge 2L^2$ and $\rho \ge L$, $i \in \boldsymbol{\mathcal{N}}$.

Assumption 3: The objective function of problem (14) has a lower bound f^* while satisfying the constraint.

Then, we have the following main theorem regarding the convergence performance of the ADMM for problem (14).

Theorem 1: Suppose the above assumptions hold. Then the iteration sequence $\left\{\{\boldsymbol{x}_i^k\}, \boldsymbol{u}^k, \{\boldsymbol{\lambda}_i^k\}\right\}$ of the ADMM converges to the set of stationary solutions of problem (14), i.e.

$$\lim_{k\to\infty} \operatorname{dist}\left(\left(\{\boldsymbol{x}_i^k\}, \boldsymbol{u}^k, \{\boldsymbol{\lambda}_i^k\}\right), \boldsymbol{Z}^*\right) = 0, \tag{25}$$

where $\boldsymbol{Z}^*$ is the set of stationary solutions $(\{\boldsymbol{x}_i^*\}, \boldsymbol{u}^*, \{\boldsymbol{\lambda}_i^*\})$ of problem (14), i.e.

$$\boldsymbol{u}^* \in \underset{\boldsymbol{u}\in\boldsymbol{\mathcal{U}}}{\operatorname{argmin}} h(\boldsymbol{u}) + \sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^*, \boldsymbol{x}_i^* - \tilde{\boldsymbol{u}}_i\right\rangle, \tag{26}$$

$$\nabla f_i\left(\boldsymbol{x}_i^*\right) + \boldsymbol{\lambda}_i^* = 0, \tag{27}$$

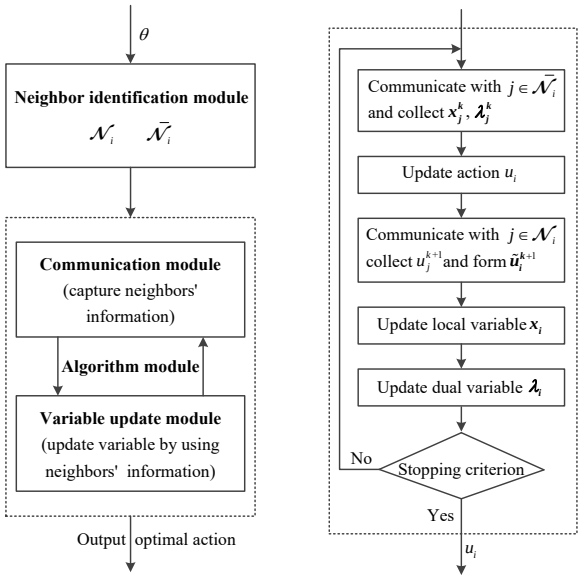


Fig. 3. Control scheme of turbine i. (a) Control framework of turbine i. (b) Algorithm flowchart of turbine i.

$$\boldsymbol{x}_i^* = \tilde{\boldsymbol{u}}_i^*, \ i \in \boldsymbol{\mathcal{N}}, \tag{28}$$

$\mathrm{dist}(\boldsymbol{y}, \boldsymbol{Z}^*)$ denotes the distance between a vector $\boldsymbol{y}$ and the set $\boldsymbol{Z}^*$, i.e.

$$\mathrm{dist}\left(\boldsymbol{y}, \boldsymbol{Z}^*\right) = \min_{\boldsymbol{z} \in \boldsymbol{Z}^*} \|\boldsymbol{y} - \boldsymbol{z}\|. \tag{29}$$

Proof: See the Appendix A. ■

For wind farm power optimization problem, the Assumption 1 means that the gradients of the turbine power generation models are required to be Lipschitz continuous. The Assumption 2 implies that the selected penality parameter ρ in (16) should guarantee $\mathcal{L}_i\left(\boldsymbol{x}_i, \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k\right)$ with nonconvex turbine power generation models strongly convex with $\boldsymbol{x}_i$ in (20). Then the ρ should be chosen large enough to achieve convergence when the proposed algorithm is implemented. The Assumption 3 means that the power output of wind farm is bounded above. It is worth mentioning that the above assumptions are moderate—they are technical assumptions needed to prove the convergence of the proposed method and impose no restriction on how the turbines should be built or controlled: The Assumption 1 and 3 can be trivially met for physical systems including the wind turbine and one can always find a large enough ρ such that Assumption 2 holds. As a consequence, applying the proposed ADMM method to wind farm power optimization problem has the following convergence property:

Corollary 1: The iteration sequence $\left\{\boldsymbol{u}^k\right\}$ generated by Algorithm 1 converges to the set of stationary solutions of problem (6).

B. Implementation

The control scheme of turbine i is given in Fig. 3, $i \in \boldsymbol{\mathcal{N}}$. The scheme consists of neighbor identification module and algorithm module, as shown in Fig. 3(a). The neighbor identification module of turbine i identifies its neighbor sets $\boldsymbol{\mathcal{N}}_i$ and $\bar{\boldsymbol{\mathcal{N}}}_i$ for newly measured wind direction θ. Note that the neighbor sets may change with changes of wind direction as different wind directions likely lead to different wake interaction patterns among the turbines. The algorithm module includes two submodules, namely communication module and variable update module. The communication module collects the information of neighboring turbines according to the requirements of variable update module and identified neighbor sets. The variable update module updates optimization variables by using collecting information.

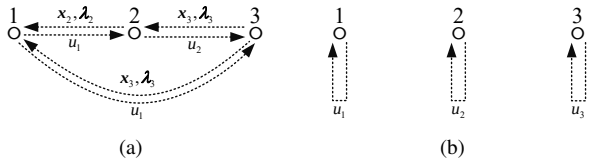


Fig. 4. Three-turbine wind farm control example. (a) Turbine communication. (b) Control implementation.

The flowchart of algorithm module of turbine i is detailedly explained in Fig. 3(b), where the control action u_i is updated using (22) by exploiting the information from itself and its neighbor set $\bar{\boldsymbol{\mathcal{N}}}_i$, and local variable $\boldsymbol{x}_i$ and dual variable $\boldsymbol{\lambda}_i$ are updated using (20) and (21) by exploiting the information from itself and its neighbor set $\boldsymbol{\mathcal{N}}_i$. Note that if $\bar{\boldsymbol{\mathcal{N}}}_i = \varnothing$, the turbine i is in the last row along the wind direction and its wake does not affect any other turbines. In this case, the turbine i updates its action u_i only exploiting its own information. On the other hand if $\boldsymbol{\mathcal{N}}_i = \varnothing$, the turbine i is in the first row along the wind direction and no turbine can affect it by wakes, in which the turbine i updates its variables $\boldsymbol{x}_i$ and $\boldsymbol{\lambda}_i$ only exploiting its own information. The optimal control action u_i would be given by algorithm module when the stopping criterion is satisfied. Note that all the optimization variables of each turbine are updated at the turbine in sequential manner. An illustrative three-turbine wind farm control example is given in Fig. 4; Fig. 4(a) shows the communications between turbines for the wake interaction graph in Fig. 2(a); Fig. 4(b) indicates that each turbine controls independently its state using optimal control action given by its algorithm module.

In the following text, we give some discussions on the proposed algorithm:

(a) It is desired that the wind direction does not change during the process of obtaining the optimal solution by the proposed algorithm. Otherwise, the solution may be not optimal for new wind direction. Note that the wind direction can change on the time scale of minutes. This implies that the presented algorithm should perform optimization on the order of seconds to adapt to the changes in real time—as we will see in later simulations, this is generally not a problem.

(b) Compared with the centralized method which may be appealing in terms of communication cost as each turbine only needs to communicate with wind farm controller, the proposed distributed method likely leads to high communication burden if each turbine communicates with its all coupling turbines. However, note that only weak wake interaction exists for two turbines with large spacing. Thus, it is reasonable for each turbine to establish communication only with its coupled

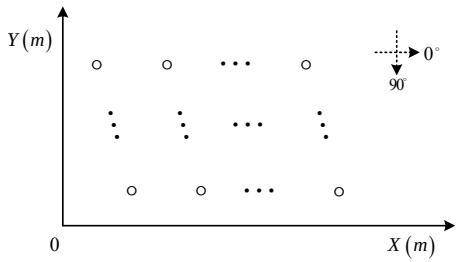


Fig. 5. Wind farm layout.

neighboring turbines. This can greatly reduce communication cost at the expense of minor loss of wind farm power gain than accurate distributed implementation.

In addition, the proposed distributed method also requires that each turbine can decide its own action instead of receiving the control action from wind farm controller (as in the centralized method). This requirement can be easily satisfied as the computing unit is relative cheap (compared to the cost of a turbine) and each turbine updates its action only by solving a low dimensional optimization problem.

(c) Note that the decentralized optimization methods have also potential to reduce computation burden due to their parallel computation. And the methods may require less communication than distributed optimization methods [10]. However, a single point of failure, e.g., in wind farm controller, can cause the whole wind farm to fail. The distributed optimization can usually eliminate single point of failure and thus shows higher reliability compared with decentralized optimization, which is attractive for large wind farms due to the high demand to reliability even if it implies more communication. To our knowledge, there is no model-based decentralized optimization method for the wind farm power optimization problem.

(d) Although this paper focuses on AIF-based control strategy to improve the power generation performance of wind farm, the proposed ADMM-based distributed power optimization method can be applied to yaw-angle-based wake redirection control without any difficulties, which is another strategy to increase the power output of wind farm.

V. Simulation Results

In this section, three simulation examples are given to illustrate the performance of the proposed method for wind farm power optimization. The greedy policy, where each turbine aims to maximize its own power output without considering the wake effect, is used as a baseline to show the algorithm's performance. The first example is performed in wind farms with different sizes to demonstrate the scalability of the proposed method. The second example verifies the performance of the proposed method for large wind farms in different wind directions. The third example tests the effect of communication delay and loss on the performance of the proposed method.

A. Wind Farms with Different Sizes

The wind farm layout shown in Fig. 5 is considered, where the spacing between adjacent turbine pair is $560m$. The power

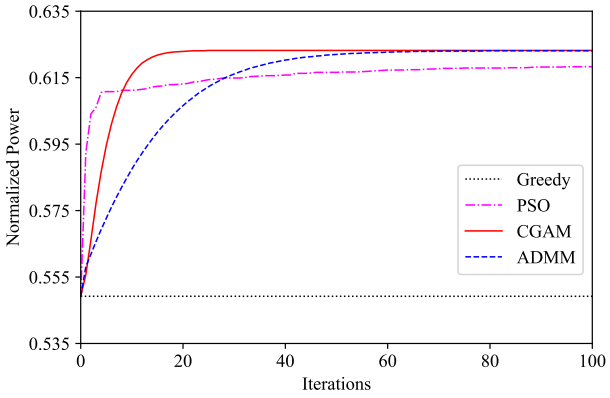


Fig. 6. Power trajectories of the wind farm under different algorithms.

TABLE I
Convergence Performance of the Different Algorithms

Method	Power gain	Iterations	Computation time
PSO	12.59%	95	1367.1s
CGAM	13.48%	25	580.1s
ADMM	13.46%	79	2.3s

generation model based on FLORIS model in [29] is used to verify the performance of the proposed distributed ADMM. The model parameters can be found in [29]. A common control constraint $\boldsymbol{\mathcal{U}}_i = \{u_i \,|0.1 \leq u_i \leq 0.33\}$ is applied to turbine $i \in \boldsymbol{\mathcal{N}}$. The upstream wind speed V_∞ is set as $8m/s$. The 40 degree wind direction is selected to demonstrate the performance of the proposed method. The penalty parameter ρ is set as 100. The $\boldsymbol{x}_i^{k+1}$ in Step 4 of Algorithm 1 is obtained by solving (20) using gradient method, where the gradient is estimated using central difference formula and whose iteration number is set as 25. The neighbor sets $\boldsymbol{\mathcal{N}}_i$ and $\bar{\boldsymbol{\mathcal{N}}}_i$ of turbine $i \in \boldsymbol{\mathcal{N}}$ are defined for given wind direction by considering all upstream turbines and downstream turbines of turbine i within a distance of 4 turbine spacing, respectively.

For comparison purpose, a centralized optimization method, the central-difference-formula-based gradient ascent method (CGAM), is selected. It is designed by combining the gradient ascent method and the estimated gradient using central difference formula. Note that the gradient ascent method is a very representative optimization method and generally has fast convergence speed. Meanwhile, the central difference formula is a popular approach to the approximation of gradient. Therefore, the CGAM is selected as control group. In addition, the particle swarm optimization (PSO) algorithm is also selected as control group due to its wide application in wind farm power optimization. All algorithms in this paper are implemented with a PC of Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 32.GB RAM, and NVIDIA GeForce RTX 2070. Note that the proposed distributed ADMM would be run in fully parallel manner when applied to real wind farm.

Firstly, a 8 by 10 wind farm with layout shown in Fig. 5 is considered, replicating the Horns Rev wind farm layout. Fig. 6 shows the normalized power trajectories of the wind

TABLE II
CONVERGENCE PERFORMANCE OF CGAM AND ADMM IN DIFFERENT WIND FARM SIZES

Wind farm size (Layout)	Power gain		Computation time	
	CGAM	ADMM	CGAM	ADMM
$36(6 \times 6)$	9.97%	9.96%	31.2s	1.8s
$64(8 \times 8)$	12.57%	12.56%	313.8s	2.4s
$100(10 \times 10)$	14.49%	14.47%	1379.7s	2.7s

TABLE III
COMPUTATION TIME OF CGAM AND ADMM IN DIFFERENT WIND DIRECTIONS

Method	Average	Minimum	Maximum
CGAM	454.6s	276.7s	732.5s
ADMM	3.1s	0.4s	10.7s

farm under different algorithms. Table. I gives the power gains achieved by centralized PSO, centralized CGAM, and proposed distributed ADMM compared with greedy policy, and the required iterations and computation time. From Fig. 6, it can be observed that the ADMM and CGAM significantly improve the power output of wind farm compared with greedy policy, which respectively achieve 13.46% and 13.48% power gains as shown Table. I. Hence, these two algorithms guarantee similar power generation performance. While the more iterations are required for ADMM to converge, the time that the ADMM takes in each iteration is much less than that the CGAM requires, which results in that the ADMM only takes 2.3s in optimization process but CGAM requires 580.1s. Note that the PSO algorithm takes 1367.1s in 95 iterations and generates lower power gain than another two methods. Therefore, it can be concluded that the proposed distributed ADMM significantly reduces the computation time without sacrificing power gain compared with the centralized CGAM and PSO algorithm.

Furtherly, three different-size wind farms are respectively constructed using the layout shown in Fig. 5, including 36-, 64- and 100-turbine wind farms. The convergence performance of different algorithms for the wind farms are shown in Table. II. From Table. II, the proposed distributed ADMM achieves similar power gain compared with centralized CGAM for different-size wind farms but significantly reduces computational time. Meanwhile, it can be also observed that the computation time of the proposed method has no significant increase with increasing wind farm size, whereby the time required by the centralized CGAM increases exponentially. In addition, note that the percentage of wind farm power gain increases with the increment of wind farm size as the wake couplings between turbines enhances. This implies that there is greater potential to improve the power efficiency for larger wind farms.

B. Wind Farm with Different Wind Directions

The 80-turbine wind farm that replicating the Horns Rev wind farm layout is considered again. All the parameters are same as the Section V-A except for the wind direction. The 36 wind directions from 0 to 360 degree at every 10 degrees are selected to verify the performance of the proposed method in different wind directions. The simulation results are given in Fig. 7, Fig. 8 and Table. III.

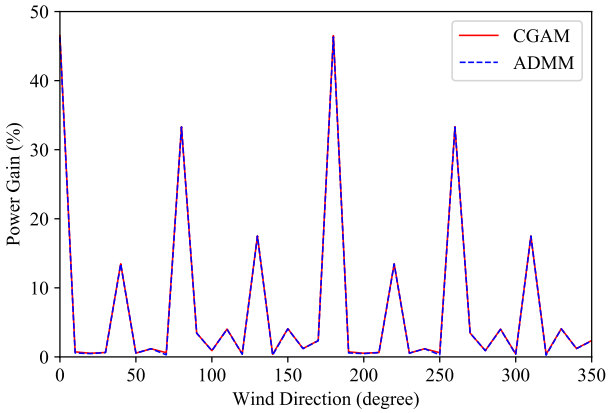


Fig. 7. Power gain of CGAM and ADMM compared with greedy policy in different wind directions.

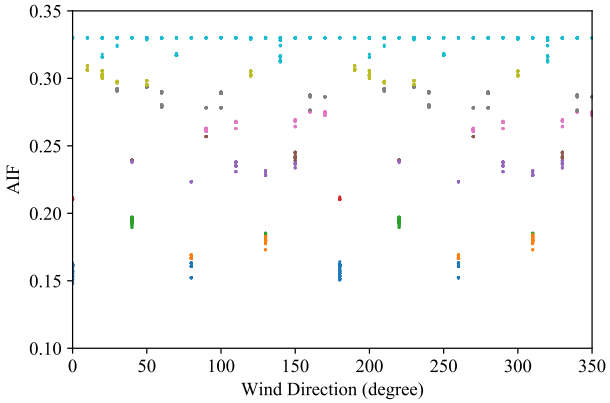


Fig. 8. Optimal actions generated by ADMM for different wind directions.

Fig. 7 indicates that the proposed distributed ADMM shows similar power gain with centralized CGAM in given wind directions. From Table. III, it can be observed that the average computation time and maximal computation time of distributed ADMM in given wind directions are 3.1s and 10.7s, respectively, which are far less than the average time on the order of minutes that wind conditions change and thus are suitable for real-time control of large wind farms to adapt to the changes of wind conditions and turbine configuration. Note that the average computation time of centralized CGAM is 454.6s and its maximal computation time is 732.5s. Therefore, the proposed ADMM significantly reduces the computation time in different wind directions compared with the centralized method as the whole wind farm power optimization problem is solved in a fully distributed manner. We also note that the computation time of the proposed method is similar to the results in [23], [25]. Different from [23], [25], the proposed method in this paper is developed considering control constraints, which can guarantee the optimal solution satisfy the control constraints of all turbines as can be clearly seen from

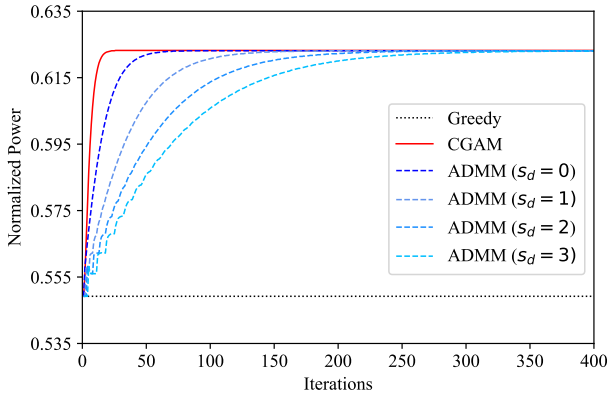


Fig. 9. Power trajectories of the wind farm under ADMM considering communication delay.

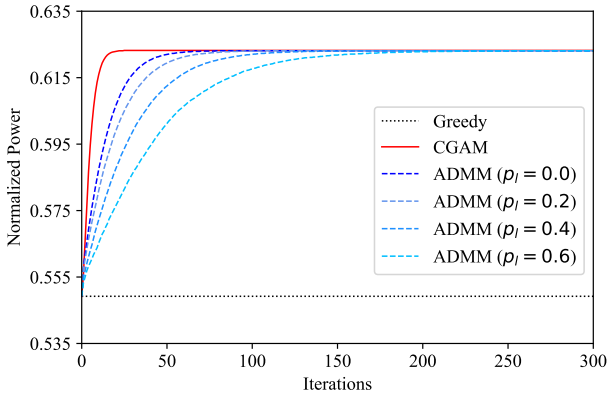


Fig. 10. Power trajectories of the wind farm under ADMM considering communication loss.

TABLE IV
CONVERGENCE SPEED OF ADMM CONSIDERING COMMUNICATION DELAY

s_d	0	1	2	3
Iterations	79	190	300	361
Computation time	2.3s	5.5s	8.7s	10.4s

Fig. 8. Meanwhile, the convergence property of the proposed method is detailedly analyzed and a better convergence result is obtained than [23], [25] as our design guarantees the convergence of the control actions. The above are desired for large wind farms to achieve real-time power optimization.

C. Effect of Communication Delay and Loss

The 80-turbine wind farm and the 40 degree wind direction are selected again to demonstrate the performance of proposed method considering communication uncertainties. The proposed method is firstly implemented considering different communication delays. Suppose that in each iteration, the turbine $i \in \boldsymbol{\mathcal{N}}$ receives the delayed information $\boldsymbol{x}_j^{k-s_d}$ and $\boldsymbol{\lambda}_j^{k-s_d}$ from the neighboring turbines $j \in \bar{\boldsymbol{\mathcal{N}}}_i$ and $u_j^{k+1-s_d}$ from the neighboring turbines $j \in \boldsymbol{\mathcal{N}}_i$, where s_d denotes the communication delay. For different values of s_d, the simulation results are given in Fig. 9 and Table. IV. Fig. 9 shows that the proposed distributed ADMM under different communication delays can still significantly improve the power output of wind farm compared with greedy policy, achieving similar power generation performance with centralized CGAM. To converge, the CGAM requires 25 iterations and totally takes 580.1s. The iterations and computation time required by proposed ADMM under different communication delays are given in Table. IV. Note that the proposed ADMM requires more iterations and computation time with increasing communication delay. Nevertheless, the proposed ADMM still remarkably reduces the computation time without sacrificing power gain compared with the centralized CGAM.

Next, the proposed ADMM is carried out considering communication loss. Assume that in each iteration, the turbine $i \in \boldsymbol{\mathcal{N}}$ can receive the information $\boldsymbol{x}_j^k$ and $\boldsymbol{\lambda}_j^k$ from the neighboring turbines $j \in \bar{\boldsymbol{\mathcal{N}}}_i$ with probability $1 - p_l$ and can not obtain the information with probability p_l due to the loss, where p_l is the communication loss rate. The setting is also applied when the turbine i receives u_j^{k+1} from the neighboring turbines $j \in \boldsymbol{\mathcal{N}}_i$. The turbine i would perform the iteration using the historical information without receiving new information. For different values of p_l, the simulation results are given in Fig. 10 and Table. V. Fig. 10 shows that the proposed distributed ADMM under different communication losses can still significantly increase the power output of wind farm than greedy policy, achieving similar power gain as centralized CGAM. From Table. V, it can be observed that the convergence speed of the proposed method decreases with increasing communication loss. Even so, the proposed ADMM can still achieve remarkable decrease in computation time compared with the centralized CGAM (580.1s).

Finally, the proposed ADMM is run for different wind

TABLE V
CONVERGENCE SPEED OF ADMM CONSIDERING COMMUNICATION LOSS

p_l	0%	20%	40%	60%
Iterations	79	107	159	249
Computation time	2.3s	3.1s	4.6s	7.2s

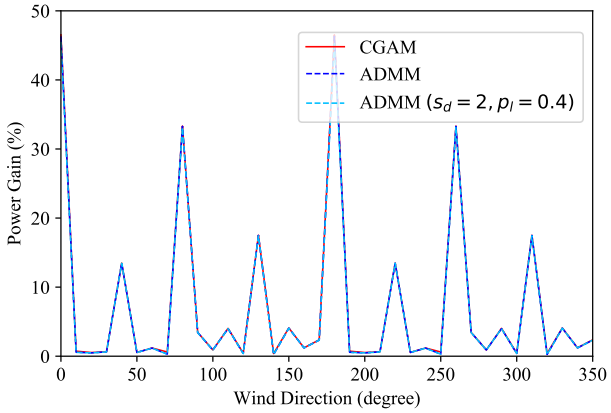


Fig. 11. Power gain of CGAM, ADMM and ADMM considering communication uncertainties compared with greedy policy in different wind directions.

TABLE VI
COMPUTATION TIME OF CGAM, ADMM AND ADMM CONSIDERING COMMUNICATION UNCERTAINTIES IN DIFFERENT WIND DIRECTIONS

Method	Average	Minimum	Maximum
CGAM	454.6s	276.7s	732.5s
ADMM	3.1s	0.4s	10.7s
ADMM ($s_d = 2$, $p_l = 40\%$)	13.7s	1.7s	44.1s

directions considering both communication delay and loss. The 36 wind directions from 0 to 360 degree at every 10 degrees are selected again. The communication delay $s_d = 2$ and communication loss rate $p_l = 40\%$ are used. The simulation results are given in Fig. 11 and Table. VI. Fig. 11 indicates that despite the existence of communication uncertainties (i.e. both delay and loss), the proposed ADMM still shows similar power gain with centralized CGAM. Table. VI shows that the convergence speed of distributed ADMM is adversely affected by the uncertainties. However, its average computation time is still far less than the average time that wind conditions change (in the order of minutes). Hence, the proposed ADMM is still effective for real-time control of large wind farms when there exist communication uncertainties.

VI. CONCLUSION

In this paper, a distributed power optimization method is proposed using ADMM for large wind farms that can be modelled to efficiently improve the power output in time-varying wind conditions and turbine configuration. The proposed method allows for turbines to be optimized in a fully distributed way by using local information, and thus has high computation efficiency, scalability and reliability. Meanwhile, the method is developed considering turbine control constraints, guaranteeing the implemented optimal action feasible for real turbines. Additionally, the convergence analysis of the proposed method is given under moderate assumptions and the result shows that it can converge to a stationary point of the optimisation problem. Three simulation examples are presented to demonstrate the effectiveness of the proposed method. The first example shows that the proposed method can significantly reduce the computation time in wind farms with different sizes compared with centralized method, and its computational time has no obvious increase with the increasing wind farm size. The second example indicates that the presented method can obviously improve the power generation performance of large wind farms under different wind conditions and its computation time is much lower than that of the centralized method. The third example verifies that the proposed method is still suitable for real-time control of large wind farms on the effect of communication delay and loss. These results show that the proposed algorithm provides an efficient design for large wind farms to achieve real-time power optimization to adapt to time-varying wind and turbine conditions.

Note that the effect of communication delay and loss on the proposed method is tested by simulation in Section V. Its rigorous theoretical analysis will be our future research. While the obtained results are promising, in this paper we have not considered the effect of power generation model uncertainties on wind farm power generation performance. An accurate power generation model may not be obtained easily for the wind farm sited in complex terrain due to complexities of wake interactions. How to use available analytical model and real-time power generation data to develop distributed power optimization method for large wind farms is a key area of our future research. Finally, our future research also includes performing experimental tests on actual wind farms.

ACKNOWLEDGEMENT

The authors would like to thank the editor and anonymous reviewers for their insightful comments and suggestions.

APPENDIX A
PROOF OF THEOREM 1

Before proving Theorem 1, we first prove the following Lemmas, which are key properties to guarantee the convergence performance of distributed ADMM for (14).

Lemma 1: Suppose Assumption 1, 2 and 3 hold. We have the following:

(a) The augmented Lagrangian function value sequence $\{\mathcal{L}_\rho(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k)\}$ is monotonically decreasing:

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k)\\ &\leq \sum\nolimits_{i=1}^{n} (\frac{L^2}{\rho} - \frac{\gamma_i}{2})\|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k\|^2 \leq 0;\end{aligned} \tag{30}$$

(b) The augmented Lagrangian function value sequence $\{\mathcal{L}_\rho(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k)\}$ is lower bounded:

$$\mathcal{L}_\rho(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k) \geq f^* > -\infty. \tag{31}$$

Proof: (a) The successive difference of the augmented Lagrangian function value is splitted into two terms as follows:

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k)\\ &= \left[\mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^k)\right]\\ &\quad + \left[\mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^k) - \mathcal{L}_\rho(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k)\right].\end{aligned} \tag{32}$$

Consider first term in (32). According to (15), (16) and (21), we have

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^k)\\ &= \sum\nolimits_{i=1}^{n} \left(\mathcal{L}_i(\boldsymbol{x}_i^{k+1}, \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^{k+1}) - \mathcal{L}_i(\boldsymbol{x}_i^{k+1}, \tilde{\boldsymbol{u}}_i^{k+1}, \boldsymbol{\lambda}_i^k)\right)\\ &= \sum\nolimits_{i=1}^{n} (\boldsymbol{\lambda}_i^{k+1} - \boldsymbol{\lambda}_i^k)^T (\boldsymbol{x}_i^{k+1} - \tilde{\boldsymbol{u}}_i^{k+1})\\ &= \sum\nolimits_{i=1}^{n} \frac{1}{\rho}(\boldsymbol{\lambda}_i^{k+1} - \boldsymbol{\lambda}_i^k)^T (\boldsymbol{\lambda}_i^{k+1} - \boldsymbol{\lambda}_i^k).\end{aligned} \tag{33}$$

Then

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\lambda}^k)\\ &= \sum\nolimits_{i=1}^{n} \frac{1}{\rho}\|\boldsymbol{\lambda}_i^{k+1} - \boldsymbol{\lambda}_i^k\|^2.\end{aligned} \tag{34}$$

Using (16), we obtain the optimality condition of subproblem (20) with respect to $\boldsymbol{x}_i$:

$$\nabla f_i(\boldsymbol{x}_i^{k+1}) + \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{x}_i^{k+1} - \tilde{\boldsymbol{u}}_i^{k+1}) = 0. \tag{35}$$

Combining (21) and (35), we have

$$\nabla f_i(\boldsymbol{x}_i^{k+1}) = -\boldsymbol{\lambda}_i^{k+1}. \tag{36}$$

Substituting (36) into (34) and using Assumption 1, we can further get

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k})\\ \le&\sum\nolimits_{i=1}^{n}\frac{L^2}{\rho}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2.\end{aligned} \tag{37}$$

Consider the second term in (32). Using (15), we obtain

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\\ =&\left[\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k})\right]\\ &+\left[\mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\right]\\ =&\left[\sum\nolimits_{i=1}^{n}\left(\mathcal{L}_i(\boldsymbol{x}_i^{k+1},\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k}) - \mathcal{L}_i(\boldsymbol{x}_i^{k},\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k})\right)\right]\\ &+\left[\mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\right].\end{aligned} \tag{38}$$

Since $\boldsymbol{u}^{k+1}$ is the optimal solution of the $\mathcal{L}_\rho\left(\boldsymbol{x}^k,\boldsymbol{u},\boldsymbol{\lambda}^k\right)$ with respect to $\boldsymbol{u}$ in (17), we have

$$\mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k}) \le 0. \tag{39}$$

Then (38) can be rewritten as

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\\ \le&\sum\nolimits_{i=1}^{n}\left(\mathcal{L}_i(\boldsymbol{x}_i^{k+1},\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k}) - \mathcal{L}_i(\boldsymbol{x}_i^{k},\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k})\right).\end{aligned} \tag{40}$$

Combining (24) and (40) yields

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\\ \le&\sum\nolimits_{i=1}^{n}\nabla_{\boldsymbol{x_i}}\mathcal{L}_i(\boldsymbol{x}_i^{k+1},\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k})^T(\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k})\\ &-\sum\nolimits_{i=1}^{n}\frac{\gamma_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2.\end{aligned} \tag{41}$$

Since $\boldsymbol{x}_i^{k+1}$ is the optimal solution of the $\mathcal{L}_i(\boldsymbol{x}_i,\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k})$ with respect to $\boldsymbol{x}_i$ in (20), we get

$$\nabla_{\boldsymbol{x}_i}\mathcal{L}_i(\boldsymbol{x}_i^{k+1},\tilde{\boldsymbol{u}}_i^{k+1},\boldsymbol{\lambda}_i^{k})=0.$$

Then (41) can be rewritten as

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\\ \le&-\sum\nolimits_{i=1}^{n}\frac{\gamma_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2.\end{aligned} \tag{42}$$

According to (32), (37), and (42), we obtain

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\\ \le&\sum\nolimits_{i=1}^{n}\frac{L^2}{\rho}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2-\sum\nolimits_{i=1}^{n}\frac{\gamma_i}{2}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2\\ =&\sum\nolimits_{i=1}^{n}(\frac{L^2}{\rho}-\frac{\gamma_i}{2})\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2.\end{aligned} \tag{43}$$

From (43) and $\rho\gamma_i \ge 2L^2$ (Assumption 2), we get that

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k})\\ \le&\sum\nolimits_{i=1}^{n}(\frac{L^2}{\rho}-\frac{\gamma_i}{2})\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2\le 0.\end{aligned}$$

Namely (30) holds.

(b) From (15), (16) and (36), we derive

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1})\\ =&\sum\nolimits_{i=1}^{n}\left(f_i(\boldsymbol{x}_i^{k+1})+(\boldsymbol{\lambda}_i^{k+1})^T(\boldsymbol{x}_i^{k+1}-\tilde{\boldsymbol{u}}_i^{k+1})\right)\\ &+\frac{\rho}{2}\sum\nolimits_{i=1}^{n}\|\boldsymbol{x}_i^{k+1}-\tilde{\boldsymbol{u}}_i^{k+1}\|^2+h(\boldsymbol{u}^{k+1})\\ =&\sum\nolimits_{i=1}^{n}\left(f_i(\boldsymbol{x}_i^{k+1})+\nabla f_i(\boldsymbol{x}_i^{k+1})^T(\tilde{\boldsymbol{u}}_i^{k+1}-\boldsymbol{x}_i^{k+1})\right)\\ &+\frac{\rho}{2}\sum\nolimits_{i=1}^{n}\|\boldsymbol{x}_i^{k+1}-\tilde{\boldsymbol{u}}_i^{k+1}\|^2+h(\boldsymbol{u}^{k+1}).\end{aligned} \tag{44}$$

As ∇f_i is Lipschitz continuous with positive constant $L>0$ (Assumption 1), we have

$$\begin{aligned}f_i(\tilde{\boldsymbol{u}}_i^{k+1}) \le& f_i(\boldsymbol{x}_i^{k+1})+\nabla f_i(\boldsymbol{x}_i^{k+1})^T(\tilde{\boldsymbol{u}}_i^{k+1}-\boldsymbol{x}_i^{k+1})\\ &+(L/2)\|\tilde{\boldsymbol{u}}_i^{k+1}-\boldsymbol{x}_i^{k+1}\|^2.\end{aligned} \tag{45}$$

Using $\rho \ge L$ (Assumption 2), (44) and (45), we obtain

$$\mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1}) \ge \sum\nolimits_{i=1}^{n}f_i(\tilde{\boldsymbol{u}}_i^{k+1})+h(\boldsymbol{u}^{k+1}).$$

According to Assumption 3, the (31) holds. ■

Lemma 1 implies that if the assumptions are true, the augmented Lagrangian function value sequence $\{\mathcal{L}_\rho(\boldsymbol{x}^k,\boldsymbol{u}^k,\boldsymbol{\lambda}^k)\}$ converges using monotone bounded theorem. Next, we prove another important Lemma.

Lemma 2: Suppose Assumption 1, 2 and 3 hold. Let $(\{\boldsymbol{x}_i^*\},\boldsymbol{u}^*,\{\boldsymbol{\lambda}_i^*\})$ be any limit point of the iteration sequence $\left\{\{\boldsymbol{x}_i^k\},\boldsymbol{u}^k,\{\boldsymbol{\lambda}_i^k\}\right\}$ generated by distributed ADMM. Then

$$\boldsymbol{u}^* \in \underset{\boldsymbol{u}\in\boldsymbol{\mathcal{U}}}{\operatorname{argmin}}\, h(\boldsymbol{u})+\sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^*,\boldsymbol{x}_i^*-\tilde{\boldsymbol{u}}_i\right\rangle, \tag{46}$$

$$\nabla f_i\left(\boldsymbol{x}_i^*\right)+\boldsymbol{\lambda}_i^*=0, \tag{47}$$

$$\boldsymbol{x}_i^*=\tilde{\boldsymbol{u}}_i^*,\ i\in\boldsymbol{\mathcal{N}}, \tag{48}$$

which show that any limit point of distributed ADMM is a stationary point of problem (14).

Proof: According to (30), we obtain

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{0},\boldsymbol{u}^{0},\boldsymbol{\lambda}^{0}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1})\\ =&\left[\mathcal{L}_\rho(\boldsymbol{x}^{0},\boldsymbol{u}^{0},\boldsymbol{\lambda}^{0}) - \mathcal{L}_\rho(\boldsymbol{x}^{1},\boldsymbol{u}^{1},\boldsymbol{\lambda}^{1})\right]\\ &+\left[\mathcal{L}_\rho(\boldsymbol{x}^{1},\boldsymbol{u}^{1},\boldsymbol{\lambda}^{1}) - \mathcal{L}_\rho(\boldsymbol{x}^{2},\boldsymbol{u}^{2},\boldsymbol{\lambda}^{2})\right]\\ &+\cdots+\left[\mathcal{L}_\rho(\boldsymbol{x}^{k},\boldsymbol{u}^{k},\boldsymbol{\lambda}^{k}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1})\right]\\ \ge&\sum\nolimits_{i=1}^{n}(\frac{\gamma_i}{2}-\frac{L^2}{\rho})\|\boldsymbol{x}_i^{1}-\boldsymbol{x}_i^{0}\|^2\\ &+\sum\nolimits_{i=1}^{n}(\frac{\gamma_i}{2}-\frac{L^2}{\rho})\|\boldsymbol{x}_i^{2}-\boldsymbol{x}_i^{1}\|^2\\ &+\cdots+\sum\nolimits_{i=1}^{n}(\frac{\gamma_i}{2}-\frac{L^2}{\rho})\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|^2\\ \ge&\sum\nolimits_{j=0}^{k}\sum\nolimits_{i=1}^{n}(\frac{\gamma_i}{2}-\frac{L^2}{\rho})\|\boldsymbol{x}_i^{j+1}-\boldsymbol{x}_i^{j}\|^2.\end{aligned} \tag{49}$$

Using (31) yields

$$\begin{aligned}&\mathcal{L}_\rho(\boldsymbol{x}^{0},\boldsymbol{u}^{0},\boldsymbol{\lambda}^{0}) - \mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1})\\ =&\left[\mathcal{L}_\rho(\boldsymbol{x}^{0},\boldsymbol{u}^{0},\boldsymbol{\lambda}^{0}) - f^*\right]+\left[f^* - \mathcal{L}_\rho(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\lambda}^{k+1})\right]\\ \le&\mathcal{L}_\rho(\boldsymbol{x}^{0},\boldsymbol{u}^{0},\boldsymbol{\lambda}^{0}) - f^* < \infty.\end{aligned} \tag{50}$$

From (49) and (50), we get

$$\sum\nolimits_{j=0}^{k}\sum\nolimits_{i=1}^{n}(\frac{\gamma_i}{2}-\frac{L^2}{\rho})\|\boldsymbol{x}_i^{j+1}-\boldsymbol{x}_i^{j}\|^2<\infty. \tag{51}$$

As $\rho\gamma_i \geq 2L^2$ (Assumption 2) for all i, we have

$$\lim_{k\to\infty}\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|=0. \tag{52}$$

According to (36) and Assumption 1, we derive

$$\begin{aligned}\|\boldsymbol{\lambda}_i^{k+1}-\boldsymbol{\lambda}_i^{k}\| =&\|\nabla f_i(\boldsymbol{x}_i^{k+1})-\nabla f_i(\boldsymbol{x}_i^{k})\|\\ \leq& L\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}_i^{k}\|.\end{aligned} \tag{53}$$

Based on (52) and (53) , we get

$$\lim_{k\to\infty}\|\boldsymbol{\lambda}_i^{k+1}-\boldsymbol{\lambda}_i^{k}\|=0. \tag{54}$$

Further using (21) and (54), we obtain

$$\lim_{k\to\infty}\|\boldsymbol{x}_i^{k+1}-\tilde{\boldsymbol{u}}_i^{k+1}\|=0, \tag{55}$$

which means for any i,

$$\boldsymbol{x}_i^{*}=\tilde{\boldsymbol{u}}_i^{*}. \tag{56}$$

Namely (48) holds.

Taking limit on both sides of (36), we have

$$\nabla f_i(\boldsymbol{x}_i^{*})+\boldsymbol{\lambda}_i^{*}=0.$$

Namely (47) holds.

Since $\boldsymbol{u}^{k+1}$ is the optimal solution of the $\mathcal{L}_\rho(\boldsymbol{x}^k,\boldsymbol{u},\boldsymbol{\lambda}^k)$ with $\boldsymbol{u}\in\boldsymbol{\mathcal{U}}$ in (17), there is a $\boldsymbol{g}^{k+1}\in\partial h(\boldsymbol{u}^{k+1})$ such that

$$\begin{aligned}&-\sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^{k}+\rho(\boldsymbol{x}_i^{k}-\tilde{\boldsymbol{u}}_i^{k+1}),\tilde{\boldsymbol{u}}_i-\tilde{\boldsymbol{u}}_i^{k+1}\right\rangle\\ &+\left\langle\boldsymbol{g}^{k+1},\boldsymbol{u}-\boldsymbol{u}^{k+1}\right\rangle\geq 0,\quad\forall\boldsymbol{u}\in\boldsymbol{\mathcal{U}}.\end{aligned} \tag{57}$$

Then

$$\begin{aligned}&-\sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^{k}+\rho(\boldsymbol{x}_i^{k}-\tilde{\boldsymbol{u}}_i^{k+1}),\tilde{\boldsymbol{u}}_i-\tilde{\boldsymbol{u}}_i^{k+1}\right\rangle\\ &+h(\boldsymbol{u})-h(\boldsymbol{u}^{k+1})\geq 0,\quad\forall\boldsymbol{u}\in\boldsymbol{\mathcal{U}}.\end{aligned} \tag{58}$$

Taking limit on both sides of (58) and applying (56) yields

$$\begin{aligned}&h(\boldsymbol{u})+\sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^{*},\boldsymbol{x}_i^{*}-\tilde{\boldsymbol{u}}_i\right\rangle\\ \geq& h(\boldsymbol{u}^{*})+\sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^{*},\boldsymbol{x}_i^{*}-\tilde{\boldsymbol{u}}_i^{*}\right\rangle,\quad\forall\boldsymbol{u}\in\boldsymbol{\mathcal{U}}.\end{aligned} \tag{59}$$

This implies

$$\boldsymbol{u}^{*}\in\operatorname*{argmin}_{\boldsymbol{u}\in\boldsymbol{\mathcal{U}}}h(\boldsymbol{u})+\sum\nolimits_{i=1}^{n}\left\langle\boldsymbol{\lambda}_i^{*},\boldsymbol{x}_i^{*}-\tilde{\boldsymbol{u}}_i\right\rangle.$$

Namely (46) holds. ■

Finally, we perform the proof of Theorem 1 using above results in following contents.

Proof: We first prove that the sequences $\{\boldsymbol{u}^k\}$, $\{\boldsymbol{x}_i^k\}$ and $\{\boldsymbol{\lambda}_i^k\}$ lie in some compact sets. Obviously, the admissible set $\boldsymbol{\mathcal{U}}$ is a compact set. The (17) shows the sequence $\{\boldsymbol{u}^k\}$ belongs to the compact set $\boldsymbol{\mathcal{U}}$. This also means that $\tilde{\boldsymbol{u}}_i^k$ lies in a compact set. Then from (55), we can get that $\{\boldsymbol{x}_i^k\}$ belongs to a compact set. The Lipschitz continuity of ∇f_i and boundedness of $\{\boldsymbol{x}_i^k\}$ show that $\{\nabla f_i(\boldsymbol{x}_i^k)\}$ is a bounded sequence. Therefore using (36), we can obtain that $\{\boldsymbol{\lambda}_i^k\}$ is a bounded sequence and thus must lie in a compact set.

Then we prove main result (25) by contradiction. Suppose that

$$\lim_{k\to\infty}\operatorname{dist}\left((\{\boldsymbol{x}_i^k\},\boldsymbol{u}^k,\{\boldsymbol{\lambda}_i^k\}),\boldsymbol{Z}^*\right)\neq 0. \tag{60}$$

Hence there exists a $\varepsilon>0$ such that for any positive integer N, when $N=1$, there is a positive integer $k_0>N$ satisfying

$$\operatorname{dist}\left((\{\boldsymbol{x}_i^{k_0}\},\boldsymbol{u}^{k_0},\{\boldsymbol{\lambda}_i^{k_0}\}),\boldsymbol{Z}^*\right)>\varepsilon; \tag{61}$$

When $N=k_0$, there is a positive integer $k_1>N$ satisfying

$$\operatorname{dist}\left((\{\boldsymbol{x}_i^{k_1}\},\boldsymbol{u}^{k_1},\{\boldsymbol{\lambda}_i^{k_1}\}),\boldsymbol{Z}^*\right)>\varepsilon; \tag{62}$$

$$\cdots$$

When $N=k_{j-1}$, there is a positive integer $k_j>N$ satisfying

$$\operatorname{dist}\left((\{\boldsymbol{x}_i^{k_j}\},\boldsymbol{u}^{k_j},\{\boldsymbol{\lambda}_i^{k_j}\}),\boldsymbol{Z}^*\right)>\varepsilon; \tag{63}$$

$$\cdots$$

Then we can obtain a subsequence $\{\{\boldsymbol{x}_i^{k_j}\},\boldsymbol{u}^{k_j},\{\boldsymbol{\lambda}_i^{k_j}\}\}$ of the sequence $\left\{\left\{\boldsymbol{x}_i^{k}\right\},\boldsymbol{u}^{k},\left\{\boldsymbol{\lambda}_i^{k}\right\}\right\}$ satisfying for any a $\{\boldsymbol{x}_i^{k_j}\},\boldsymbol{u}^{k_j},\{\boldsymbol{\lambda}_i^{k_j}\}$,

$$\operatorname{dist}\left((\{\boldsymbol{x}_i^{k_j}\},\boldsymbol{u}^{k_j},\{\boldsymbol{\lambda}_i^{k_j}\}),\boldsymbol{Z}^*\right)>\varepsilon>0. \tag{64}$$

From the previous argument, we can easily obtain that the subsequence $\{\{\boldsymbol{x}_i^{k_j}\},\boldsymbol{u}^{k_j},\{\boldsymbol{\lambda}_i^{k_j}\}\}$ also lie in some compact sets. Then there exists a convergent subsequence $\{\{\boldsymbol{x}_i^{k_{j_l}}\},\boldsymbol{u}^{k_{j_l}},\{\boldsymbol{\lambda}_i^{k_{j_l}}\}\}$ for subsequence $\{\{\boldsymbol{x}_i^{k_j}\},\boldsymbol{u}^{k_j},\{\boldsymbol{\lambda}_i^{k_j}\}\}$ such that

$$(\{\boldsymbol{x}_i^{k_{j_l}}\},\boldsymbol{u}^{k_{j_l}},\{\boldsymbol{\lambda}_i^{k_{j_l}}\})\to(\{\hat{\boldsymbol{x}}_i\},\hat{\boldsymbol{u}},\{\hat{\boldsymbol{\lambda}}_i\})(l\to\infty), \tag{65}$$

where $(\{\hat{\boldsymbol{x}}_i\},\hat{\boldsymbol{u}},\{\hat{\boldsymbol{\lambda}}_i\})$ is limit point. By (65), there is $\bar{L}>0$ such that for any $l\geq\bar{L}$,

$$\left\|(\{\boldsymbol{x}_i^{k_{j_l}}\},\boldsymbol{u}^{k_{j_l}},\{\boldsymbol{\lambda}_i^{k_{j_l}}\})-(\{\hat{\boldsymbol{x}}_i\},\hat{\boldsymbol{u}},\{\hat{\boldsymbol{\lambda}}_i\})\right\|\leq\varepsilon/2. \tag{66}$$

Using Lemma 2 yields $(\{\hat{\boldsymbol{x}}_i\},\hat{\boldsymbol{u}},\{\hat{\boldsymbol{\lambda}}_i\})\in\boldsymbol{Z}^*$. Then we have

$$\begin{aligned}&\operatorname{dist}\left((\{\boldsymbol{x}_i^{k_{j_l}}\},\boldsymbol{u}^{k_{j_l}},\{\boldsymbol{\lambda}_i^{k_{j_l}}\}),\boldsymbol{Z}^*\right)\\ \leq&\operatorname{dist}\left((\{\boldsymbol{x}_i^{k_{j_l}}\},\boldsymbol{u}^{k_{j_l}},\{\boldsymbol{\lambda}_i^{k_{j_l}}\}),(\hat{\boldsymbol{u}},\{\hat{\boldsymbol{x}}_i\},\{\hat{\boldsymbol{\lambda}}_i\})\right)\\ \leq&\varepsilon/2.\end{aligned} \tag{67}$$

This contradicts to (64). Therefore (25) holds. ■

REFERENCES

[1] L. Ye, C. Zhang, Y. Tang, W. Zhong, Y. Zhao, P. Lu, B. Zhai, H. Lan, and Z. Li, "Hierarchical model predictive control strategy based on dynamic active power dispatch for wind power cluster integration," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4617–4629, May. 2019.

[2] H. Zhao, J. Zhao, J. Qiu, G. Liang, and Z. Y. Dong, "Cooperative wind farm control with deep reinforcement learning and knowledge-assisted learning," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6912–6921, Nov. 2020.

[3] W. Li, M. Zhu, P. Chao, X. Liang, and D. Xu, "Enhanced FRT and postfault recovery control for MMC-HVDC connected offshore wind farms," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1606–1617, Mar. 2020.

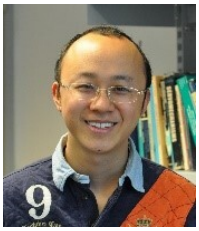
[4] G. W. E. Council, "Global wind report 2021," https://gwec.net/global-wind-report-2021/, accessed May 31, 2021.

[5] U. Ciri, M. A. Rotea, and S. Leonardi, "Model-free control of wind farms: A comparative study between individual and coordinated extremum seeking," *Renew Energ*, vol. 113, pp. 1033–1045, Jun. 2017.

[6] H. Liao, W. Hu, X. Wu, N. Wang, Z. Liu, Q. Huang, C. Chen, and Z. Chen, "Active power dispatch optimization for offshore wind farms considering fatigue distribution," *Renew Energ*, vol. 151, pp. 1173–1185, Nov. 2020.
[7] M. Vali, V. Petrović, L. Y. Pao, and M. Kühn, "Model predictive active power control for optimal structural load equalization in waked wind farms," *IEEE Trans. Control Syst. Technol.*, pp. 1–15, 2021, doi: 10.1109/TCST.2021.3053776.
[8] J. R. Marden, S. D. Ruben, and L. Y. Pao, "A model-free approach to wind farm control using game theoretic methods," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1207–1214, Jul. 2013.
[9] S. Siniscalchi-Minna, F. D. Bianchi, C. Ocampo-Martinez, J. L. Domínguez-García, and B. D. Schutter, "A non-centralized predictive control strategy for wind farm active power control: A wake-based partitioning approach," *Renew Energ*, vol. 150, pp. 656–669, Jan. 2020.
[10] X. Fan, E. Crisostomi, D. Thomopulos, B. Zhang, R. Shorten, and S. Yang, "An optimized decentralized power sharing strategy for wind farm de-loading," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 136–146, Jan. 2021.
[11] Z. Wang and W. Wu, "Coordinated control method for DFIG-based wind farm to provide primary frequency regulation service," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2644–2659, May. 2018.
[12] T. Ahmad, O. Coupiac, A. Petit, S. Guignard, N. Girard, B. Kazemtabrizi, and P. C. Matthews, "Field implementation and trial of coordinated control of wind farms," *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1169–1176, Jul. 2018.
[13] J. A. Dahlberg, "Assessment of the Lillgrund windfarm: Power performance, wake effects," Vatenfall Vindkraft AB, Sweden, 6_1 LG Pilot Report, Tech. Rep., Sep. 2009.
[14] N. Deljouyi, A. Nobakhti, and A. Abdolahi, "Wind farm power output optimization using cooperative control methods," *Wind Energy*, vol. 24, pp. 502–514, May. 2021.
[15] D. van der Hoek, S. Kanev, J. Allin, D. Bieniek, and N. Mittelmeier, "Effects of axial induction control on wind farm energy production-A field test," *Renew Energ*, vol. 140, pp. 994–1003, Mar. 2019.
[16] Y. Tang, Z. Ren, and N. Li, "Zeroth-order feedback optimization for cooperative multi-agent systems," in *59th IEEE Conf. Decis. Control*, Dec. 2020, pp. 3649–3656.
[17] J.-M. Xu and Y. C. Soh, "A distributed simultaneous perturbation approach for large-scale dynamic optimization problems," *Automatica*, vol. 72, pp. 194–204, Jul. 2016.
[18] S. Zhong and X. Wang, "Decentralized model-free wind farm control via discrete adaptive filtering methods," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2529–2540, Jul. 2018.
[19] Z. Xu, H. Geng, and B. Chu, "A hierarchical data-driven wind farm power optimization approach using stochastic projected simplex method," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3560–3569, Jul. 2021.
[20] J. Park and K. H. Law, "Cooperative wind turbine control for maximizing wind farm power using sequential convex programming," *Energy Convers. Manage.*, vol. 101, pp. 295–316, Jun. 2015.
[21] A. Behnood, H. Gharavi, B. Vahidi, and G. Riahy, "Optimal output power of not properly designed wind farms, considering wake effects," *JEPE*, vol. 63, pp. 44–50, Jun. 2014.
[22] B. Gu, Y. Liu, J. Yan, L. Li, and S. Kang, "A wind farm optimal control algorithm based on wake fast-calculation model," *J. Sol. Energy Eng.*, vol. 138, pp. 1–5, Apr. 2016.
[23] J. Annoni, C. Bay, T. Taylor, L. Pao, P. Fleming, and K. Johnson, "Efficient optimization of large wind farms for real-time control," in *Proc. Annu. Amer. Control Conf.*, Jun. 2018, pp. 6200–6205.
[24] P. Graf, J. Annoni, C. Bay, D. Biagioni, D. Sigler, M. Lunacek, and W. Jones, "Distributed reinforcement learning with ADMM-RL," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 4159–4166.
[25] J. Annoni, E. Dall'Anese, M. Hong, and C. J. Bay, "Efficient distributed optimization of wind farms using proximal primal-dual algorithms," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 4173–4178.
[26] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, Jan. 2016.
[27] X. Wang, J. Yan, B. Jin, and W. Li, "Distributed and parallel ADMM for structured nonconvex optimization problem," *IEEE Trans. Cybern.*, pp. 1–13, Dec. 2019, doi: 10.1109/TCYB.2019.2950337.
[28] S. Boersma, B. M. Doekemeijer, P. Gebraad, P. A. Fleming, and J. Wingerden, "A tutorial on control-oriented modeling and control of wind farms," in *Proc. Amer. Control Conf.*, May. 2017, pp. 1–18.
[29] P. M. O. Gebraad, F. W. Teeuwisse1, J.-W. van Wingerden, P. A. Fleming, S. D. Ruben, J. R. Marden, and L. Y. Pao, "Wind plant power optimization through yaw control using a parametric model for wake effects-a CFD simulation study," *Wind Energy*, vol. 19, pp. 95–114, Dec. 2016.

Zhiwei Xu (Graduate Student Member, IEEE) received the B.S. degree from the College of Mathematics and Statistics, Northwest Normal University, Lanzhou, China, in 2015, and the M.S. degree from the School of Information Science and Engineering, Central South University, Changsha, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China.

His current research interests include online learning, data-driven optimization, model-based learning control, reinforcement learning, distributed optimization, and their applications to cooperative wind farm control.



Bing Chu received the B.Eng. degree in automation and the M.Sc. degree in control science and technology from Tsinghua University, Beijing, China, in 2004 and 2007, respectively, and the Ph.D. degree in automatic control and systems engineering from the University of Sheffield, Sheffield, U.K. in 2009. He was a Post-Doctoral Researcher with the University of Oxford, Oxford, U.K., from 2010 to 2012. He is currently an Associate Professor in Electronics and Computer Science with the University of Southampton, Southampton, U.K.

His current research interests include iterative learning and repetitive control, analysis and control of large-scale networked systems, applied optimization theory, and their applications to robotics, energy and sustainability, and next generation healthcare. Dr. Chu was a recipient of a number of awards including the Best Paper Prize of the 2012 United Kingdom Automatic Control Council International Conference on Control and Certificate of Merit for the 2010 IET Control and Automation Doctoral Dissertation Prize.

Hua Geng (S'07-M'10-SM'14-F'21) received the B.S. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2003 and the Ph.D. degree in control theory and application from Tsinghua University, Beijing, China, in 2008. From 2008 to 2010, he was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada. He joined Automation Department of Tsinghua University in June 2010.

His current research interests include advanced control on power electronics and renewable energy conversion systems. He has authored more than 170 technical publications and holds more than 20 issued Chinese/US patents. He is the editors of IEEE Trans. on Energy Conversion and IEEE Trans. on Sustainable Energy, associate editors of IEEE Trans. on Industry Applications, IET Renewable Power Generation, Control Engineering Practice. He is an IEEE Fellow and an IET Fellow, Standing Director of China Power Supply Society (CPSS), vice chair of IEEE IAS Beijing Chapter.

Xiaohong Nian received the B.S. degree from Northwest Normal University, Lanzhou, China, in 1985, the M.S. degree from Shandong University, Jinan, China, in 1996, and the Ph.D. degree from Peking University, Beijing, China, in 1999. From 2004 to 2008, he was a Research Fellow with the Institute of Zhuzhou Electric Locomotive, Zhuzhou, China. He is currently a Professor with Central South University, Changsha, China.

His current research interests include coordinated control and optimization of complicated multiagent systems, unmanned aerial vehicles, converter technology, and control of tractive power supply systems in highspeed train.