Exploring the application of reinforcement learning to wind farm control

Exploring the application of reinforcement learning to wind farm control

Henry Korb[1], Henrik Asmuth[1], Merten Stender[2] and Stefan Ivanell[1]

[1] Uppsala University, Wind Energy Section, Campus Gotland, 621 57 Visby, Sweden
[2] Hamburg University of Technology, Dynamics Group, Am Schwarzenberg-Campus 1, 21073 Hamburg, Germany

E-mail: henry.korb@geo.uu.se

Abstract. Optimal control of wind farms to maximize power is a challenging task since the wake interaction between the turbines is a highly nonlinear phenomenon. In recent years the field of Reinforcement Learning has made great contributions to nonlinear control problems and has been successfully applied to control and optimization in 2D laminar flows. In this work, Reinforcement Learning is applied to wind farm control for the first time to the authors' best knowledge. To demonstrate the optimization abilities of the newly developed framework, parameters of an already existing control strategy, the helix approach, are tuned to optimize the total power production of a small wind farm. This also includes an extension of the helix approach to multiple turbines. Furthermore, it is attempted to develop novel control strategies based on the control of the generator torque. The results are analysed and difficulties in the setup in regards to Reinforcement Learning are discussed. The tuned helix approach yields a total power increase of 6.8 % on average for the investigated case, while the generator torque controller does not yield an increase in total power. Finally, an alternative setup is proposed to improve the design of the problem.

1. Introduction

The current control paradigm most often employed for wind turbines is standard greedy control, taking into account only a single turbine. However, in a wind farm the effects of wakes can reduce the power production of downstream turbines significantly [1]. Therefore, much research has been conducted to find ways to mitigate wake effects. Approaches using constant derating have not been consistently successful and measured increases in power production were shown to be sensitive to the simulation environment [2]. Therefore, approaches where the derating is altered dynamically have also become of interest in recent years. Already in static approaches a wide range of optimization methods has been applied, ranging from genetic programming approaches to model predictive control [2]. To develop a dynamic strategy, Munters and Meyers applied receding horizon optimization in [3]. In [4] they mimicked the optimal behaviour with a sinusoidal variation of the thrust force. This was later validated by Frederik et al. in [5]. A similar approach that is also based on the sinusoidal variation of control parameters, is the helix approach developed in [6]. In this approach, the blades are pitched individually in such a way that a sinusoidal tilt and yaw moment acts on the wake. This results in a helical deflection of the wake. In this work, a new optimization approach, based on Reinforcement Learning (RL), will be applied to wind farm control. The approach does not make any prior assumptions about the

wake interaction nor does it rely on information about the full flow-field available from numerical simulations. Instead, this method uses only data comparable to that obtainable by a LiDAR sensor. RL is an area of machine learning that has been applied successfully to a wide range of control problems and recently also to flow control problems, such as drag reduction [7]. However, most flow control applications have been limited to laminar, 2D applications due to the large amount of simulated time required. To overcome the prohibitively long wall time necessary for RL, this work uses large eddy simulations via the lattice-Boltzmann method (LBM-LES), which can reduce the wall time of wind turbine simulations significantly [8]. By employing LBM-LES this will be the first time RL is applied to a fully turbulent, 3D flow control problem. The remainder of this paper is organized as follows: first, the basics of RL are re-visited. Next, the setup of the test case of a three-turbine wind farm is presented alongside a description of the RL-framework used. The results of applying the optimization framework to the parameters of the helix strategy are shown as well as results of trying to develop a new control strategy. Finally the results are discussed and an outlook into future work is given.

2. Methodology

2.1. Wind farm control

The power P generated by a turbine depends on the rotor speed ω and the torque exerted by the generator, M_{gen}. The rotor speed of a turbine changes according to

$$I\frac{\mathrm{d}\omega}{\mathrm{d}t} = M_{\text{aero}}(\theta, \lambda) - M_{\text{gen}}, \tag{1}$$

where I is the moment of inertia of the rotor and generator, M_{aero} the aerodynamic moment, that depends on the blade pitch θ and tip speed ratio λ. Typically, wind turbines are operated to maximize efficiency per turbine, commonly referred to as greedy control [9]. Greedy control does not take into account any effect on the downstream turbines in a wind farm. In induction based control, the thrust and power at the upstream turbines are reduced in order to increase the wind speed at the downstream turbines [2]. On the other hand, the aforementioned helix control approach aims at reducing the wake effects by wake-steering [6]. To this end, the blades are pitched so that a rotating moment in tilt and yaw is exerted on the wake, causing a helical deflection of the entire wake. This can be accomplished by prescribing the pitch angle θ_b for blade b according to

$$\theta_b(t) = A_\theta \sin(\psi_b(t) + \omega_{\text{e}} t), \tag{2}$$

where A_θ is the amplitude of the oscillation, ψ_b is the azimuthal angle of the blade and ω_{e} is the excitation frequency of the rotating moment. In this work, Reinforcement Learning is used to control the generator torque as well as the amplitude of the oscillations.

2.2. Reinforcement Learning

Reinforcement Learning (RL) was developed as a mathematical description of the trial-and-error learning observed in humans and animals. It is based on the Markov decision process (MDP), which models the interaction of a so called agent with an environment. The agent takes an action $\mathbf{A}$ and the environment responds by transitioning to a state $\mathbf{S}$ and providing a reward R. Based on the state $\mathbf{S}$, the agent now takes another action $\mathbf{A}$ according to policy π. This cycle continues for an arbitrary amount of time, which is called the episode with length T_{E}. The goal of the learning process is to make the agent choose actions that maximize the reward in the long run. To account for the long-term effects of an action, the discounted return G_t is defined for each time step t in the episode:

$$\mathsf{G}_t = \sum_{t'=t}^{T_{\text{E}}} \gamma^{t'-t} \mathsf{R}_{t'}, \tag{3}$$

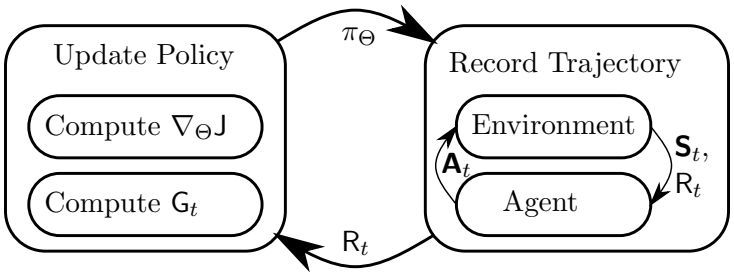

Figure 1. Reinforcement learning algorithm

with the discount rate $\gamma \in [0, 1]$ and $\mathsf{R}_{t'}$ the reward of the time step t'. A low value of γ emphasizes short-term effects while a high value takes into account effects in the far future.[10] In policy gradient methods the policy is given explicitly as a probability function $\pi_\Theta(\mathbf{A}|\mathbf{S})$, from which the action can be sampled while its parameters Θ are optimized to maximize a performance measure J. The gradient of J can then be used to update parameters of the policy according to stochastic gradient descent (SGD) or its extensions such as the Adam algorithm [11]. The updated parameters Θ_{n+1} can be computed via SGD from the current parameters Θ_n by

$$\Theta_{n+1} = \Theta_n + \alpha \nabla_\Theta \mathsf{J}(\Theta), \tag{4}$$

with the learning rate α controlling the step size in the SGD.
Recently the proximal policy optimization (PPO) has been proposed as a performance measure, which has shown great improvements in the convergence rate [12]. It approximates the improvement in expected return of a policy update. In RL the function to learn is often parameterized via an artificial neural network (ANN). A sketch of the entire RL algorithm is given in Figure 1.
In the case of this work, the agent acts as the controller of the turbines, with the action being the controlled variable, e.g. the generator torque. The environment combines the flow field of the wind park as well as the physical response of the turbine, i.e. the rotor speed, power and so on. The reward is defined to be the total power produced by the wind farm. The state consists of the streamwise velocity component of a set of velocity probes as well as the rotor speed of the turbines.
To compute the value of the controlled variable, the information included in the state is normalized and used as the input vector of the ANN, which represents the policy. The output of the ANN is then used as the mean value of a normal distribution. From this distribution an action is sampled. This is then transformed to the value of the controlled variable via an affine transformation and smoothed as suggested in [13].

3. Test case

In this work the wind farm is simulated with the cumulant lattice Boltzmann solver `elbe` [14]. It carries out an implicit LES simulation, where the turbines are modelled as actuator lines [8, 15]. The wind farm consists of a row of three NREL 5MW turbines with a diameter $D = 126\,\mathrm{m}$ and a stream-wise spacing of $L_x = 5D$. The domain measures $6D \times 6D \times 19D$ in the lateral directions y, z, and stream-wise direction x, respectively. A region of nested refinement is placed $2D$ downstream of the inlet in the center of the cross-stream plane. Mann-type synthetic turbulence [16] is introduced at the inlet with a mean wind speed of $V_0 = 10.5\,\mathrm{m/s}$ and a turbulence intensity of $TI_0 = \sqrt{\overline{u'^2} + \overline{v'^2} + \overline{w'^2}}/\sqrt{3}V_0 = 5\%$. One domain flow-through time thus refers to $T_\mathrm{ft} = 228\,\mathrm{s}$. The description of the domain is gathered in Table 1. The RL algorithm is implemented using the TF-Agents library [17], utilizing the PPO algorithm. Following an initial testing of multiple

Table 1. Flow conditions and domains

Quantity	Value
Outer Region	$6D \times 6D \times 19D$
Coarse Resolution	8 nodes/D
N grid points	$364,952$
Inner Region	$5D \times 5D \times 16D$
Fine Resolution	16 nodes/D
N grid points	1.721.344
TI_0	5%
V_0	10.5 m/s
T_{ft}	228 s
Δt	4.3588×10^{-2} s

Table 2. Parameters of trained agents

Parameter	H	M-long	M-short
controlled variable	A_ϕ	M_{gen}	M_{gen}
γ	0.95	0.99	0.95
T_{E}	500 s	1500 s	500 s

architectures, an ANN comprised of three layers of long short-term memory cells [18] was chosen as the policy. The first two layers have a width of 400 cells and the last of a width of the number of controllable variables, i.e. three in the case of this wind farm. The network is updated via the Adam algorithm. The ANN sets the expected value of a normal distribution with a variance of 0.1. The agent interacts with the environment with a frequency of 1 Hz in simulated time. The velocity probes comprising the state are placed in groups of 13 in the shape of a cross with a span of 1.5 D. The crosses are placed in the center of the cross-stream plane, two crosses are placed upstream of the first turbine with a distance of D, while four are placed after the first and second turbine respectively. All quantities in the state vector (velocities and turbine properties) are normalised to a range of -1 to 1 before being fed to the ANN. In practice, multiple episodes are averaged to increase the certainty of the policy updates. These episodes can be simulated in parallel, since the same policy is used [13]. For this work, six independent simulations are run in parallel, as to optimally use the computing resources.

Three different approaches are considered, in the first one, the agent controls the amplitude of the oscillations of the pitch angle, A_ϕ, in a park controlled with the helix approach. For the sake of brevity this agent will be referred to as agent H. The Strouhal number of the oscillations is set to $St = \frac{\omega_{\mathrm{e}} D}{2\pi V_0} = 0.25$, in accordance with [6]. In the other two approaches, agents M-long and M-short respectively, control the generator torque. They differ in the discount rate and the length of the episode. While the values for M-short were kept consistent with the setup of Agent H, Agent M-long puts more emphasis on long-term effects. Further information about the three agents is gathered in Table 2. A comparison of T_{E} with the width of the time steps of the simulation, Δt, shows the exceptionally high computational cost per update, e.g. almost 2.2×10^{10} node updates have to be performed for one update of agent M-short. Other learning parameters, such as learning rate, variance of the action and hyperparameters of the ANN were tuned in preliminary studies not shown here for the sake of brevity.

4. Optimizing the Helix approach - Results and Discussion

First the agent H is trained. This is done to demonstrate the general ability of the framework to optimize a simple parameter of a control strategy that has been shown to increase power production. The agent is trained for about 3750 T_{ft}, equivalent to more than 5×10^6 s of total simulated time. Figure 2 shows the controlled variable and the reward, i.e. the amplitude of the oscillation and the produced power, respectively, throughout the training. By the end of the training the controller has not yet reached a stable point, however, the same amplitudes were set at the first and second turbine multiple times. It is therefore assumed that the agent

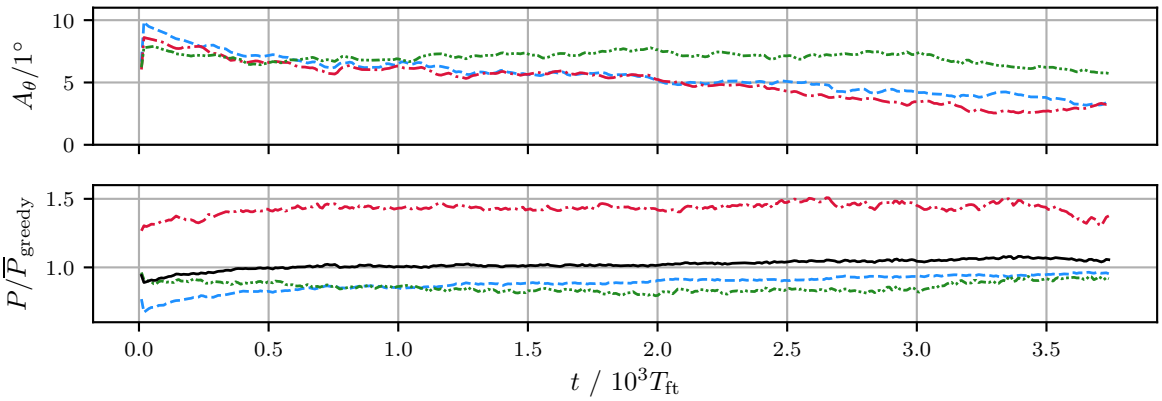

Figure 2. — Total, - - Turbine 1, -·- Turbine 2, -··- Turbine 3. Rolling average of amplitude and power production of the wind farm throughout the training of agent H. Respective powers normalized by mean per turbine of a greedy controlled park.

will return to setting the same amplitudes at these two turbines while improving only the last turbine. The reasons for this will be discussed later on. The figure shows that the rolling average values of the amplitude change rapidly in the beginning. After this initial phase the set value changes slower. It is notable that the amplitude at the first two turbines decreases continually, with intermittent plateaus. Furthermore, the amplitudes are very similar in value throughout training. It would be expected that the amplitude of the last turbine reduces to zero, since a helical motion after the last turbine is likely to have no impact on the total power production, as there are no wake deficits to mitigate. However, due to the structure of the reward, this turbine, which produces the least power, is the slowest to be optimized. The first and the third turbine produce less power than in the greedy case. Especially the first turbine produces significantly less power in the beginning of training. However, due to a steady increase throughout training it produces nearly as much power as the greedy case after training is completed. The second turbine significantly outperforms the greedy control case, thus leading to an increase in total power produced by the park. With progression of the training the second turbine decreases in power. Yet, the increase in power by the other two turbines compensates for the loss in power leading to a net gain in total power.

A more detailed view of the evolution of the control strategy during the training can be seen in Figure 3. It shows that the agent sets a constant amplitude and therefore regresses to the helix approach as described by Frederik et al. in [6], instead of adding additional frequencies by altering the amplitude for example. The central column shows that this happens quickly. After half the training time the agent exhibits no dynamic reaction to the input. It emphasizes again that the optimization of the last turbine is slowest, as a reaction to the input is still visible in the central column.

The mean power of the run after the training compared to a greedy controlled park is gathered in Table 3. Overall, the mean power is increased by almost 7 %. This is due to an increase by more than 40 % in power production of the second turbine. The first turbine shows that the per-turbine efficiency is reduced by around 5 %. Again we find a decrease in power production of the third turbine. This does not comply with the expectations, since the helix of the second turbine should improve the inflow velocity. This will be investigated further later on.

The averaged flow field as well as the turbulence intensity are shown in Figure 4. As expected it shows a significantly higher inflow velocity at the second turbine. This is caused by increased mixing and deflection due to the helix approach. The helix is visible as the band of decreased

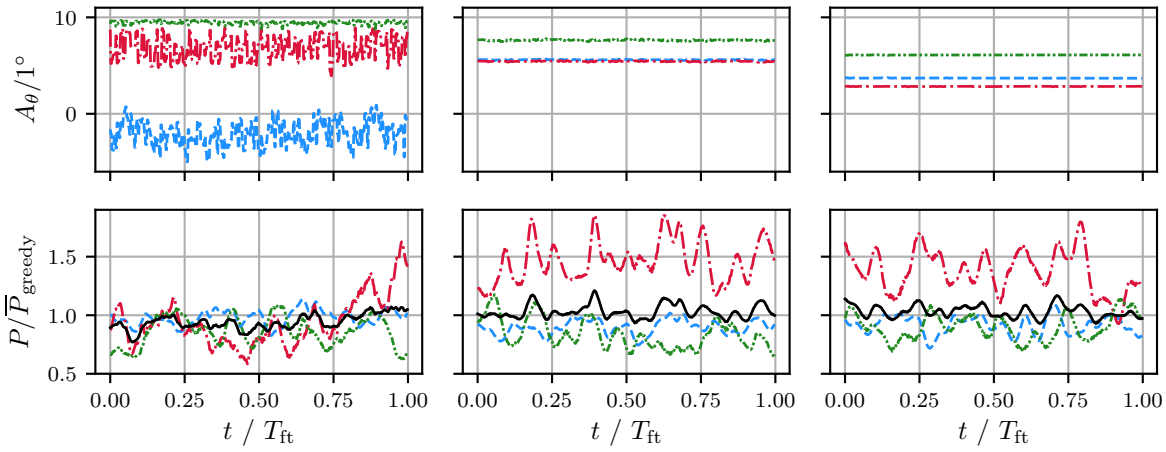

Figure 3. — Total, - - - Turbine 1, -·-· Turbine 2, ·-·-· Turbine 3. Snapshots of amplitude and power production before (left), after half of training (center) and after full training (right) of agent H. Same normalization as in Figure 2.

Table 3. Mean power of a park controlled by agent H compared to a greedy-controlled park.

	P_{greedy}	P_{helix}	
	mean	mean	rel. mean
Total	9.8 MW	10.5 MW	+6.8 %
Turbine 1	5.1 MW	4.8 MW	−5.0 %
Turbine 2	2.6 MW	3.7 MW	+43 %
Turbine 3	2.2 MW	2.0 MW	−8.8 %

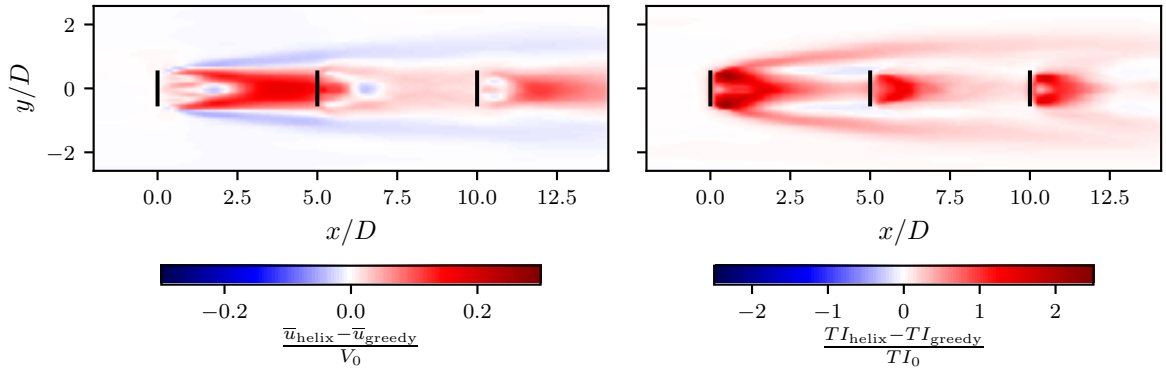

Figure 4. — Turbines. Contour plots of the difference of mean velocity (left) and turbulence intensity (right) between park controlled by agent H and a greedy controller. Average of vertical and horizontal plane.

velocity surrounding the area of increased velocity. However, this trend can not be repeated after the second turbine, where the wake velocity is only slightly increased in comparison to the greedy case. Thus, the inflow conditions at the third turbine are only slightly improved. Furthermore,

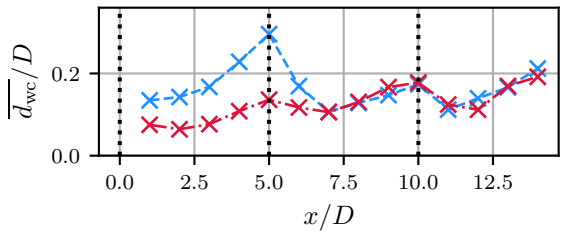

Figure 5. --- Helix, -·- Greedy, ····· Turbines. Average distance of wake centers to rotor centers in 14 cross-stream planes. Wake centers were calculated for snapshots of the cross-stream flow field.

the greedy controlled turbine reduces the wind velocity more than the helix controlled turbine. Therefore, the efficiency of the third turbine must have decreased. Combining the lowered efficiency with the only slightly improved inflow speed explains the decreased power production. The turbulence intensity field shows that an area of high TI exists immediately behind each turbine. Furthermore, a hull of increased turbulence intensity can be found due to an increase of wake steering. In front of the second turbine the turbulence intensity in the center is elevated whereas the turbulence intensity near the tips decreased. The wake of the second turbine shows a similar characteristic as the first wake but less pronounced.
The different large scale motions of the wakes are apparent in Figure 5. It shows the mean distance of the wake center to the center of the rotors, $\overline{d_{\mathrm{wc}}}$, computed via a 2D gaussian least-square fit using the samwich toolbox[1] [19]. Due to wake meandering $\overline{d_{\mathrm{wc}}}$ is not zero in the greedy case. However, when controlled by agent H, the mean distance increases with a more pronounced helical motion. It clearly shows that a helix exists after the first turbine but that the helix collapses in the wake of the second turbine. Thus, the velocity deficit behind the second turbine and, hence, the power of the third turbine do not improve. Investigating the reasons for the collapse of the helix are outside the scope of this work, but are necessary to extend the helix approach to a larger number of turbines.
The results in this section show that the RL algorithm requires a lot of simulated time for optimization. However, it was shown that the agent is able to optimize the parameters of the helix approach and found an increase of almost 7 % in total power.

5. Developing a new control strategy - Results and Discussion

5.1. Results

In this section the training of two agents controlling the generator torque is described. First, the training of an agent using long episodes and a high discount rate, agent M-long, is discussed.
The progression of the training is shown in Figure 6. The training was carried out for 5000 $\mathrm{T_{ft}}$, equating to almost 7×10^6 s of total simulated time. The time series of the generator torque shows a steady increase at the first turbine and to a smaller extend also at the third turbine. The average generator torque at the second turbine changes little throughout the training. The torque at the second and the third turbine is also significantly lower than in the greedy control case. Only the torque of the first turbine reaches values similar to that of a greedy controller. In general, the lower generator torque set at all turbines also results in a decreased power production. Parallel to the generator torque, the power production of the first turbine increases continuously. The third turbine shows little improvement, while the power produced by the second turbine decreases slightly. The different rates of improvement can be explained by the

[1] https://github.com/ewquon/waketracking

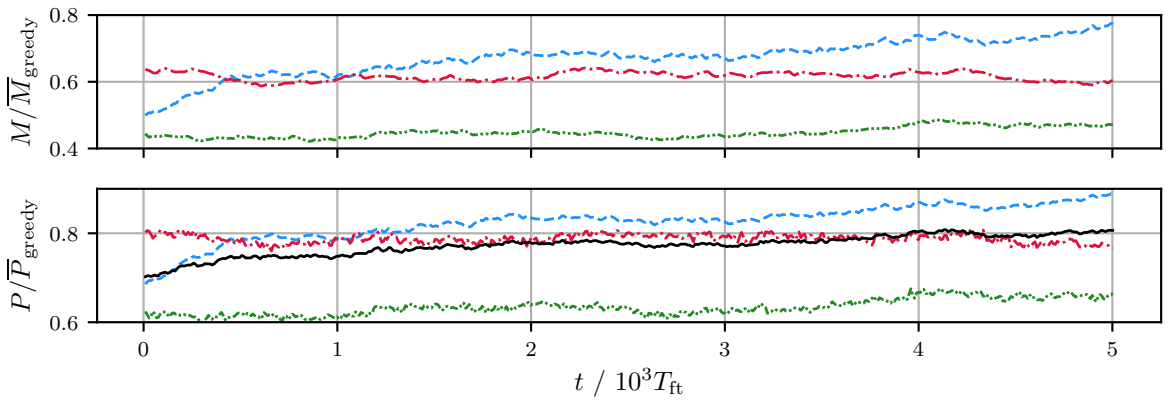

Figure 6. — Total, - - Turbine 1, -·- Turbine 2, -··- Turbine 3. Rolling average of generator torque and power production of wind park controlled by agent M-long. Same normalization as Figure 2.

definition of the reward. Since the first turbine produces the most power, a relative increase yields a greater absolute increase in total power, the quantity that is used as the reward.

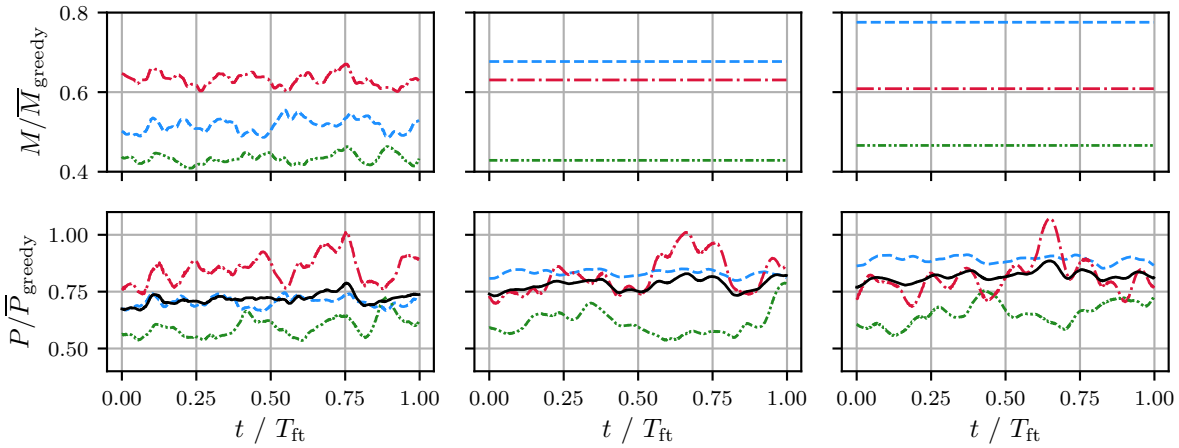

Figure 7. — Total, - - Turbine 1, -·- Turbine 2, -··- Turbine 3. Generator torque and power production of a wind park controlled by agent M-long. Before (left), after half of training (center) and after training (right). Same normalization as in Figure 2.

Figure 7 presents a similar picture. The power production at turbine 1 is highest, while the second and third turbine produce significantly less power. Notably the agent regresses to setting a constant generator torque at all three turbines after half of the training and only changes the value of the constant torque. This strategy can be viewed as a slowly reacting greedy controller, since the inflow is of constant mean velocity, superimposed with turbulent fluctuations. It can be expected that with further training, the agent will evolve to set generator torques at all three turbines closer to the mean torque set by a greedy controller. However, due to the high computational cost and the slow progression of the training, this was deemed too costly. To reduce the high computational cost per network update, a second agent, agent M-short, was trained. To reduce the simulated time necessary per update, a shorter period was used.

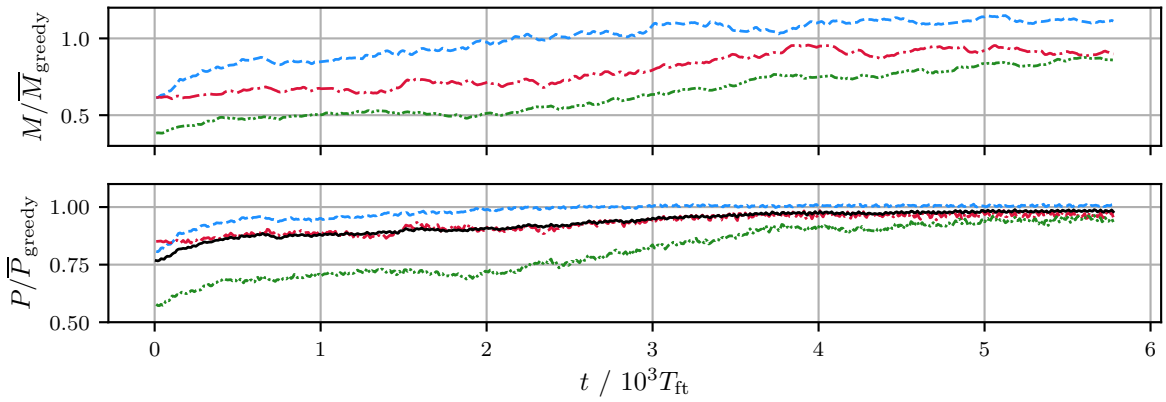

Figure 8. — Total, -- Turbine 1, -·- Turbine 2, -·· Turbine 3. Rolling average of generator torque and power production of a wind park controlled by agent M-short. Same normalization as in Figure 2

The rolling average of the generator torques and power production are shown in Figure 8. In comparison to the previous agent it can be seen that the agent evolves faster. Moreover, the generator torque is in the same range as the mean greedy torque. The power production by the first turbine is very close to that of the greedy controller, while the second and third turbine exhibit somewhat lower power production. Thus, the total power produced is slightly less than in the greedy comparison case. As was the case for the training of other agents, the turbine with the highest power production also improves the fastest. The generator torques still change in the end of the training. However, this change is small in comparison to the earlier changes and the power generated per turbine shows minor reactions to the changes. It is therefore assumed, that the training has reached a final state and the agent will not improve further.

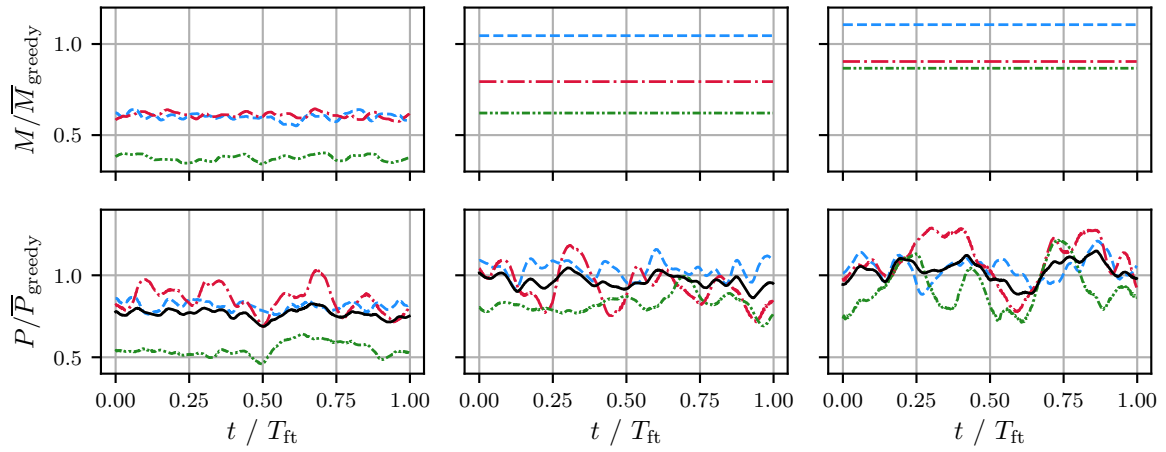

Figure 9. — Total, -- Turbine 1, -·- Turbine 2, -·· Turbine 3. Generator torque and power production of wind park controlled by agent M-short. Before (left), after half of training (center) and after training (right). Same normalization as Figure 2.

The development of the control strategy can be seen at Figure 9. It shows mostly the same progression as the last agent. The generator torque seizes to react to the the turbulent

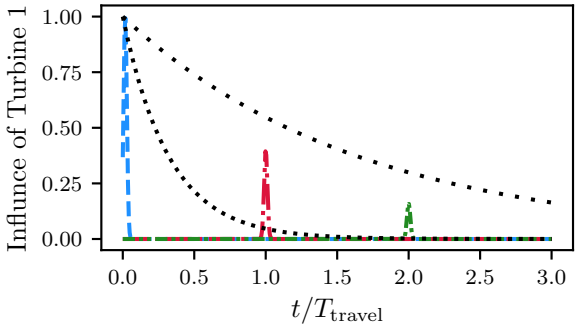

Figure 10. Turbine 1, Turbine 2, Turbine 3, $\gamma = 0.95$ $\gamma = 0.99$. Illustration of the influence of actions at the first turbine on other turbines compared to discounting factor.

fluctuations and simply sets a constant torque. Further training only changes the value of the constant torque. The only difference is the faster evolution of the constant value. This is due to three times more updates per simulated time. The instantaneous power production shows a similar picture. After half of the training, the first two turbines already show a power production in the range of the greedy controller. After the full training the third turbine has improved to the same range as the greedy controller as well. However, the goal was not to redevelop the greedy controller, but to find strategies that improve the total power production.

5.2. Discussion

Finding the reason why the agents were not able to discover strategies that improve the farm efficiency naturally relies on heuristic approaches, as the neural network acts a black-box and an examination is difficult to impossible [20]. Assuming that the PPO-algorithm is robust and correctly implemented, the main reasons must be found in the choices made for this work and in the coupling of the CFD-simulation to the PPO-algorithm. The values used as input for the ANN need to be normalized. In this work an *a priori* defined normalization was used. However, it can be argued, that other normalizations, which in turn would allow for more sensitive control, can be advantageous. This is a possible explanation for the neglect of the dynamics of the input parameters. Another major difficulty in the formulation of the optimization problem is the definition of the reward. In this work it was always chosen to be the total power production. However, due to their positioning in the domain, the first turbine will naturally produce significantly more power and therefore improving the efficiency of the first turbine holds the highest increase in reward. The third important factor is the length of the period and the discount rate. A higher discount rate emphasizes long term effects, but requires a longer period leading to less updates per simulated time. This is true in general, yet the problem of wind farm control features another difficulty. Every action is associated with a return. The higher the correlation between action and return, the more efficient the optimization. In the studied case, the action consists of three components, i.e. the generator torque set at each respective turbine. The return on the other hand combines the influences of the actions into a single value. Furthermore, the action influences the reward multiple times, because the influence on the flow-field is carried downstream to the other turbines. This is illustrated in Figure 10. The discount rate sets the rate at which the future rewards' weights decay in the calculation of the return. In order to capture the influence an action at the first turbine has on the second turbine, a high discount rate needs to be chosen. However, this also implies the inclusion of many time steps in between, when the reward is not influenced by the action. This decreases the overall correlation of action and return. This is especially critical in a turbulent flow, since

the stochastic nature of turbulence already introduces noise into the reward signal. In order to circumvent this problematic formulation we propose a rephrasing of the problem. Instead of one agent and one environment, each turbine could be controlled by a different agent, each with their own definition for the reward. The reward would be formulated in such a way that it only includes information that can possibly be influenced by the action. For example the reward for the agent controlling the first turbine would be defined as

$$\mathrm{R}_t^1 = P_t^1 + P_{t+T_{\mathrm{travel}}}^2 + P_{t+2T_{\mathrm{travel}}}^3 \tag{5}$$

where T_{travel} is the minimum time required by information to travel from one turbine to the next, hence approximately L_x/V_0. This would allow for a lower discount rate and decreases the noise in the reward signal, while the wake interaction is still taken into account.

6. Conclusion

In this work, Reinforcement Learning was applied to a fully turbulent flow control problem for the first time. It was used to maximize power production of a small wind farm of three turbines. When applied to the optimization of the helix approach it successfully improved some of its parameters although the high computational cost forbid a full optimization. It was found that the helical motion of the wake collapsed after the second turbine, leading to smaller increases in total power production than could be expected from previous work. Nevertheless, an overall increase of 6.8 % was found.

When applied to the optimization of the generator torque, the algorithm was not able to discover a new control strategy. Instead the controller set constant generator torques in the range of the mean generator torque set by a greedy controller. Several possible reasons for this were discussed, the main reason being the phrasing of the problem and the time delay in the interaction of the turbines. It was therefore proposed to rephrase the setup of the problem in three separate agents with elimination of the time delays. This will have to be investigated in future work. This work emphasizes the exceptionally high computational cost of RL for control of turbulent flows. However, it also demonstrates that RL for control of 3D, turbulent flows is within reach of modern, fast methods such as the LBM. It also highlights the difficulty of finding the right formulation of the optimization problem. Nevertheless, it also shows how RL might be a viable optimization algorithm in the future.

References

[1] Nilsson K, Ivanell S, Hansen K S, Mikkelsen R, Sørensen J N, Breton S P and Henningson D 2015 *Wind Energy* **18** 449–467 ISSN 1099-1824
[2] Kheirabadi A C and Nagamune R 2019 *Journal of Wind Engineering and Industrial Aerodynamics* **192** 45–73 ISSN 0167-6105
[3] Munters W and Meyers J 2018 *Energies* **11** 177
[4] Munters W and Meyers J 2018 *Wind Energy Science* **3** 409–425 ISSN 2366-7451
[5] Frederik J A, Weber R, Cacciola S, Campagnolo F, Croce A, Bottasso C and van Wingerden J W 2020 *Wind Energy Science* **5** 245–257 ISSN 2366-7451
[6] Frederik J A, Doekemeijer B M, Mulders S P and van Wingerden J W 2020 *Wind Energy* ISSN 1095-4244
[7] Rabault J, Kuchta M, Jensen A, Réglade U and Cerardi N 2019 *Journal of Fluid Mechanics* **865** 281–302 ISSN 0022-1120, 1469-7645
[8] Asmuth H, Olivares-Espinosa H, Nilsson K and Ivanell S 2019 *Journal of Physics: Conference Series* **1256** 012022 ISSN 1742-6588, 1742-6596
[9] Hansen M O L 2008 *Aerodynamics of Wind Turbines* 2nd ed (London ; Sterling, VA: Earthscan) ISBN 978-1-84407-438-9
[10] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* second edition ed Adaptive Computation and Machine Learning Series (Cambridge, Massachusetts: The MIT Press) ISBN 978-0-262-03924-6
[11] Kingma D P and Ba J 2017 *arXiv:1412.6980 [cs]* Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 (*Preprint* 1412.6980)

[12] Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O 2017 *arXiv:1707.06347 [cs]* (*Preprint* 1707.06347)
[13] Rabault J and Kuhnle A 2019 *Physics of Fluids* **31** 094105 ISSN 1070-6631
[14] Janßen C F, Mierke D, Überrück M, Gralher S and Rung T 2015 *Computation* **3** 354–385
[15] Asmuth H, Olivares-Espinosa H and Ivanell S 2020 *Wind Energy Science* **5** 623–645 ISSN 2366-7443
[16] Mann J 1998 *Probabilistic Engineering Mechanics* **13** 269–282 ISSN 0266-8920
[17] Guadarrama S, Korattikara A, Ramirez O, Castro P, Holly E, Fishman S, Wang K, Gonina E, Wu N, Kokiopoulou E, Sbaiz L, Smith J, Bartók G, Berent J, Harris C, Vanhoucke V and Brevdo E 2018 [Online; accessed 25-June-2019]
[18] Hochreiter S and Schmidhuber J 1997 *Neural Computation* **9** 1735–1780 ISSN 0899-7667, 1530-888X
[19] Quon E W, Doubrawa P and Debnath M 2020 *J. Phys.: Conf. Ser.* **1452** 012070
[20] Ghorbani A, Abid A and Zou J 2019 *Proceedings of the AAAI Conference on Artificial Intelligence* **33** 3681–3688 ISSN 2374-3468