Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Model-Free Optimization Scheme for Efficiency Improvement of Wind Farm Using Decentralized Reinforcement Learning ⋆

Zhiwei Xu * **Hua Geng** * **Bing Chu** ** **Menghao Qian** *** **Ni Tan** ***

* *Department of Automation, Tsinghua University and Beijing National Research Center for Information Science and Technology, Beijing 100190, China (e-mail: xuzw18@mails.tsinghua.edu.cn, genghua@tsinghua.edu.cn).*
** *School of Electronics and Computer Science, Faculty of Physical Sciences and Engineering, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: B.Chu@soton.ac.uk)*
*** *Department of Electrical Engineering, Tsinghua University, Beijing 100190, China (e-mail: {qmh17, tn17}@mails.tsinghua.edu.cn)*

Abstract: Wake interactions caused by the complex wakes between the turbines within a wind farm have significant adverse effect on the total power generation of the wind farm. To mitigate the effect of wake interactions and optimize the total power output of wind farm, this paper proposes a model-free control scheme using reinforcement learning by developing a decentralized Q learning method. The proposed approach guarantees that the output power of wind farm converges to the optimal total power under different wind conditions, and further ensures the gradual changes of control variables of wind turbines and thus avoids the unexpected sharp drop of the power generation performance of wind farm. Simulation results are provided to demonstrate the effectiveness of the proposed method.

Keywords: Wind farm, power optimization, model-free approach, decentralized control, Q learning method.

1. INTRODUCTION

In wind farm, the wakes generated by upstream wind turbines can significantly degrade the output power of the downstream wind turbines due to reduced wind speed inside the wakes (Park and Law (2016)). The greedy policy which is widely used in practice, where each turbine works only to maximize its own individual output power (Gebraad et al. (2016)), may not be able to produce the maximum output power of the wind farm. It is a locally optimal control strategy due to the neglect of the wake interactions among wind turbines (Marden et al. (2013)). The interactions through wakes and their effect on the output power of wind farm have been investigated experimentally in (Adaramola and Krogstad (2011), Dahlberg (2009)) and the results show that the serious power loss for the entire wind farm is made below rated wind speeds.

To reduce the effect of wake interactions and increase the power output of wind farm, the cooperative control of wind farm has attracted the great interest of researchers (Park and Law (2016)). The cooperative control strategies of wind farm can be classified into two categories, the model-based methods and model-free methods.

The model-based methods for wind farm can be commonly divided into the following three steps. The wake interaction model among wind turbines is firstly constructed. Then, the optimization problem that maximizes the output power of wind farm is formulated. The cooperative control strategy of wind farm is finally developed by solving the formulated optimization problem. As examples, in Park et al. (2013), the wake interaction model based on Park model is linearized by the first order Taylors expansion and thus the steepest descent method is employed to obtain the optimal control inputs. In Heer et al. (2014), Jensen model is used to model wake interactions and a heuristic algorithm is proposed to find the optimal set points. The model-based control methods have significant difficulties in practice, mainly because that the widely used wake models, such as Park model and Jensen model, can only represent the ideal characterizations of the wind turbine wake and could not accurately capture the actual system dynamics (Zhong and Wang (2018)). Furthermore, the system parameters in these models are difficult to obtain in practice, especially for some wind farms built in coteau or highland area. Consequently, the model-based methods usually have limited practical performance.

The model-free methods, on the other hand, aim to maximize the output power of wind farm using only the control

⋆ This work was supported by the National Natural Science Foundation of China under Grants 61722307 and 5191101838.

inputs and the power measurements without needing an analytic expression of the wake model (Gratacos (2017)). For examples, in Marden et al. (2013), two game theoretic learning algorithms are proposed, including safe experimentation dynamics and payoff-based distributed learning for Pareto optimality. In Park and Law (2016), a Bayesian ascent algorithm is developed, which can rapidly and almost monotonically look for a local optimum. However, the above algorithms are carried out under constant wind condition. A distributed simultaneous perturbation approach is applied to wind farm for energy maximization, which accommodates slowly changing wind condition (Xu and Soh (2016)). A adaptive scheme is presented by using gradient-based optimization technique (Gebraad and Wingerden (2015)), which can adapt to the changing wind direction quickly. Since the strong wake interactions among many turbines may exist, it will be insufficient for the turbines that their gradient information is estimated by only using information from the nearest neighbouring downstream turbine. Two decentralized discrete adaptive filtering algorithms are proposed in Zhong and Wang (2018). However, the sharp change of control action may happen when the adaptive filtering algorithms are applied, which is not desirable for wind turbines due to the limitations in stability and the unexpected drops of the output power of wind farm (Park and Law (2016)). In Graf et al. (2019), a combination of the alternating direction method of multipliers and reinforcement learning is developed to optimize the output power of wind farm, where it is very difficult to achieve the effective partition of turbines due to the complexity of wake interactions.

To address the aforementioned limitations, this paper proposed a model-free wind farm control scheme using reinforcement learning by developing a decentralized Q learning method. Q learning is a machine learning algorithm, aiming at enabling the agent to learn how to behave through interactions with the environment (Hung and Givigi (2017), Sutton and Barto (1998)). A key advantage of Q learning is that it does not need the prior model of the environment and it also has the ability to escape local minima due to its stochastic optimization property with many successful applications (Anderlini et al. (2017), Liu et al. (2017), Kofinas et al. (2017), and Xu et al. (2012)), making Q learning a particularly suitable tool for our power optimization problem of wind farm. As an online learning method by interacting with the system, Q learning has ability to adapt to the dynamic environment of wind farm, e.g. time varying wind condition. Since the action can be selected from the predesigned action set by Q learning, the gradual changes of control variables of wind turbines can be guaranteed, and thus the unexpected drop of output power of wind farm can be avoided. Simulation is performed in different wind conditions and results are close to the optimal behaviours of all turbines without requiring any prior knowledge on wake interaction model. As far as we know, the model-free wind farm power optimization scheme based on decentralized Q learning is firstly developed in this paper.

The remaining parts of this paper are organized as follows. In Section 2, the wind farm power optimization problem is described. In Section 3, a decentralized model-free control scheme is proposed based on Q learning algorithm to perform the power optimization of wind farm. Section 4 presents the simulation results to demonstrate the effectiveness of the proposed scheme and finally, conclusions and possible directions for future research are given in Section 5.

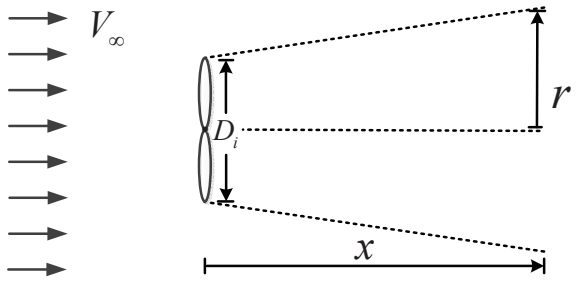


Fig. 1. Single turbine wake example

2. WIND FARM POWER OPTIMIZATION PROBLEM

In this section, the power generation model of wind farm and its power optimization problem are formulated.

A wind farm with n wind turbines is considered and let $N = \{1, 2, \cdots, n\}$ be the set of all turbines. The control variable of wind turbine $i \in N$ is chosen as its axial induction factor (AIF) u_i, whose admissible set is given by the set $\mathcal{U}_i = \{u_i \,|0 \leq u_i \leq 0.5\}$. The AIF accounts for the reduction of the wind velocity over rotor plane, which can be adjusted by the blade pitch and generator torque. The joint axial induction factor of all turbines is represented by the tuple $u = (u_1, \cdots, u_n)$, whose admissible set is denoted as $\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_n$, where $\times$ is the Cartesian product.

When one wind turbine extracts energy from the wind, it will cause changes of the downstream wind flow (Boersma (2017)). The altered flow is called the wake of wind turbine, through which the upwind turbine will affect the wind speed and output power of downwind turbines and thus decreases the power output of whole wind farm. A key modelling challenge for wind farm is describing the interactions among the turbines due to wakes (Marden et al. (2013)). The Park model (Katic et al. (1986)) is the one of the most popular wake models, which gives the wake velocity profile of wind turbine (Xu and Soh (2016)). It is also applied in this paper to resemble the interactions between the turbines.

Consider the situation in Fig. 1, where turbine i is the only turbine. V_∞ is the freestream wind speed. D_i denotes the diameter of turbine i, x is the distance from turbine i along the wind direction, and r is the distance orthogonal to the wind direction. Between the top and bottom dotted lines is the wake area generated by turbine i. Denote $V_i(x, r, u_i)$ as the wake velocity profile at point (x, r) generated by turbine i with the AIF u_i. According to Park model,

$$V_i(x, r, u_i) = V_\infty (1 - \delta V_i(x, r, u_i))$$

where $\delta V_i(x, r, u_i)$ represents the fractional deficit of the velocity at the point (x, r) and is expressed as

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

where k is the roughness coefficient that measures the slope of the wake expansion from turbine (Marden et al. (2013)).

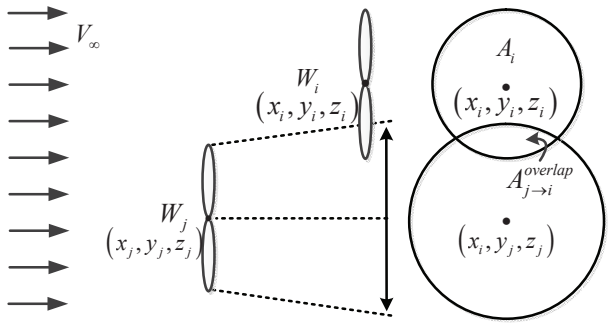


Fig. 2. A two-turbine wake interaction example

Based on the Park model, the aggregate wind velocity $V_i(u)$ at an arbitrary wind turbine i can be calculated as follows

$$V_i(u) = V_\infty(1 - \delta V_i(u))$$

and the aggregated velocity deficit $\delta V_i(u)$ generated by all the upstream turbines of turbine i can be formulated as

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

where x_i is the distance of turbine i from a common vertex along the wind direction, A_i is the area of the disk generated by the blade of turbine i, $A_{j\to i}^{overlap}$ is the part area of the A_i that overlaps with the wake generated by turbine j. An example with two wind turbines is given in Fig. 2, where turbine i is denoted as W_i and its 3-D location relative to the common vertex is defined as (x_i, y_i, z_i).

Remark 1: Note that the Park model will only be used for simulating the wake, whose uncertainties do not influence the performance evaluations of the algorithm in the power optimization problem of wind farm because it is not used in the control design.

The power generated by turbine i is given by

$$P_i(u) = (1/2)\rho A_i C_p(u_i) V_i(u)^3 \tag{1}$$

where ρ is the density of air and $C_p(u_i)$ is the power efficiency coefficient defined as

$$C_p(u_i) = 4u_i(1-u_i)^2 \tag{2}$$

The total output power of wind farm is simply the sum of the power generated by all individual turbines (Zhong and Wang (2018)), namely

$$P(u) = \sum_{i\in N} P_i(u) \tag{3}$$

In this paper, we focus on developing a cooperative control scheme to increase the total output power (3) of wind farm. More specifically, the optimal joint axial induction factor should be obtained by solving the following optimization problem without using wake interaction model:

$$u^{opt} = \arg\max_{u\in\mathcal{U}} P(u) \tag{4}$$

Remark 2: From (1) and (2), it can be shown that ignoring the turbine interactions, i.e. the couplings between the aggregate wind velocity $V_i(u)$, the $u_i = 1/3$ for wind turbine i is the optimal input. However, as mentioned earlier, this called greedy policy might not be optimal in maximizing the total output power of the wind farm due to the wake interactions among the turbines.

3. DECENTRALIZED Q LEARNING METHOD FOR WIND FARM POWER OPTIMIZATION

In this section, a decentralized model-free control scheme for wind farm is developed based on Q learning algorithm to solve the optimization problem (4).

3.1 Wind Farm Power Optimization as Multi-agent Markov Decision Process

To achieve this, the power optimization problem of wind farm is formulated into a multi-agent Markov decision processes, which is denoted by a tuple $(I, S, \mathcal{A}, T, r)$: $I = \{1, 2, \cdots, n\}$ is a set of agents, where the agent $i \in I$ represents the controller of wind turbine i; $S = \times_i S_i$ is the set of states of wind farm and $\times$ is the Cartesian product operator, where S_i is the set of observable states of agent i and the state $s_i \in S_i$ is defined as $s_i = [u_i\ V_d]^T$, where V_d denotes wind direction; $\mathcal{A}$ is the set of joint actions of all agents and $\mathcal{A} = \times_i \mathcal{A}_i$, where $\mathcal{A}_i$ denotes the set of action a_i of the agent i and is defined as

$$\mathcal{A}_i = \{-\Delta u_i, 0, +\Delta u_i\} \tag{5}$$

where Δu_i is the change of AIF u_i of the wind turbine i. The update formula of u_i is designed as

$$u_{i,t+1} = u_{i,t} + a_{i,t+1} \tag{6}$$

where $a_{i,t+1} \in \mathcal{A}_i$ denotes the action taken by agent i at time step $t+1$. Note that the change of control variable u_i from time step t to $t+1$ is decided by the action $a_{i,t+1}$, whose admissible set $\mathcal{A}_i$ can be designed in advance as (5). Therefore, the gradual change of control variable u_i for wind turbine i can be guaranteed, and the unexpected drop of output power of wind farm can be avoided. T is the state transition probability function of wind farm, defined as $T : S \times \mathcal{A} \times S \to [0, 1]$. It is very challenging to build a model for T due to the great complexity of the wake interactions among the turbines. Then model-free reinforcement learning method is desired for the power optimization problem of wind farm. r is a reward function and the reward r_{t+1} that agent i receives at time step $t+1$ is designed as

$$r_{t+1} = P_{t+1} - P_t \tag{7}$$

where r_{t+1} represents the change of output power of wind farm for being in state $s_t = [s_{1,t}, \cdots, s_{n,t}] \in S$ and taking the action $a_t = [a_{1,t}, \cdots, a_{n,t}] \in \mathcal{A}$.

The goal of agent i is to find an optimal policy h_i^* that maximizes its return shown in (8). The return is the expected cumulative aggregation of discounted reward (7) while starting from a given state $s_{i,0}$, taking a given action $a_{i,0}$, and following policy h_i.

$$Q_i(s_{i,0}, a_{i,0}) = E_{h_i}\left(\sum_{t=0}^{\infty}\gamma^t r_{t+1}\right) \tag{8}$$

where $\gamma \in [0, 1)$ is the discount factor. All agents receive a common reward from (7) and then they have same goal. Thus, the power optimization problem (4) of wind farm is modelled as a fully cooperative game. In this paper, the discount factor γ in (8) is set as zero. This means that each agent only considers the one-step reward. The action

taken by agent $i \in I$ at time step t is to maximize the output power of wind farm at time step $t+1$.

3.2 A Decentralized Q Learning Method

The Q-learning was first proposed in (Watkins (1989)), which has been a very popular model-free reinforcement learning method. It consists of two parts, namely Q value estimation and action selection. Fig 3 shows the schematic diagram of Q learning algorithm developed for agent $i \in I$. The Q value is firstly estimated by an Q function (action-value function), which gives the expected return of taking action $a_i \in \mathcal{A}_i$ in a given state $s_{i,t}$. Then an action $a_{i,t}$ is selected from $\mathcal{A}_i$ by using action selection policy.

In the traditional Q learning, the Q function is modelled as a lookup table (Wei et al. (2016)). After each state transition, the Q function is updated by using observed state transition and reward, i.e., data tuple of the form $(s_{i,t}, a_{i,t}, s_{i,t+1}, r_{t+1})$, based on the following iterative formula (Watkins and Dayan (1992)):

$$Q_{i,t+1}(s_{i,t}, a_{i,t}) = Q_{i,t}(s_{i,t}, a_{i,t}) + \eta\left[r_{t+1} + \gamma \max_{a_i} Q_{i,t}(s_{i,t+1}, a_i) - Q_{i,t}(s_{i,t}, a_{i,t})\right] \quad (9)$$

where $\eta \in (0,1]$ is the learning rate and specifies how far the current estimate $Q_{i,t}(s_{i,t}, a_{i,t})$ is adjusted toward the update target $r_{t+1} + \gamma \max_{a_i} Q_{i,t}(s_{i,t+1}, a_i)$. The expression in the square bracket is the temporal difference, i.e., the difference between the estimates of the optimal Q value of $(s_{i,t}, a_{i,t})$ at two successive time steps $t+1$ and t. The observable state s_i of agent i is made up of its AIF u_i and wind direction V_d, which are continuous variables. Then the state space S_i is so large that the lookup table cannot store the Q values of all the state-action pairs. To eliminate the need of the large lookup table, the Q function of agent i in this paper is also approximated via artificial neural network (ANN) as shown in Fig 4. Based on the formula (9), the weight matrix θ_i of the approximate Q function can be updated by using (Wei et al. (2016))

$$\theta_{i,t+1} = \theta_{i,t} + \eta \Delta\theta_{i,t} \quad (10)$$

$$\Delta\theta_{i,t} = [r_{t+1} + \gamma \max_{a_i} Q_{i,t}(s_{i,t+1}, a_i; \theta_{i,t}) - Q_{i,t}(s_{i,t}, a_{i,t}; \theta_{i,t})] \times \nabla_{\theta_i} Q_{i,t}(s_{i,t}, a_{i,t}; \theta_{i,t}) \quad (11)$$

where $\nabla_{\theta_i} Q_{i,t}(s_{i,t}, a_{i,t}; \theta_{i,t})$ denotes the gradient of the approximate Q function $Q_{i,t}(s_{i,t}, a_{i,t}; \theta_{i,t})$ with respect to its weight matrix θ_i at time step t.

The pure exploitation causes easily the agent to fall into the suboptimal solution when one action is selected, whereas excessive exploration degrades the convergence performance of the Q learning algorithm by consuming too much time (Xu et al. (2012)). To achieve the balance, the ε-greedy algorithm, a classical exploratory policy, is adopted by agent i , which selects action according to

$$a_{i,t} = \begin{cases} \arg\max_{a_i} Q_{i,t}(s_{i,t}, a_i; \theta_{i,t}) & \text{with probability } 1-\varepsilon \\ \text{a random action in } \mathcal{A}_i & \text{with probability } \varepsilon \end{cases} \quad (12)$$

where $\varepsilon \in (0,1)$ is the exploration probability. In (12), with probability $1-\varepsilon$, an action that has maximal Q value is chosen, but with probability ε an action at random is instead selected.

Algorithm 1: Decentralized Q Learning Algorithm for Wind Farm Power Optimisation

1. Initialize learning rate η, exploration rate ε, and weight matrix $\theta_{i,0}$, $i \in I$.
2. measure initial state $s_{i,0}$.
3. for every time step $t = 0, 1, 2, \cdots$, do
4. estimate $Q_{i,t}(s_{i,t}, a_i; \theta_{i,t})$ for $s_{i,t}$ and choose action $a_{i,t}$ from $\mathcal{A}_i$ by ε greedy algorithm (12);
5. apply $a_{i,t}$, measure next state $s_{i,t+1}$ and calculate reward r_{t+1} according to (7);
6. update $\theta_{i,t}$ by (10) and (11);
7. End for

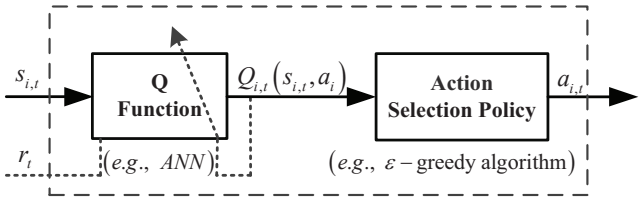


Fig. 3. Schematic diagram of Q learning algorithm for agent $i \in I$.

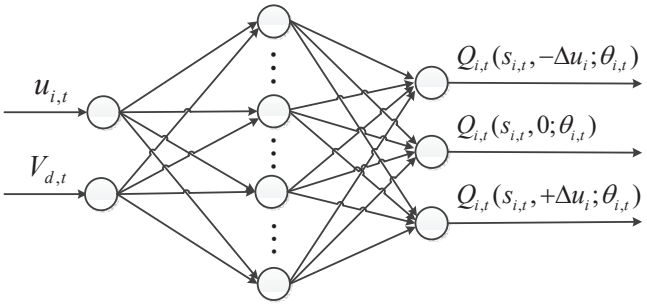


Fig. 4. Schematic diagram of the approximate Q function based on ANN for agent i.

The flow chart of the decentralized Q learning algorithm proposed for wind farm power optimisation is shown in Algorithm 1. Note that during the implement process of the algorithm, it requires no any prior knowledge of the transition function T and learns the optimal policy by online interaction with wind farm. The weight update (10) and (11) of the approximate Q function of agent i is carried out without using the information of other agents. Meanwhile, from Fig. 3 and Fig. 4, it can be observed that the input of the Q learning algorithm for agent i do not use the observable information from other agents. Then, the decentralized solution of optimal joint axial induction factor is finished by the proposed algorithm.

4. SIMULATION RESULTS

In this section, two simulation examples are presented to verify the performance of the decentralized Q learning algorithm in different wind conditions.

As shown in Fig. 5, the wind farm with three turbines is considered. Its layout is an isosceles right triangle area.

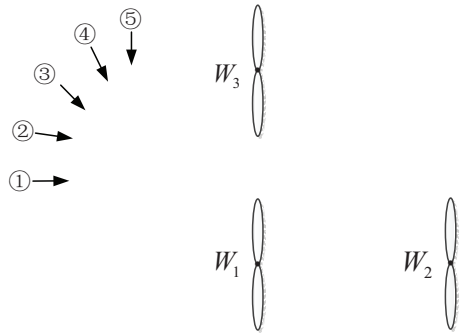
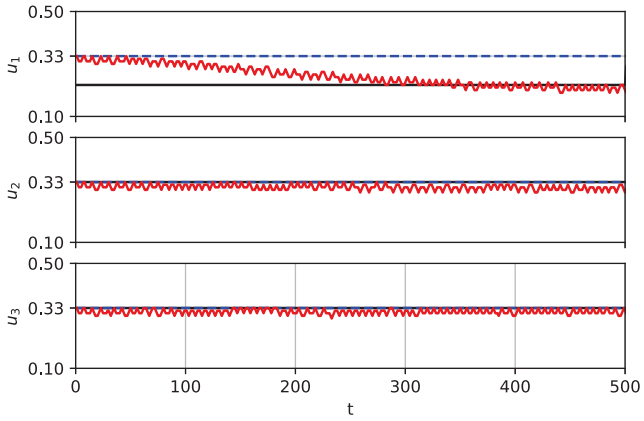
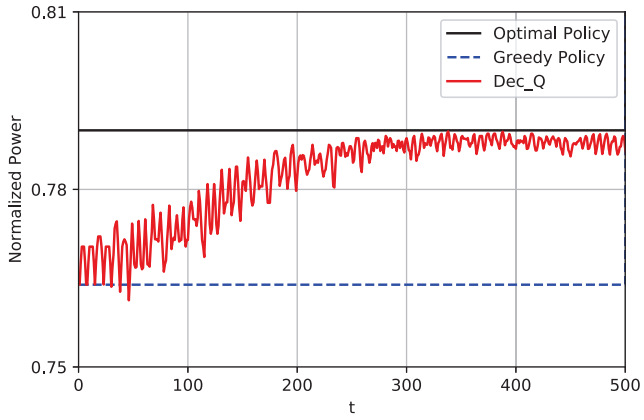


Fig. 5. A 3-turbine wind farm with various wind condition.



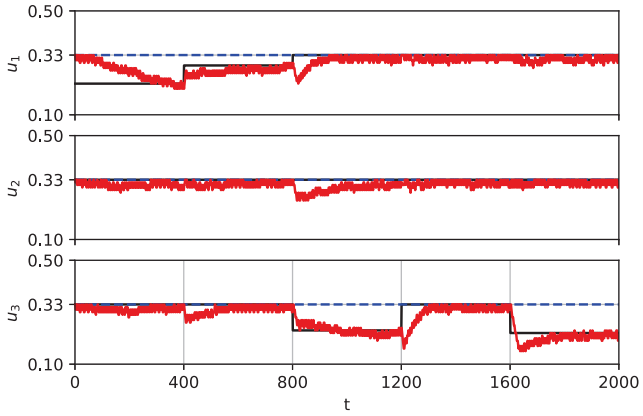
(a) The evolution of AIFs.



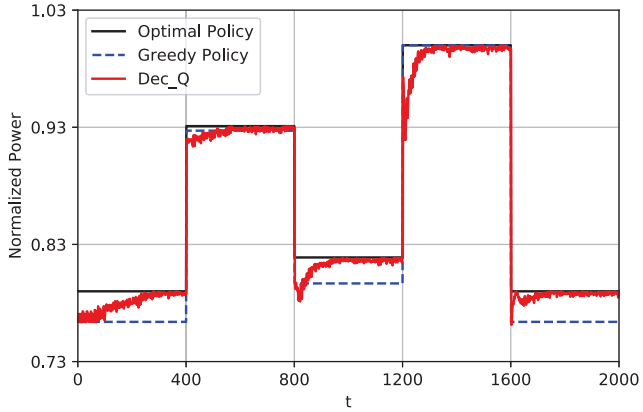
(b) The evolution of normalized power on wind farm.

Fig. 6. Simulation results on wind farm under constant wind condition.

The spacing between turbine W_1 and W_2 is 400 m. All the turbines are of the same size and have a diameter D of 80 m. The roughness coefficient is $k = 0.075$. The density of the air is $\rho = 1.225kg/m^3$. The upstream wind speed is set as V_∞=$8m/s$. A common AIF set $\mathcal{U}_i = \{u_i \,|0.1 \leq u_i \leq 0.33\}$ is applied to turbine $i \in N$ in this simulation. It is a subset of the general AIF set of $\mathcal{U}_i = \{u_i \,|0 \leq u_i \leq 0.5\}$, but it is sufficient to show the performances of algorithms (Zhong and Wang (2018)). Suppose that the yaw control of turbines can guarantee their blade disk planes be perpendicular to the wind direction. The action set of agent i is designed as $\mathcal{A}_i = \{-0.01, 0, 0.01\}$, which can guarantee the gradual change of control input u_i of the turbine i according to (6). The initial joint AIF profile of wind farm is set



(a) The evolution of AIFs.



(b) The evolution of normalized power on wind farm.

Fig. 7. Simulation results on wind farm under time varying wind condition.

as $u_0 = (0.33, 0.33, 0.33)$. The parameters of Q learning algorithm are set as $\varepsilon = 0.01$, $\eta = 0.95$. The three-layer neutral network is used to model the Q function of agent i, who has two inputs, five hidden nodes, as well as three outputs. For comparison purpose, the wind farm power optimizations based on greedy policy and optimal policy are also performed, respectively. The optimal policies of wind farm under different wind conditions are obtained over the total joint AIF set $\mathcal{U}$ by exhaustive search.

4.1 Simulation results under constant wind condition

In this example, the constant wind direction is used, which points horizontally from west to east. Fig. 6 presents simulation results, including the evolution of the AIFs and the evolution of normalized power of wind farm. The normalization of wind farm power output is performed by the total power without wake interaction. The Dec_Q in Fig. 6 stands for decentralized Q learning algorithm.

Fig. 6(a) illustrates that the AIFs of all turbines can converge to the optimal values with the application of decentralized Q learning algorithm. Further, there is no sharp change in the control, which is desirable in practice. Fig. 6(b) shows that the output power of wind farm using proposed algorithm achieves great improvement than the one based on greedy policy and approaches the optimal output power. Therefore, the proposed algorithm can improve the power efficiency of wind farm under constant wind condition.

4.2 Simulation results under time varying wind condition

This example assumes that the upstream wind sweeps from west-east direction to north-south direction, having an angle set of $\{0°, 10°, 45°, 65°, 90°\}$ as shown in Fig. 5. The corresponding simulation results are given in Fig. 7.

From Fig. 7(a), it can be observed that the presented algorithm can guarantee the AIFs of all turbines converge to the optimal values under time-varying wind condition. Fig. 7(b) shows that with the proposed algorithm the wind farm obtains higher power output than the one with greedy policy, whose output power is close to the optimal power. It means that the decentralized Q learning algorithm can adapt to time varying wind condition by online learning and improve the power efficiency of wind farm.

5. CONCLUSIONS

In this paper, a model-free decentralized Q learning method is proposed for the power optimization problem of wind farm. Preliminary simulation results indicate that the proposed algorithm can converge to the optimal power without using wake interaction model. It can also adapt to different wind conditions through online learning. This approach also avoids the sharp change of control variable of wind turbine and therefore it is beneficial to the stability of the wind farm. Future research includes improving the algorithm's convergence rate based on eligibility traces or experience replay, rigorous proof of the algorithm's convergence properties, as well as simulation and experimental tests on a large-scale wind farm.

REFERENCES

Adaramola, M.S., Krogstad, P.A. (2011). Experimental investigation of wake effects on wind turbine performance. *Renewable Energy*, 36(8), 2078-2086.

Anderlini, E., Forehand, D., Stansell, P., Xiao, Q., and Abusara, M. (2017). Control of a point absorber using reinforcement learning. *IEEE Transactions on Sustainable Energy*, 7(4), 1681-1690.

Boersma, S., Doekemeijer, B.M., Gebraad, P.M.O., Fleming, P.A., Annoni, J., Scholbrock, A.K., Frederik, J.A., and Wingerden, J.W.V. (2017). A tutorial on control-oriented modeling and control of wind farms. *2017 American Control Conference*, 1-18.

Dahlberg, J.A. (2009). Assessment of the Lillgrund windfarm: power performance, wake effects. Vatenfall Vindkraft AB, 6_1 LG Pilot Report.

Gebraad, P.M.O., Teeuwiss, F.W., Wingerden, J.W., Fleming, P.A., Ruben, S.D., Marden, J.R, Pao, L.Y. (2016). Wind plant power optimization through yaw control using a parametric model for wake effects-a CFD simulation study. *Wind Energy*, 19(1), 95-114.

Gebraad, P.M.O., Wingerden, J.W. (2015). Maximum power-point tracking control for wind farms. *Wind Energy*, 18(3), 429-447.

Graf, P., Annoni, J., Bay, C., Biagioni, D., Sigler, D., Lunacek, M., Jones, W. (2019). Distributed reinforcement learning with ADMM-RL. *2019 American Control Conference*, 4159-4166.

Gratacos, J.O. (2017). *In-operation learning of optimal wind farm operation strategy.* DTU Wind Energy-M-0165.

Heer, F., Esfahani, P.M., Kamgarpour, M., Kamgarpour, M., and Lygeros, J. (2014). Model based power optimisation of wind farms. *2014 European Control Conference*, 1145-1150.

Hung, S.M., Givigi, S.N. (2017). A Q-learning approach to flocking with UAVs in a stochastic environment. *IEEE Transactions on Cybernetics*, 47(1), 186-197.

Katic, I., Højstrup, J., and Jensen, N.O.(1986). A simple model for cluster efficiency. *Proceedings of European wind energy association conference and exhibition*, 407-410

Kofinas, P., Doltsinis, S., Dounis, A.I., Vouros, G.A. (2017). A reinforcement learning approach for MPPT control method of photovoltaic sources. *Renewable Energy*, 108, 461-473.

Liu, Z., Luo, Y., Zhuo, R., Jin, X. (2017). Distributed reinforcement learning to coordinate current sharing and voltage restoration for islanded DC microgrid. *Journal of Modern Power Systems and Clean Energy*, 6(2), 364-374.

Marden, J.R., Ruben, S.D., Pao, L.Y. (2013). A model-free approach to wind farm control using game theoretic method. *IEEE Transactions on Control Systems Technology*, 21(4), 1207-1214.

Park, J., Kwon, S., Law, K.H. (2013). Wind farm power maximization based on a cooperative static game approach. *Proceedings of SPIE - The International Society for Optical Engineering*, 8688(3), 108-111.

Park, J., Law, K.H. (2016). Bayesian Ascent: A data-driven optimization scheme for real-time control with application to wind farm power maximization. *IEEE Transactions on Control Systems Technology*, 24(5), 1655-1668.

Sutton, R., Barto, A. (1998). *Reinforcement Learning: An Introduction.* The MIT Press.

Watkins C.J.C.H. (1989). *Learning from delayed rewards*, Ph.D. dissertation, Kings College, Cambridge, U.K.

Watkins, C.J.C.H., Dayan, P. (1992). Technical note: Q-learning. Machine Learning. *Machine Learning*, 8(3-4), 279-292.

Wei, C., Zhang, Z., Qiao, W., Qu, L. (2016). An adaptive network-based reinforcement learning method for MPPT control of PMSG wind energy conversion systems. *IEEE Transactions on Power Electronics*, 31(11), 7837-7848.

Xu, J.M., Soh, Y.C. (2016). A distributed simultaneous perturbation approach for large-scale dynamic optimization problems. *Automatica*, 72, 194-204.

Xu, Y., Zhang, W., Liu, W., Liu, W., Ferrese, F. (2012). Multiagent-based reinforcement learning for optimal reactive power dispatch. *IEEE Transactions on Systems, Man, and Cybernetics*, 42(6), 1742-1751.

Zhong, S., Wang, X. (2018). Decentralized model-free wind farm control via discrete adaptive filtering methods. *IEEE Transactions on Smart Grid*, 9(2), 2529-2540.