TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Maschinenwesen, Institut für Strömungsmechanik, Professur für Strömungsmechanik

Diploma Thesis

Coupling of an artificial neural network with LES-LBM to improve wind farm control

submitted in partial fulfillment of the requirements for the degree "Diplomingenieur"

Henry Torsten Korb

born on 02.03.1996

in Ratingen

submitted on 13. 09. 2020

1st reviewer Prof. Dr.-Ing. habil. J. Fröhlich

2nd reviewer PD Dr.-Ing. habil. J. Stiller

Supervisor M. Sc. H. Asmuth

Co-Supervisor M. Sc. R. Jain

Kurzfassung

Kopplung eines künstlichen neuronalen Netzwerks mit LES-LBM zur Verbesserung einer Windpark-Steuerung

15 zeilen

Abstract

Coupling of an artificial neural network with LES-LBM to improve wind farm control

15 lines

Contents

Nomenclature

Latin Symbols	Unit	Description
a	$-$	Axial interference / induction factor
a'	$-$	Tangential interference / induction factor
$\mathbf{a}$	$-$	Action vector
$\hat{\mathbf{a}}$	$-$	Applied action vector
A	m^2	Area
$\mathbf{A}$	$-$	Action vector
b	$-$	Weights
$\mathbf{b}$	$-$	Vector of biases
$\mathbf{c}$	$\mathrm{m/s}$	Constant velocity vector
c	$\mathrm{m/s}$	Constant velocity
$\mathbf{c}$	$-$	Cell state vector
C	$-$	Coefficient
$C_{\alpha,\beta,\gamma}$	$-$	Cumulant
D	m	Rotor diameter
$\mathbf{e}$	m	Basis vector
E	$\mathrm{J/m}^3$	Energy density
E	$-$	Expected value
f	$\mathrm{kgs}^3/\mathrm{m}^6$	Distribution function
f	$-$	Activation function
$\mathbf{F}$	N	Force
F	$-$	Distribution function in frequency space
G	$-$	Return

I	kgm^2	Moment of inertia
J	–	Performance measure / objective
k	–	Central moment
K	–	Number of layers
L	m	Length
M_{aero}	Nm	Aerodynamic torque
M_{gen}	Nm	Generator torque
$M_{\alpha,\beta,\gamma}$	–	Raw moment
M	–	Number of nodes in layer
Ma	–	Mach number
$\mathbf{n}$	–	Normal vector
N_b	–	Number of blades
$N_{A,e}$	–	Number of actions per episode
$N_{e,b}$	–	Number of episodes per batch
$N_{t,A}$	–	Number of timesteps per action
N	–	Number of inputs into layer
p	Pa	Pressure
p	–	State transition function
pdf	–	Probability density function
pr	–	Probability ratio
P	W	Power
Pr	–	Probability
q	–	Action-value function
r	m	Radius / radial coordinate
r_f	–	Forget ratio

r	$-$	Reward
r_p	$-$	Probability ratio
R	m	Rotor radius
$\hat{R}$	$\mathrm{J/kg/K}$	Specific gas constant
R	$-$	Reward
s	$-$	Solidity
$\mathbf{S}$	$-$	State vector
t	s	Time
St	$-$	Strouhal number
t	$-$	Sensitivity
T	N	Thrust force
TI	$-$	Turbulence intensity
$\hat{T}$	K	Temperature
T	$-$	Length of Episode
$\mathbf{u}$	$\mathrm{m/s}$	Macroscopic velocity vector
u	$\mathrm{m/s}$	Streamwise velocity
$\mathbf{v}$	$\mathrm{m/s}$	Macroscopic velocity vector
v	$\mathrm{m/s}$	Vertical velocity
v	$-$	Value function
$\mathbf{V}$	$\mathrm{m/s}$	Wind velocity
V	m^3	Volume
V_0	$\mathrm{m/s}$	Mean wind speed
w	$\mathrm{m/s}$	Crosswise velocity
W	$-$	Weight matrix of layer
$\mathbf{x}$	m	Vector of position
x	m	Streamwise coordinate

$\mathbf{x}$	$-$	Activation
y	m	Wallnormal coordinate
$\mathbf{y}$	$-$	Layer output
z	m	Spanwise coordinate
$\mathbf{z}$	$-$	Layer input

Greek Symbols	Unit	Description
α	rad	Angle of attack
α_f	$-$	Decay rate of exponential filter
β	$-$	Probability clipping ratio
γ	$-$	Discount rate
Γ	$-$	Noise
δ	$-$	Delta distribution
ϵ	m	Smearing width
ε_{p,l_2}	$-$	Policy l_2-regularization coefficient
$\varepsilon_{v,l}$	$-$	Policy l_2-regularization coefficient
ε_{v,l_2}	$-$	Value loss coefficient
ζ	$\mathrm{m/s}$	Microscopic velocity
Z	$-$	Wavenumber
η	$-$	Gaussian filter kernel
$\boldsymbol{\theta}$	$-$	Parameter vector
θ	$-$	Angle
Θ	$-$	Azimuth angle
κ	Nms^2	Greedy controller proportionality constant

λ	$-$	Tip-speed ratio
μ	$-$	Probability parameter
ν	m^2/s	Kinematic viscosity
$\boldsymbol{\xi}$	m/s	Microscopic velocity vector
ξ	m/s	Microscopic velocity
$\boldsymbol{\Xi}$	$-$	Wavenumber vector
Ξ	$-$	Wavenumber
π	$-$	Policy
ρ	kg/m^3	Density
σ	$-$	Sigmoid function
υ	m/s	Microscopic velocity
Υ	$-$	Wavenumber
ϕ	rad	Angle
φ	$-$	General control variable
Φ	rad	Pitch angle
ψ	$-$	Learning rate
Ψ	rad	Yaw angle
ω	rad/s	Angular velocity
$\hat{\omega}$	$1/\mathrm{s}$	Relaxation frequency
Ω	$\mathrm{kgs}^2/\mathrm{m}^6$	Collision operator

Indices	Description
D	Drag
i	running index
I	Induced
j	running index

k	running index
l	running index
L	Lift
m	Mean
max	Maximum
n	Normal
P	Power
r	Radial
s	Sound
t	tangential
T	Thrust
x	Streamwise
θ	Azimuthal

Additional Symbols	Description
$\|\|(\mathbf{.})\|\|$	Euclidian norm
∇	Nabla operator
$\delta_{i,j}$	Kronecker delta
Δ	Step
$\mathcal{O}(.)$	Order
$(\mathbf{.}) \cdot (\mathbf{.})$	Inner product
$(.)'$	Fluctuation
$[.]^T$	Transposed vector

Abbreviations	Description
ABL	Atmospheric boundary layer
ALM	Actuator line model
ANN	Artificial neural network
BEM	Blade element / momentum theory
BGK	Bhatnagar-Gross-Krook
CFD	Computational fluid dynamics
DNS	Direct numerical simulation
DRL	Deep reinforcement learning
GPU	Graphical processing unit
HAWT	Horizontal axis wind turbine
IEA	International Energy Agency
LBM	Lattice Boltzmann method
LBE	Lattice Boltzmann equation
LES	Large-eddy simulation
LSTM	Long short-term memory
MDP	Markov decision process
MRT	Multiple relaxation times operator
NREL	National Renewable Energy Laboratory
NSE	Navier-Stokes equations
pdf	Particle-distribution Function
PPO	Proximal policy optimization
RL	Reinforcement learning
SGD	Stochastic gradient descent

1. Introduction

While human made climate change only begins to effect the countries of Europe and North America, the predictions for the near and far future are devastating [25]. One of the main drivers of climate change is the increase in concentration of greenhouse gases in the atmosphere. The production of electricity with fossil fuels accounts for up to a third of greenhouse gas emissions in major industrial countries like the United States [24] and Germany [41]. To reduce the emission of greenhouse gases the energy sector is shifting to renewable sources of energy such as wind and solar [27].
To reduce investment and maintenance costs, wind turbines are often arranged in wind farms of multiple, sometimes hundreds of turbines. However, this also reduces the overall efficiency of the turbines due to wake losses [40]. The simplest way to reduce the wake losses would be to increase the spacing between turbines. However, this reduces the aforementioned benefits of wind farms and can not be done after the farm has been built. Another approach is to control the turbines in a way, that reduces the deficits due to wakes. This approach has the advantage of not placing restrictions on the layout of the park and can applied after the park has been built. Wake interaction can be reduced in two ways. The wake is either steered away from the downstream turbines, usually by changing the yaw of the turbine or the wake deficit is reduced by curtailing the upstream turbines. Curtailment was first studied by Steinbuch et al. [54]. Multiple studies since then have tested this strategy and a recent survey of them found that a static curtailment does not significantly increase the overall power production, when tested with high fidelity simulations [32]. However, the exploration of dynamic approaches has also begun. Goit and Munters used receding horizon optimal control to prove that an increase of 16 percent is possible [20]. Based on these results, Munters and Meyers proposed a sinusoidal variation of C_P and also found increased production [39]. This was also confirmed in wind tunnel experiments and further simulations by Frederik et al. [11]. Based on these results, Frederik et al. used an oscillating pitch angle to induce a moment on the wake to steer the wake in a helical motion, also recording increases in power production in a setup of two turbines by 7.5 percent [10].
The interactions of wake and turbines in a wind farm can be viewed as a non-linear system. Methods of machine learning has proven to be a very useful tool to tackle non-linear systems and non-linear control. One method suitable for non-linear control is reinforcement learning (RL). The field of RL emerged as a combination of dynamic programming and the theory of trial-and-error-learning in psychology. They were first combined into the modern field of RL in the 1970s and 1980s through pioneering work by Klopf, Barto and Sutton [55, p. 20-21]. RL has been used in a vast range of applications, such as playing Go and Chess [51] or controlling a Mars lander [14]. One group of methods from the field of RL is policy-gradient methods. In active flow control the first application of policy-gradient methods was by Rabault to reduce drag in a von Kármán vortex street [45]. Furthermore, it was used by Belus et al. to stabilize a thin fluid film [5].

A critical point when combining fluid simulations with RL is the high computational cost of a single time step of the fluid simulation and the large number of time steps required for RL. The lattice Boltzmann method has shown great potential to lower the computational cost of fluid simulations [36]. In contrast to fluid solvers based on the Navier-Stokes equations it can easily take advantage of the highly parallel hardware such as graphical processing units [35]. It is also suitable for highly turbulent flows [15]. Asmuth et al. recently applied it to large eddy simulations of wind farms for the first time [4].
In this work, reinforcement learning will be applied to maximize power production in an LBM-LES simulation of a small wind park. In the first chapter, the theoretical foundations for wind turbine control and simulation, as well as RL and LBM are laid and a more detailed explanation of already existing control strategies will be given. In the second chapter, the implementation and setup of the simulations is explained. In the following chapter the results of the simulations will be presented and examined. Finally a conclusion and outlook into future developments is given.

2. Theoretical Background

2.1. Wind Turbines

2.1.1. Description of a Turbine and Flow Features

First a few basic definitions of the model used for wind turbines will be given. Today's mainstream wind turbines consist of a rotor with three blades, with a horizontal axis, therefore they are called horizontal axis wind turbines (HAWT), and the rotor points upwind. This rotor is mounted to the nacelle, which holds the gearbox and the generator. The nacelle sits on top of the tower. The diameter of the rotor is D. A schematic of a HAWT is given in Figure 2.1. It shows the wind velocity $\mathbf{V}$, the azimuthal angle Θ and the angular velocity ω of the rotor , the pitch angle of one of the blades Φ_0, the yaw angle Ψ of the turbine and the torque due to aerodynamic forces and the generator, M_{aero} and M_{gen}, respectively.

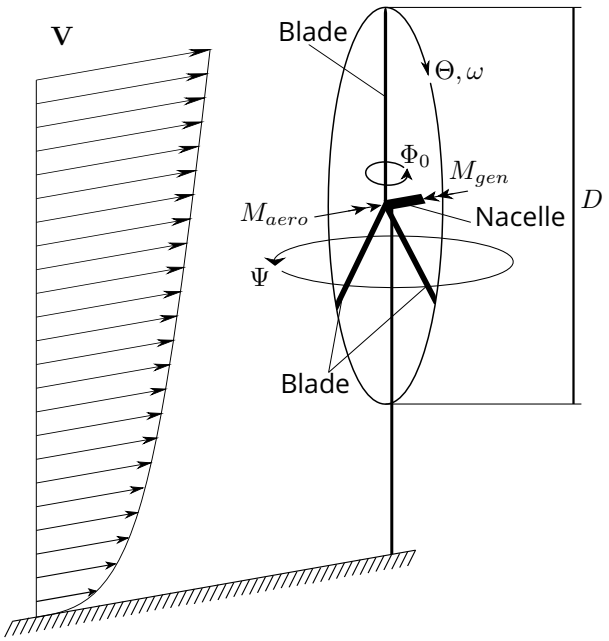

Figure 2.1.: Schematic of a HAWT in the ABL with relevant angles, lengths, velocities and moments.

The flow regime a wind turbine is subjected to is the atmospheric boundary layer (ABL), which features wind velocities of the order of $\mathcal{O}(1\,\mathrm{m/s})$ to $\mathcal{O}(10\,\mathrm{m/s})$ [30, p. 5]. Furthermore the flow is turbulent and sheared. For further information on the ABL and the challenges in modelling it, the reader is referred to [30] and [26]. Due to the nature of the ABL the wind speed is higher at larger hub heights, and a turbine with a larger diameter can produce more power since the available power P_{avail} is proportional to the area A swept by the rotor. Advances in materials science and turbine de-

sign have therefore lead to a steady increase in turbine sizes and nominal capacity [46]. Since data from real world turbines is not publicly available, the academic world uses theoretical reference turbines such as the NREL 5 MW Wind Turbine [29] or the recently proposed IEA Wind 15-Megawatt Offshore Reference Wind Turbine [12] for simulations.[22]
A turbine induces a wake downstream, characterized by an increase of turbulence intensity and a decrease of average velocity by 50 percent and more [2]. This poses a problem in wind farms, where the wake of the upstream turbines decreases the inflow velocity of the downstream turbines and thus the generated power. More details on wakes can be found in [6].

2.1.2. Wind Turbine Control

To generate power from the aerodynamic torque acting on the rotor a generator applies a countering torque. Furthermore, the turbine is equipped with different actuators, which can be used to manipulate the behaviour in order to achieve certain goals. The operating conditions of wind turbines can be classified into three regions, depending on the wind speed. In region I, below the cut-in speed, no power is generated and the wind is used to speed up the rotor. In between cut-in and rated speed lays region II, where the goal is to maximize the generated power. At rated speed, the turbine generates the maximum power. Above rated speed, in region III, the control is focussed on the quality of the generated electricity and to minimize loads on the turbine. An example of a detailed control curve can be found in [29]. [6]
The controllable variables are Ψ, Φ and M_{gen}. To maximize the power, the yaw has to be adjusted so the turbine points upwind. The pitch and generator torque have to be adjusted so that M_{gen} and ω are maximized, since the generated power P_{gen} is:

$$P_{gen} = \omega M_{gen}. \tag{2.1}$$

The blades are designed so that the optimal blade pitch is zero. At a given wind speed and constant blade pitch, it can be shown that $M_{aero} \propto \omega^2$. Applying the law of conservation of angular momentum to the rotor and generator, with I being the moment of inertia of rotor and generator, yields:

$$I\frac{\mathrm{d}\omega}{\mathrm{d}t} = M_{aero} - M_{gen}. \tag{2.2}$$

Therefore, controlling the torque in region II according to

$$M_{gen} = \kappa\omega^2 \tag{2.3}$$

maximizes P_{gen}, with κ being a proportionality constant that can be found experimentally. This control mechanism will be referred to as greedy control, since it seeks to maximize generated power

of a single turbine. [22, p.63 - 77]

2.1.3. Blade Element / Momentum Theory

To increase efficiency of wind turbines and parks as well as model their lifetime, it is necessary to study the the physical phenomena connected to wind turbines.To do so, models of the turbines and the airflow have been developed. Blade Element / Momentum Theory was developed to analyze the loads on a rotor. It combines one-dimensional momentum theory and local forces on a section of a blade, the so called blade element. Momentum theory assumes the flow to be steady, inviscid, incompressible and axisymmetric. The rotor is assumed to have an infinite number of blades and thus is equivalent to a permeable disc. It is based on the integral forms of conservation of mass, momenta and energy:

$$\oint_{\partial V} \rho \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}A = 0 \tag{2.4}$$

$$\oint_{\partial V} \rho u_x \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}A = T - \oint_{\partial V} p \mathbf{n} \cdot \mathbf{e}_x \, \mathrm{d}A \tag{2.5}$$

$$\oint_{\partial V} \rho r u_\theta \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}A = M_{aero} \tag{2.6}$$

$$\oint_{\partial V} \left(p + \frac{1}{2} \rho \|\mathbf{u}\|^2 \right) \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}A = P. \tag{2.7}$$

The surface of the control volume V is denoted as ∂V, density is denoted as ρ, the velocity vector is $\mathbf{u} = [u_x, u_r, u_\theta]$ and $\mathbf{n}$ is the normal vector pointing outwards of the control volume. T is the thrust force acting on the rotor in streamwise direction and P the power extracted by the rotor. The three dimensionless quantities tip speed ratio λ, thrust coefficient C_T and power coefficient C_P are defined as follows:

$$\lambda = \frac{\omega R}{V_0} \tag{2.8}$$

$$C_T = \frac{T}{\frac{1}{2} \rho A V_0^2} \tag{2.9}$$

$$C_P = \frac{P}{\frac{1}{2} \rho A V_0^3}, \tag{2.10}$$

with R being the radius of the rotor and and V_0 is the mean wind speed. Applying these equations to a control volume enclosed by the stream tube around the rotor disc under the assumption that $u_x = u_r = const$ in the rotor disc, yields these basic relationships for the thrust and power:

$$T = 2\rho A V_0^2 a(1-a), \quad C_T = 4a(1-a) \tag{2.11}$$

$$P = 2\rho A V_0^3 a(1-a)^2, \quad C_P = 4a(1-a)^2. \tag{2.12}$$

The axial interference factor a is a measure for the influence of the rotor disc on the velocity in the stream tube and defined as

$$a = 1 - \frac{u_r}{V_0}. \tag{2.13}$$

From (2.12) it is possible to find a maximum power coefficient, which is given by

$$C_{P,max} = \frac{16}{27} \approx 59.3\%, \quad a = \frac{1}{3}, \tag{2.14}$$

and is referred to as the Betz-Joukowski limit. In practice this limit can not be reached due to higher dimensional effects such as the rotation of the wake. Sørenensen gives an assessment of the assumptions made in [53]. [53, p. 7 - 11]

The local forces acting on the blade are calculated assuming a 2D flow around an blade element, which has the form of an airfoil. The forces and velocities in the local coordinate system of the blade are shown in Figure 2.2, with x being the global streamwise direction. The forces $\mathbf{F}_n$ and $\mathbf{F}_t$ are the forces on the blade element in normal and tangential direction, respectively, while the lift and drag forces are $\mathbf{F}_L$ and $\mathbf{F}_D$. The undisturbed wind speed is V_0, ωr is the velocity due to the rotation of the rotor and the induced velocity is defined as $\mathbf{V}_i = [-aV_0, a'\omega r]$, with axial and tangential induction factor a and a'. The sum of these velocities is the relative velocity $\mathbf{V}_{rel}$.

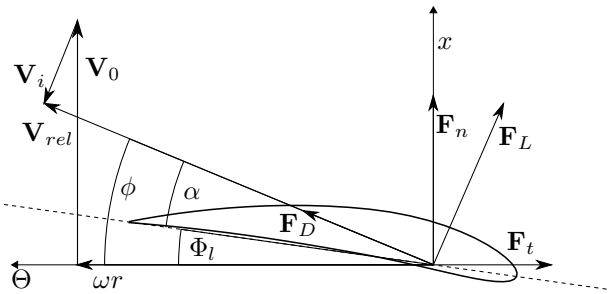

Figure 2.2.: Schematic of the local forces and velocities at a blade element.

The thrust and torque on the rotor within an infinitesimally thick stream tube are calculated as follows:

$$\frac{\mathrm{d}T}{\mathrm{d}r} = N_b F_n = \frac{1}{2} N_b \rho L_c \|\mathbf{V}_{rel}\|^2 C_n \tag{2.15}$$

$$\frac{\mathrm{d}M_{aero}}{\mathrm{d}r} = N_b r F_t = \frac{1}{2} N_b \rho L_c \|r\mathbf{V}_{rel}\|^2 C_t, \tag{2.16}$$

with N_b being the number of blades and L_c the chord length of the airfoil. The force coefficients C_n and C_t can be found by projecting lift and drag of the airfoil, which can usually be found as tabulated values, into the global coordinate system. Lift and drag depend on the local angle of attack α and the Reynolds number. The angle of attack α can be found through the difference of the angle between rotor plane and $\mathbf{V}_{rel}$, denoted as ϕ and the local pitch Φ_l. Combining (2.15) and

(2.16) with the thrust and torque found by applying momentum theory to the same stream tube yields:

$$a = \frac{1}{4\sin(\phi)^2/\left(sC_n\right) + 1} \tag{2.17}$$

$$a' = \frac{1}{4\sin(\phi)\cos(\phi)/\left(sC_t\right) - 1}, \tag{2.18}$$

with the solidity $s = N_b c/(2\pi r)$. This allows for an iterative computation of the infinitesimal torque and thrust as described in Algorithm 1. Integrating these with respect to the radius yields M_{aero} and T. In practice, the blade is defined as finite sized blade elements and the integration becomes a summation. [53, p. 100 - 103]

Many corrections for BEM have been found to increase the accuracy of the model. Among them is

Algorithm 1 Algorithm to compute thrust and torque on a blade element

for all elements **do**
  $a \leftarrow 0$
  $a' \leftarrow 0$
  repeat
    $\phi \leftarrow \tan^{-1}((1-a)/(\lambda r/R(1+a')))$
    $\alpha \leftarrow \phi - \Phi_l$
    compute C_n and C_t from lift and drag tables
    calculate new a and a' from (2.17) and (2.18)
  until a and a' converge
end for
calculate $\int \mathrm{d}T$ and $\int \mathrm{d}M$ according to (2.15) and (2.16)

the Prandtl tip loss factor, which proposes a factor to correct for the error due to the assumption of an infinite number of blades. Furthermore, above an axial induction factor of $1/3$, the assumptions made by BEM become inaccurate, as momentum theory predicts an expansion of the wake that is significantly too large. A correction for the calculation of C_t has been proposed by Glauert. Many other corrections for effects caused by the wake have also been proposed, such as a coupled near and far wake model by Pirrung et al. [42]. [53, p. 103 - 104]

2.1.4. Actuator Line Model

To study the interaction of fluid and turbine in detail, the flow field has to be resolved by means of computational fluid dynamics (CFD). Since fully resolving the shape of the blades would require a prohibitively fine resolution, the influence of the blade on the fluid is modelled. An overview over the models applied can be found in [7] and [32]. One such model is the actuator line model (ALM), which models each blade as a line of forces on the fluid. Like in BEM, the forces on the blade are calculated from local velocities and airfoil data at the points r_j along the ith actuator line $\mathbf{e}_i$. These

forces are then distributed by applying a convolution with a Gaussian filter kernel $\eta(d)$:

$$\eta(d) = \frac{1}{\epsilon^2 \pi^{3/2}} e^{(-(d/\epsilon)^2)} \tag{2.19}$$

$$\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{N_b} \int_0^R \left(\mathbf{F}_n(r) + \mathbf{F}_t(r)\right) \eta(\|\mathbf{x} - r\mathbf{e}_i\|)\, \mathrm{d}r. \tag{2.20}$$

The parameter ϵ is referred to as the smearing width, since it controls the stretching of the bell curve. As shown for example by Asmuth et. al [4], the ALM does not account for velocity induced by the root and tip vortices. This leads to an overprediction of forces on the blade, most significantly of the tangential force near the tip. Among others, Meyer-Forsting et al. have proposed corrections based on an iterative correction of the relative velocity [38]. [52]

2.2. Machine Learning for a Continuous Control Problem

2.2.1. Reinforcement Learning

Markov Decision Process

The mathematical formulation on which RL is based is the Markov Decision Process (MDP). Its two main components are the agent and the environment. Given a state $\mathbf{S}_t$ the agent takes an action $\mathbf{A}_t$. The environment responds to the action $\mathbf{A}_t$ with changing its state to $\mathbf{S}_{t+1}$ and giving feedback to the agent in form of a reward R_t. The interaction takes place at discrete time steps t and the sequence of state, action and reward is referred to as the trajectory. The dynamics of the MDP are described by the state transition function p that is defined as the probability Pr of transitioning to state $\mathbf{s}'$ and reward r given state $\mathbf{s}$ and action $\mathbf{a}$:

$$\mathsf{p}(\mathbf{s}', \mathsf{r}|\mathbf{s}, \mathbf{a}) \doteq Pr(\mathbf{S}_t = \mathbf{s}', \mathsf{R}_t = \mathsf{r}|\mathbf{S}_{t-1} = \mathbf{s}, \mathbf{A}_{t-1} = \mathbf{a}). \tag{2.21}$$

It defines the probability of state $\mathbf{s}'$ with reward r occuring, given the state $\mathbf{s}$ and action $\mathbf{a}$. Note that the following derivations will be constricted to finite MDPs, meaning that state and action space are discrete. However, the concepts all are transferable to continuous action and state space.
For a process to be a Markov Decision Process, p must only depend on $\mathbf{s}$ and $\mathbf{a}$. Therefore, $\mathbf{s}$ must include all information necessary to determine the future behaviour of the environment. This is not limited to information currently present in the environment. When thinking of this in terms of the wind farm problem at hand, the state could include data about wind speeds at the current time but also from time steps in the past. This approach allows to model virtually any interaction as a MDP, simply by including every bit of information from the beginning of time into the state. Obviously,

this is not feasible and therefore a careful choice of the information in the state is necessary.
The goal of the learning process is to maximize the sum of the rewards in the long run. Therefore a new quantity is defined, the return G_t that includes not only R_t but also the rewards received in future time steps. While in many applications of RL, the process naturally comes to an end, referred to as the terminal state $\mathbf{S}_T$, in problems of continuous control this is not the case. Therefore the timeline is broken up into episodes of length T. This allows for a finite computation of G_t. A typical formulation of G_t, referred to as a discounted return is:

$$G_t \doteq R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} ... = \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'}, \quad \gamma \in [0, 1]. \tag{2.22}$$

It includes a discount rate γ, that emphasizes rewards in the near future. If $\gamma = 0$, $G_t = R_t$, if $\gamma = 1$, the return is the sum of all future rewards. [55, p. 47- 57]
Now the goal of the learning process is defined, but not what to learn. There exist three possible answers to this question: a model of the environment, a value function or a policy. Also combinations of these components are possible. In the case of continuous control, most common approaches are model-free, meaning either learning a value function, a policy or both. Therefore model-based methods will not be discussed further and the reader is referred to the book by Sutton and Barto [55].
In loose terms, the policy π guides the agent on which action to take. More formally, $\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})$ is the probability of action $\mathbf{A}_t = \mathbf{a}$ given a state $\mathbf{S}_t = \mathbf{s}$ under the policy π with its parameters set to $\boldsymbol{\theta}$. The value function of a state $\mathbf{s}$ under a policy π is denoted as $v_\pi(\mathbf{s})$ and is the expected return if the agent acts according to π, starting from state $\mathbf{s}$. Note that for convenience the parameters of π were dropped. For MDPs it can be defined as:

$$v_\pi(\mathbf{s}) \doteq \mathrm{E}_\pi \left[G_t | \mathbf{S}_t = \mathbf{s} \right] = \mathrm{E}_\pi \left[\sum_{t'=t}^{T} \gamma^{t'-t} R_{t'} | \mathbf{S}_t = \mathbf{s} \right] \tag{2.23}$$

$$= \sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}) \sum_{\mathbf{s}'} \sum_{r} p(\mathbf{s}', r|\mathbf{s}, \mathbf{a}) \left(r + \gamma v_\pi(\mathbf{s}') \right). \tag{2.24}$$

In the form of (2.24), the equation is referred to as the Bellmann equation and its unique solution is the value function v_π. Analogously, the action-value function is the expected reward of taking action $\mathbf{a}$ at state $\mathbf{s}$ under the policy π, denoted by $q_\pi(\mathbf{s}, \mathbf{a})$. It is defined by [55, p. 58-59]:

$$q_\pi(\mathbf{s}, \mathbf{a}) \doteq \mathrm{E}_\pi \left[G_t | \mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a} \right] = \mathrm{E}_\pi \left[\sum_{t'=t}^{T} \gamma^{t'-t} R_{t'} | \mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a} \right]. \tag{2.25}$$

Policy Gradient Methods

With a known value function, it is possible to construct a policy and by improving the value function, the policy can be improved. However, there exist advantages to directly improving the policy without determining the value function, especially for continuous state and action spaces. Policy gradient methods use the gradient of a performance measure $\mathsf{J}(\boldsymbol{\theta})$ with respect to the parameters $\boldsymbol{\theta}$ of a policy. This gradient can be used in optimization algorithms, such as stochastic gradient descent (SGD) [55, p. 201] or its extensions such as Adam [33]. In its basic form, SGD performs an update according to:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \psi \nabla_{\boldsymbol{\theta}} \mathsf{J}(\boldsymbol{\theta}). \tag{2.26}$$

The parameter ψ is called the learning rate. The policy gradient methods differ now only in the performance measure. The first such algorithm proposed is the REINFORCE algorithm [57]. Its definition of $\nabla_{\boldsymbol{\theta}} \mathsf{J}(\boldsymbol{\theta})$ is:

$$\nabla_{\boldsymbol{\theta}} \mathsf{J}(\boldsymbol{\theta}) = \mathrm{E}_{\pi}\left[\sum_{\mathbf{a}} \pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{S}_t) \mathsf{q}_{\pi}(\mathbf{S}_t, \mathbf{a}) \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{S}_t)}{\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{S}_t)}\right] \tag{2.27}$$

$$= \mathrm{E}_{\pi}\left[\mathsf{G}_t \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\mathbf{A}_t|\mathbf{S}_t)}{\pi_{\boldsymbol{\theta}}(\mathbf{A}_t|\mathbf{S}_t)}\right] \tag{2.28}$$

$$\approx \frac{1}{N_{e,b}} \sum^{N_{e,b}} \frac{1}{\mathsf{T}} \sum_{t}^{\mathsf{T}} \mathsf{G}_t \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\mathbf{A}_t|\mathbf{S}_t)}{\pi_{\boldsymbol{\theta}}(\mathbf{A}_t|\mathbf{S}_t)}. \tag{2.29}$$

It follows from the policy gradient theorem and substitution of all values of **A** and **S** with the actions and states from one trajectory. Thus, all the values necessary for the computation of the gradient are known. The expected value can approximated by the mean of a batch of episodes with $N_{e,b}$ being the number of episodes per batch. [55, p.324-328]
While this algorithm makes a computation possible, it is inefficient. Therefore newer methods have been devised such as the Trust Region Policy Optimization (TRPO) [48] and the Proximal Policy Optimization (PPO) [49]. They are closely related and both propose a surrogate performance measure that limits the size of the gradient to ensure monotonic improvement in the case of TRPO and a close approximate in the case of PPO. However, the computation of the surrogate in PPO is simpler and more efficient, which helped policy gradient methods to become one of the most used algorithms in continuous control problems. One version of the surrogate performance measure, which is also referred to as objective, is:

$$\mathsf{pr}_t(\boldsymbol{\theta}) \doteq \frac{\pi_{\boldsymbol{\theta}}(\mathbf{A}_t|\mathbf{S}_t)}{\pi_{\boldsymbol{\theta}_{old}}(\mathbf{A}_t|\mathbf{S}_t)} \tag{2.30}$$

$$\mathsf{J} \doteq \mathrm{E}_{\pi}\Big[\min\Big(\mathsf{pr}_t(\boldsymbol{\theta})\mathsf{a}_{\pi}(\mathbf{A}_t|\mathbf{S}_t), \mathrm{clip}(\mathsf{pr}_t(\boldsymbol{\theta}), 1-\beta, 1+\beta)\mathsf{a}_{\pi}(\mathbf{A}_t|\mathbf{S}_t)\Big)\Big]. \tag{2.31}$$

The probability ratio pr_t compares the policy after the update to the policy before the update. Therefore $\pi_{\boldsymbol{\theta}}$ has to be approximated. $\mathsf{a}_\pi(\mathbf{A}_t|\mathbf{S}_t)$ is an estimator of the advantage function, which is defined as $\mathsf{a}_\pi(\mathbf{A}_t|\mathbf{S}_t) = \mathsf{q}_\pi(\mathbf{A}_t|\mathbf{S}_t) - \mathsf{v}_\pi(\mathbf{S}_t)$. This estimator also has to be found, however, this is a regular optimization problem, which can be solved by use of SGD or Adam. The parameter β is referred to as the clipping ratio. [49]
In addition to the ratio probability clipping, other regularizations can be added to the objective function, for example an l_2 regularization, that adds a penalty proportional to $\|\boldsymbol{\theta}\|^2$.

2.2.2. Artificial Neural Networks

Feed-forward networks

In the section above, a way to update parameters of the policy or value function was described, however, no description of the policy function itself was given. In principal any function with a set of parameters can be used, but usually, an artificial neural network (ANN) is used. ANNs are comprised of layers of neurons. More precisely, a layer k of N neurons takes as input a vector $\mathbf{z}$ of length M. The output of the layer is a new vector $\mathbf{y}^{\mathsf{k}}$, which is then the input for the next layer of the network. In a simple feed-forward layer $\mathsf{y}_j^{\mathsf{k}}$ is computed according to:

$$\mathsf{y}_j^{\mathsf{k}} = \mathsf{f}(\mathsf{w}_{i,j}^{\mathsf{k}}\, \mathsf{z}_i^{\mathsf{k}} + \mathsf{b}_j^{\mathsf{k}}). \tag{2.32}$$

The entries $\mathsf{w}_{i,j}^{\mathsf{k}}$ of the matrix $\mathbf{W}^{\mathsf{k}}$ are called the weights of the layer and the entries $\mathsf{b}_j^{\mathsf{k}}$ of the vector $\mathbf{b}^{\mathsf{k}}$ are called its bias. f is called the activation function, which can be chosen freely, but typical choices include the hyperbolic tangent ($\tanh$), the sigmoid-function (σ) or the softplus ($\mathrm{softplus}$) function. A small overview over some of the key features of these functions is given in Table 2.1. One desirable property for an activation function is that it is differentiable at least once in the entire domain. Thus a network of two layers with $\tanh$ as an activation function describes the function

$$\mathbf{y} = \mathbf{tanh}(\mathbf{b^1} + \mathbf{W^1} \cdot \mathbf{tanh}(\mathbf{b^0} + \mathbf{W^0} \cdot \mathbf{z})), \tag{2.33}$$

with $\mathbf{z}$ being the input to the network and $\mathbf{y}$ its output and the activation function being applied element-wise to the vectors. [9, p. 2-2 - 2-12].

Recurrent neural networks

It is obvious that a feed-forward network only produces an output influenced by the current activation. There can be no influence of previous activations for example in a time-series. Yet there exist many applications for which such a feature would be useful, for example the field of natural

Table 2.1.: Activation functions commonly used in neural networks

Name	Domain	Range	Definition	Derivative
$\tanh$	$(-\infty, +\infty)$	$(-1, -1)$	$\tanh(x) = (e^x - e^{-x})(e^x + e^{-x})^{-1}$	$1 - \tanh^2(x)$
σ	$(-\infty, +\infty)$	$(0, -1)$	$\sigma(x) = (1 + e^{-x})^{-1}$	$\sigma(x)\sigma(-x)$
softplus	$(-\infty, +\infty)$	$(0, +\infty)$	$\text{softplus}(x) = \ln(e^x + 1)$	$\sigma(x)$

language processing [19] or transient physical processes. This led to the development of recurrent neural networks. These include a hidden state, which is preserved. Especially the development of the long short-term memory (LSTM) cell has had great impact on the success of recurrent neural networks due to its ability to preserve hidden states over a longer period of time [23]. An LSTM layer consists of one or multiple cells, with the input being the entire or parts of the activation. Each cell passes a vector called the cell state $\mathbf{c}$ as well as its output $\mathbf{y}$ to itself in the next time step. Thus a cell has as inputs at time t the cell state and output of the previous time step, $\mathbf{c}_{t-1}$ and $\mathbf{y}_{t-1}$, as well as the current activation $\mathbf{z}_t$. On the inside, the cell consists of a forget-gate, an input-gate and an output-gate. The forget-gate determines the influence of the cell-state, the input-gate determines the influence of the current time-step on the cell-state. The output-gate then computes the output based on the updated cell-state. A schematic of the cell is given in Figure 2.3. From left to right the sigmoid layers are the forget-, input- and output-gate, whereas the $\tanh$-layer corresponds most closely to the layer of a feed-forward-network.

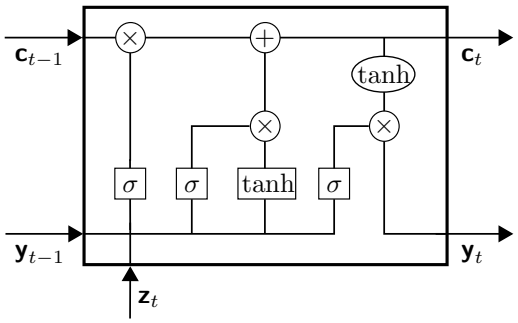

Figure 2.3.: Schematic representation of an LSTM cell, operation in boxes represent layers with trainable weights, operations in ellipses are pointwise operations.

Deep Reinforcement Learning

If RL is combined with a neural network, it is often referred to as deep reinforcement learning or DRL. In case of a policy gradient method, the neural network is part of the policy. For example, the state can be used as the input of a neural network and the output of the network can be used to set the parameters of a distribution function. Assuming that each entry in the action vector $\mathbf{A}$ is a

statistically independent random variable with the probability density function pdf, parametrized by μ_i, the policy can be written as

$$\pi_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{S}) = \prod_i pdf(\mathsf{A}_i, \mu_i(\mathbf{S}, \boldsymbol{\theta})). \tag{2.34}$$

If μ_i is the output of a K-layered network with activation function f, it is:

$$\mu_i(\mathbf{S}, \boldsymbol{\theta}) = \mathsf{f}^{\mathsf{K}}(\mathsf{b}_i^{\mathsf{K}} + \mathsf{w}_{i,j}^{\mathsf{K}} \mathsf{f}^{\mathsf{K}-1}(...\mathsf{f}^0(\mathsf{b}_k^0 + \mathsf{w}_{k,l}^0 \mathsf{S}_l)...)). \tag{2.35}$$

Then the parameters $\boldsymbol{\theta}$ of the policy $\pi_{\boldsymbol{\theta}}$ are the weights and biases of the neural network:

$$\boldsymbol{\theta} = \left[\mathsf{b}_j^{\mathsf{k}}, \mathsf{w}_{i,j}^{\mathsf{k}}\right]^T. \tag{2.36}$$

A schematic of DRL with a policy gradient method is given in Figure 2.4.

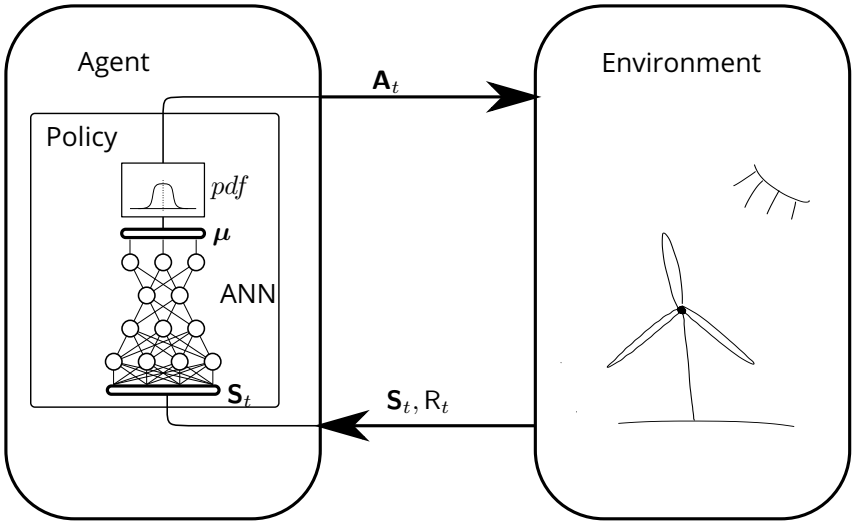

Figure 2.4.: Schematic of a Markov decision process for a policy gradient method with a neural network.

Training of a Neural Network

To use a neural network as policy in a policy gradient method algorithm, a way to compute the gradient of the objective J has to be found. The parameters of the policy are the weights and biases of the network, therefore the partial derivatives of J with respect to all weights and biases have to be found. Both described policy gradient methods express the objective gradient as a function j of the gradient of the policy:

$$\nabla_{\boldsymbol{\theta}} \mathsf{J} = \mathsf{j}\left(\frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{S})}{\pi_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{S})}\right). \tag{2.37}$$

Under the same assumptions as above, the gradient term is: [55, p. 335]

$$\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{S})}{\pi_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{S})} = \prod_i \frac{\partial pdf(\mathsf{A}_i, \mu_i(\mathbf{S},\boldsymbol{\theta}))}{\partial \mu_i(\mathbf{S},\boldsymbol{\theta})} \frac{\nabla_{\boldsymbol{\theta}}\mu_i(\mathbf{S},\boldsymbol{\theta})}{pdf(\mathsf{A}_i, \mu_i(\mathbf{S},\boldsymbol{\theta}))}. \tag{2.38}$$

Computing the gradient

$$\nabla_{\boldsymbol{\theta}}\mu_i|_{\mathsf{S}} = \left[\left.\frac{\partial \mu_i}{\partial \mathsf{b}_j^{\mathsf{k}}}\right|_{\mathsf{S}}, \left.\frac{\partial \mu_i}{\partial \mathsf{w}_{j,k}^{\mathsf{k}}}\right|_{\mathsf{S}} \right]^T \tag{2.39}$$

can now be done efficiently via backpropagation:

$$\mathsf{x}_i^{\mathsf{k}} = \mathsf{b}_i^{\mathsf{k}} + \mathsf{w}_{i,j}^{\mathsf{k}}\mathsf{z}_j^{\mathsf{k}}, \quad \mathsf{z}_i^{\mathsf{k}+1} = \mathsf{f}^{\mathsf{k}}(\mathsf{x}_i^{\mathsf{k}}), \quad \mathsf{z}_i^0 = \mathsf{S}_i \tag{2.40}$$

$$\mathsf{t}_{i,j}^{\mathsf{K}} = \left.\frac{\partial f^{\mathsf{K}}(x)}{\partial x}\right|_{\mathsf{x}_i^{\mathsf{K}}} \delta_{i,j}, \quad \mathsf{t}_{i,j}^{\mathsf{k}} = \mathsf{t}_{i,k}^{\mathsf{k}+1}\mathsf{w}_{k,j}^{\mathsf{k}} \left.\frac{\partial f^{\mathsf{k}}(x)}{\partial x}\right|_{\mathsf{x}_i^{\mathsf{k}}} \tag{2.41}$$

$$\left.\frac{\partial \mu_i}{\partial \mathsf{b}_j^{\mathsf{k}}}\right|_{\mathsf{S}} = \mathsf{t}_{i,j}^{\mathsf{k}}, \quad \left.\frac{\partial \mu_i}{\partial \mathsf{w}_{j,k}^{\mathsf{k}}}\right|_{\mathsf{S}} = \mathsf{t}_{i,j}^{\mathsf{k}}\mathsf{z}_k^{\mathsf{k}} \tag{2.42}$$

The term backpropagation refers to the fact, that due to the chain rule, the gradient of a layer is easily expressed through the gradient of the previous layer, which allows for an efficient computation. The sensitivity $\mathsf{t}_{i,j}^{\mathsf{k}}$ determines how sensible the action is to changes of this parameter. [9, p.11-7 - 11-13]

2.3. The Lattice Boltzmann Method

2.3.1. The Boltzmann Equation

The agent described in the section above needs to interact with an environment in order to train. In the case studied in this thesis the environment is a wind farm with multiple turbines. Therefore the fluid field within the wind park has to be calculated. Traditionally solvers based on the discretized Navier-Stokes equations (NSE) are used. However, parallelization has proven to be difficult. A newer approach is the lattice Boltzmann method (LBM). The basics of its theory will be laid out in this chapter.

LBM is based on a discretization of the Boltzmann equation derived from the kinetic theory of gases. This description is often referred to as a mesoscopic description, since it stands in between the scale of continuum theory and the scale of single particles, which will be referred to as macroscopic and microscopic scale, respectively. While the NSE describe the medium through density ρ, velocity $\mathbf{u}$ and pressure p, the Boltzmann equation considers the particle density function (pdf) f, which describes the distribution of particles in six dimensional phase space. It is therefore a function of space $\mathbf{x}$, microscopic velocity $\boldsymbol{\xi} = [\xi, \upsilon, \zeta]^T$ and time t. However, the macroscopic quantities can be

recovered from the pdf by taking its zeroth and first moments in velocity space:

$$\rho(\mathbf{x},t) = \int_{-\infty}^{\infty} f(\mathbf{x},\boldsymbol{\xi},t)\mathrm{d}\boldsymbol{\xi} \tag{2.43}$$

$$\rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t) = \int_{-\infty}^{\infty} \boldsymbol{\xi} f(\mathbf{x},\boldsymbol{\xi},t)\mathrm{d}\boldsymbol{\xi}. \tag{2.44}$$

Similarly the energy can be found by the third moment. The pressure can not be recovered directly, instead it is calculated by a state equation such as the ideal gas law. The pdf tends towards an equilibrium, which is given by the Maxwell equilibrium. . The Boltzmann equation describes the change of the pdf over time. The change in time is governed by the collision operator Ω. It describes the change of f due to collisions of particles. In the Boltzmann equation it is an integral operator, that accounts for all possible collisions. Therefore it is mathematically rather cumbersome which renders it not useful for numerical discretization. Thus another formulation for Ω is used in LBM, which is an active area of research in the LBM community [8]. The Boltzmann equation is:

$$\frac{Df(\mathbf{x},\boldsymbol{\xi},t)}{Dt} = \frac{\partial f(\mathbf{x},\boldsymbol{\xi},t)}{\partial t} + \frac{\partial f(\mathbf{x},\boldsymbol{\xi},t)}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial t} + \frac{\partial f(\mathbf{x},\boldsymbol{\xi},t)}{\partial \boldsymbol{\xi}}\frac{\partial \boldsymbol{\xi}}{\partial t} = \Omega(f). \tag{2.45}$$

The first form is the total derivative of f with respect to time. The second term in the second form is a convection term, while the third term corresponds to a forcing term. In comparison to the NSE the Boltzmann equation lacks a diffusive term. Diffusion occurs only through the collision operator. Through collision the particles tend towards equilibrium, the pdf of that equilibrium is known as the Maxwell equilibrium. It can be described by macroscopic quantities only:

$$f^{eq}(\mathbf{x},\boldsymbol{\xi},t) = \rho\left(\frac{1}{2\pi\hat{R}\hat{T}}\right)e^{-\|\mathbf{v}\|^2/(2\hat{R}\hat{T})}, \tag{2.46}$$

with $\hat{R}$ being the specific gas constant and $\hat{T}$ the temperature. The velocity $\mathbf{v}$ is defined as $\mathbf{v} = \boldsymbol{\xi} - \mathbf{u}$. [34, p. 15- 21]

2.3.2. Discretization

In order to use the Boltzmann equation for numerical simulations it needs to be discretized in space, velocity and time and a collision operator has to be defined. The velocity discretization can be based on the Hermite Series expansion, since its generating function is of the same form as the equilibrium distribution. To recover the first three moments of the distribution correctly, i.e. density, velocity and energy, truncation of the series after the first three terms is sufficient. The roots of the Hermite polynomials up to order three are the necessary discrete velocities $\boldsymbol{\xi}_i$. Together with the weights b_i, that are used in the Gauss-Hermite quadrature, the velocities form a velocity set. Velocity sets are

denoted in the DdQq-notation, with d being the number of spatial dimensions and q the number of discrete velocities. In three dimensions D3Q15, D3Q19 and D3Q27 can be used to recover the Navier-Stokes equations, for high Reynolds number flows D3Q27 is the most suitable[31]. [34, p. 73-93]
Time is discretized via the explicit Euler forward scheme, which can be shown to be second order accurate. Space is discretized uniformly into a cubic grid, so that discrete pdfs move from one node of the grid to the other in one time step. The ratio of time step to lattice width is called the lattice speed of sound c_s. The computation of a time step is usually separated into two parts, the streaming step, in which populations are advected from one node to the other, and the collision step, in which the collision operator is applied. Applying the entire discretization gives the lattice Boltzmann equation (LBE), with $\mathbf{c}_i$ being $\boldsymbol{\xi}_i/\sqrt{3}$ for convenience:

$$f_i(\mathbf{x} + \mathbf{c}_i\Delta t, \mathbf{c}_i, t + \Delta t) = \Omega_i(f) + f_i(\mathbf{x}_i, \mathbf{c}_i, t). \tag{2.47}$$

It is, compared to discretized NSE, very simple and the separation into collision and streaming makes it easily parallelizable. Furthermore the equations to compute density and velocity are: [34, p. 94-98]

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t) \tag{2.48}$$

$$\rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \sum_i \mathbf{c}_i f_i(\mathbf{x}, t) \tag{2.49}$$

Lastly a collision operator has to be found. The first applicable collision operator proposed was the Bhatnager-Gross-Krook (BGK) operator:

$$\Omega_i^{BGK} = \hat{\omega}\left(f_i - f_i^{eq}\right). \tag{2.50}$$

It is based on the fact, that the distributions tend to the equilibrium distribution, therefore the BGK operator relaxes the distributions towards equilibrium with a constant relaxation frequency $\hat{\omega}$. Via Chapmann-Enskog analysis it can be shown that this relaxation frequency is related to the kinetic viscosity ν: [34, p. 98-100, 112]

$$\nu = c_s^2\left(\frac{1}{\hat{\omega}} - \frac{\Delta t}{2}\right). \tag{2.51}$$

While the BGK operator is sufficient for low Reynolds number flows, it becomes unstable at higher Reynolds numbers. Therefore more sophisticated methods had to be developed. One approach is to transform the distributions into moment space and relax the moments independently, leading to multiple relaxation time (MRT) methods. Geier et al. argue, that they cannot be relaxed separately, since these moments are not statistically independent. However, cumulants of a distribution are

statistically independent by design, therefore they can also be relaxed independently. Furthermore, they fulfil Galilean invariance, which is not always the case for MRT methods. It can be shown that with a parametrization this method can be fourth order accurate[16]. In under-resolved flows, the parameter of this parametrization can be used to influence the numerical diffusivity of the cumulant operator, acting like an implicit sub-grid-scale model for Large-Eddy-Simulations (LES)[3]. However, the exact behaviour of the under-resolved cumulant operator is not yet known. More details on the cumulant LBM as well as a derivation for a refinement algorithm can be found in Appendix A.
While LBM has some advantages over NSE-based algorithms, the treatment of boundary conditions is often more complicated. This is due to the fact that it is necessary to prescribe the populations in the boundary nodes. No-slip and full slip boundaries are imposed by bounce-back and bounce forward, respectively and velocity boundary conditions can be imposed by prescribing the corresponding equilibrium distributions or by extending the bounce back approach [34, p. 175 - 189, 199 – 207]. Another problem often arising in LBM is the reflection of acoustic waves, especially at inlet and outlet boundaries. There exist methods to cancel these waves, while another approach is to simply increase the viscosity in a so called sponge layer near the outlet [34, p. 522 - 526].

2.4. Previous Works

2.4.1. Wind Farm Control

In subsection 2.1.2 the greedy control strategy for a single turbine was described. However, this control strategy does not take into consideration the effects of the wake on other turbines within a wind farm. Therefore more sophisticated strategies with the objective of maximizing the generated power of multiple turbines or a whole wind farm have been proposed. They can be divided into two main categories, axial induction control and wake redirection control [6]. Axial induction control tries to lower the power intake of the first turbine, so that the wind speed at the following turbines is higher. Wake redirection control aims at steering the wake away from the next turbine downstream. A fairly recent review of the vast amount of studies dedicated to this topic can be found in [32].
Overall, it shows that static axial induction control is able to increase total power, in LES-simulations usually around 5 percent. However, the magnitude of the increase depends on the simulation method and flow conditions. Aerodynamic phenomena not captured by BEM-like models can significantly decrease the generated power. An increase in turbulence intensity also leads to a decrease in power. This highlights the importance of LES simulations for studies on control. Compared to the static approaches, dynamic approaches showed to be more promising. Based on a receding-horizon optimal control, Goit and Meyers [20] were able to report an increase in power of 16 percent.

However, this approach requires knowledge of the full flow field as well as adjoint simulations and thus not applicable to real control. Based on these results, a simplified control strategy based on sinusoidal variation of the induction was proposed by Munters und Meyers [39]. This approach was further explored by Frederik et. al [10], leading to the development of the helix approach, based on a periodic variation of pitch angles. This results in a sinusoidal moment exerted on the wake and thus increasing wake mixing. They could report an increase in power of up to 7.5 percent. These results prove the theoretical potential for power increase by dynamic axial induction control and that simple control strategies can achieve considerable gains.
Static and dynamic wake redirection have also been explored, and the review in [32] concludes that it shows an even greater potential for improving power. However, this work will only focus on induction control.

2.4.2. Machine Learning in Active Flow Control

Formulating control strategies for active flow is inherently difficult, due to the non-linear nature of fluid-flow. This motivates the interest in methods of machine learning, since it has been shown in other areas that these methods can develop new strategies and deal with large state spaces, most famously in games such as Go [50], but also . Recently, some studies applying reinforcement learning have been conducted. A review of some of them can be found in [13]. The work by Verma et al. focussed on collectively swimming fishes and employed Q-learning, in which the action-value function is approximated by a neural network [56]. Rabault et al. were the first ones to apply policy gradient methods. They used it to minimize the drag on a two-dimensional von-Kármán vortex street via jets [44]. It showed the applicability of these methods to active flow control. In subsequent studies, which focussed on similar problems, i.e. two-dimensional laminar flow, more aspects of the application of DRL to flow control were studied. The use of parallel environments to reduce the wall time was proposed in [43]. In [5], policy gradient methods were successfully applied to control of a one-dimensional, unstable falling liquid film. While these studies showed promising results, the complexity of the controlled flows was still limited in comparison to a fully turbulent three-dimensional LES simulation.

3. Setup of Simulation

3.1. Implementation

3.1.1. Implementation Strategy

In this work the performance of two controllers in two simulation environments is studied. A greedy controller is implemented to be used as a baseline case. This is compared to a controller that is governed by an RL-agent. The first simulation environment which is used is a one-dimensional BEM model of a single turbine, which was implemented for this work. As it is simple yet fast, it is mainly used for testing of the implementation of the controllers. Furthermore, a parameter study of some of the parameters of the RL-agent is conducted in this environment. The second environment is an LBM-ALM environment with multiple turbines. The majority of the work is implemented in Python with the exception of the LBM-ALM, which is implemented in C++.
In contrast to most other applications of machine learning, in this case the main program is not the one governing the machine learning, but rather the simulation environment. This is due to the fact that the LBM-ALM simulation environment already existed prior to this work and it was expected to be simpler to design the RL-controller so that it could be called by the simulation environment, rather than breaking up the already existing code. Furthermore the LBM-ALM consumes the vast amount of computational time, so any potential for optimization of computational effort in this part should be used. However, this approach also created some difficulties, especially regarding the concurrency of events and scope of objects, many of which were solved by using techniques of object oriented programming. In the following, the different parts of the code developed for this work will be explained.

3.1.2. RL-Controller

The RL-Controller implements an MDP, with an environment and an agent and also governs the interaction between them. It relies on the library TF-Agents [21], which implements many RL algorithms and is based on the Python API of TensorFlow [1]. The frequency of interactions is not related to the size of the time step of the simulation environment, since the relevant time scales are not necessarily connected [44]. It is governed by a parameter, the number of time steps per action, $N_{t,A}$.
An instance of the class `PPOAgent` represents the agent and contains the methods to calculate actions based on the observations and to train the agent. It is based on the PPO-algorithm explained in 2.2.1. The actor network consists of three layers, of which the first layer is either a fully connected

LSTM cell or a regular feed forward layers. The second layer is simply a feed-forward layer. The activation function of these layers is $\tanh$. They have the same width, which is adjustable. The last layer is a feed forward layer of $\mathrm{softplus}$ nodes or $\tanh$, depending on the controlled quantity. The value network is a three layered RNN with a $\mathrm{softplus}$ activation.
The environment is implemented in the class `TurbineEnvironment`. The action **a** provided by the agent is smoothed with an exponential filter

$$\hat{\mathbf{a}}_t = \hat{\mathbf{a}}_{t-1} + \alpha_f(\mathbf{a}_t - \hat{\mathbf{a}}_{t-1}) \tag{3.1}$$

to make it continuous as done for example in [44]. The decay rate α_f is $\alpha_f = r_f^{1/N_{t,A}}$ with a forget ratio r_f of 0.99. The smoothed action $\hat{\mathbf{a}}$ is applied as control of the turbine. To allow for a more refined action space, the value of the control variables is calculated by

$$\varphi(t) = \varphi_{base} + \sum_{i=0}^{N_a} A_{\varphi,i} \cos(2\pi f_{\phi,i} t), \tag{3.2}$$

with φ being the controllable quantity, φ_{base} is a base value that is modulated by N_a cosine waves with frequencies $f_{\varphi,i}$ and amplitudes $A_{\phi,i}$. Unless controlled by the agent, these quantities are set according to the greedy control strategy, that is all amplitudes and base values are zero except the base-value of M_{gen}.
The observations and reward are also filtered exponentially, with the same rate as the action. Observations can include quantities from turbines, such as ω, but also velocities sampled from the flow-field. They can also be from multiple past time steps in addition to the current time step. All quantities in the environment are also nondimensionalized so that their magnitude is $\mathcal{O}(1)$. For each quantity a value range in SI-units as well as in the dimensionless environment is defined. The quantities are mapped via an affine transformation from one range to the other. This is especially relevant for the generator torque. If it is used as the quantity controllable by the agent, it must be in a range that is physically reasonable. Since this range is not the same for all turbines in a park, the maximum value of the generator torque at turbine i is defined as

$$M_{gen,max} = \frac{1}{1.5 + 1.3i} M_{max}, \tag{3.3}$$

with M_{max} the maximum allowed generator torque according to the NREL 5MW reference [29]. These values were found experimentally. If the environment reaches a terminal state, because a rotor turns too slow or too fast, the environment is controlled by a greedy controller, until it is in a stable state again. This was implemented since a reinitialization of the LBM-ALM simulation environment would be very costly. To reduce wall time, multiple environments can be run at the same time, each represented by a single instance of `TurbineEnvironment`. Therefore, all the `TurbineEnvironment`s are wrapped in a single instance of a `ParallelTurbineEnvironment`.

3.1.3. Greedy Controller

The greedy controller is an implementation of the baseline generator-torque controller given in [29], which is based on the greedy control strategy explained in 2.1.2. To simplify the calculations the controllable torque is the torque at the rotor, losses in generated power due to the transmission are ignored, since they are irrelevant for this work. The proportionality constant κ was optimized using the BEM environment and found to be $1.268 \times 10^6\,\mathrm{kgm}^2$.

3.1.4. LBM-ALM Environment

The LBM-ALM environment is based on the work of Asmuth et al. [3, 4]. The actuator line model is implemented in elbe, a cumulant LBM code utilizing graphical processing units (GPU) [28, 15]. For this work the code was extended to include a second order accurate refinement scheme according to Schönherr et al. [47], for further information the reader is referred to section A.2. Multiple independent domains can be instantiated in one call to the main function, with creating only one instance of the controller, allowing for a significant reduction in wall time [43]. The inlet is a uniform inflow, that can be superimposed with fluctuations of a Mann box [37], created by a synthetic turbulence generator. The outlet uses a sponge layer, as described in section 2.3.

3.1.5. Fast Implimentation of a BEM Environment

To provide the physical inputs of a one-dimensional model of the flow field and the turbine a steady state BEM-code for the computation of C_t and C_n was implemented. To minimize computational time, a table of 400 values of M_{aero} is precomputed and then linearly interpolated. The fluid field is composed of a base velocity and an optionally superimposed turbulent fluctuation of a Mann box, which is computed by a synthetic turbulence generator. The base velocity can also be varying in time, either by defining a sine wave or by regularly sampling a random velocity within a given interval.

3.1.6. Interaction of Controller and Simulation Environment

To provide a layer of abstraction between simulation environments and controllers, exchange of information takes place via the so-called visitors. Each turbine and probe is represented by an instance of a visitor. It is provided with input by the simulation environments such as M_{aero} or the

velocity, but can also set control variables such as M_{gen}. This allows for an easy recombination of simulation environments and controllers.

As an example for the interaction between the controller and the simulation environment, a time

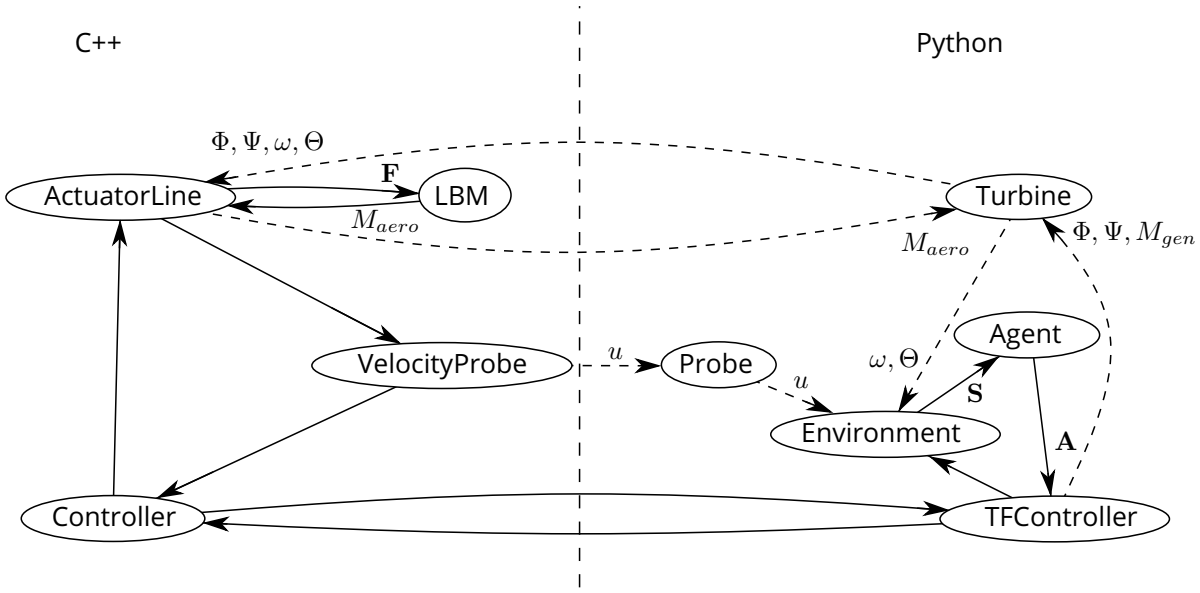

Figure 3.1.: Schematic of a time step sequence. To trace order of execution, follow solid arrows and go around the ellipses clockwise, dashed arrows represent exchange of information. Information is exchanged when execution meets the dashed lines.

step of a simulation with the ALM-LBM and a RL-Controller is given in 3.1. It shows that the probes and turbines hold information but not execute methods themselves. It also shows the implementation languages. To simplify communication across languages, objects that exchange information across languages are mirrored in both languages. This includes the visitors as well as the controller.

3.2. Setup

3.2.1. Preliminary Studies in the BEM environment

To validate the implementation of the controllers and do studies of some of the hyperparameters of the agent the BEM environment is used. The airfoil data and physical properties are taken from the NREL 5MW reference turbine [29]. The mean wind speed is $V_0 = 10.5\,\mathrm{m/s}$ and a velocity probe is placed $10.5\,\mathrm{m}$ upstream of the turbine. The control variable is M_{gen}. The state of the environment consists of the angular velocity of the turbine and the streamwise velocity of the velocity probe. Training is conducted in six environments in parallel, with independently generated turbulent in-

flow. The size of the time step of the simulation is $\Delta t = 0.1\,\mathrm{s}$. A parameter study of the following parameters is conducted:

- variance of action σ_A
- policy l_2 regularization coefficient ε_{p,l_2}
- value l_2 regularization coefficient ε_{v,l_2}

These parameters were chosen since they do not directly depend on the physics and layout of the problem. The parameters not varied are set according to Table 3.1.

Table 3.1.: Parameters of agent and environment

Parameter	Symbol	Value
Time per action	Δt_A	$1.0\,\mathrm{s}$
Number of actions per episode	$N_{A,e}$	500
Number of episodes per batch	$N_{e,b}$	12
Width of actor network		100
Width of value network		200
Variance of action	σ_A	0.1
Importance ratio clipping	β	0.2
Discount factor	γ	0.95
Policy l_2 regularization coefficient	ε_{p,l_2}	0.1
Value l_2 regularization coefficient	ε_{v,l_2}	0.1
Value network loss coefficient	$\varepsilon_{v,l}$	0.5
Learning rate	ψ	10^{-3}

3.2.2. The LBM-ALM Environment

The properties of the turbine are again taken from the NREL5 reference [29]. The diameter D of the rotor is therefore $D = 126\,\mathrm{m}$. Three turbines are placed in the center of the cross-stream plane with a distance of five diameters in streamwise direction. The first turbine is placed three diameters downstream of the inlet. The domain is 19 diameters long and six diameters wide and high. In order to reduce computational cost, a more refined domain is placed within the first domain. It is placed in the center of the crosswise plane and one diameter downstream of the inlet. 130 velocity probes are placed in the shape of crosses upstream and between the turbines with a distance of one diameter. The layout is shown in Figure 3.2. The inflow is turbulent with a turbulence intensity TI of 5 percent calculated as $TI = \sqrt{\frac{1}{3}(u'^2 + v'^2 + w'^2)}/V_0$. The parameters of the domain are gathered in Table 3.2. Training of the agents is again conducted in six environments in parallel.

Table 3.2.: Parameters of the domain

Parameter	Value
Rotor diameter	$126\,\mathrm{m}$
Outer domain H x W x L	$6\,\mathrm{D} \times 6\,\mathrm{D} \times 19\,\mathrm{D}$
Outer domain resolution	$8\,\mathrm{nodes/D}$
Inner domain H x W x L	$5\,\mathrm{D} \times 5\,\mathrm{D} \times 16\,\mathrm{D}$
Inner domain resolution	$16\,\mathrm{nodes/D}$
Turbulence intensity	$5\,\%$
Mean wind speed	$10.5\,\mathrm{m/s}$

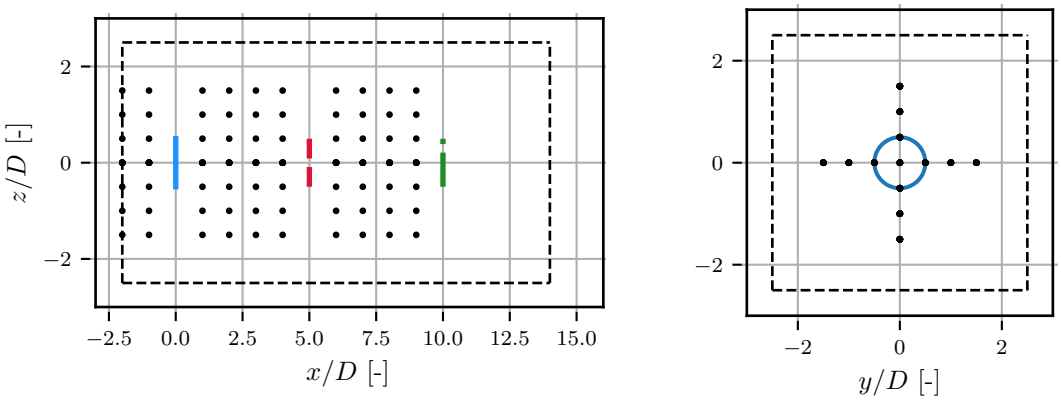

Figure 3.2.: Layout of the computational domain in the LBM-ALM environment. • Velocity probes — Turbine 0 --- Turbine 1 -·-· Turbine 2 ---- Inner domain —— Outer domain.

3.2.3. Optimizing Specific Behaviour

In the LBM-ALM Environment with the domain as specified in subsection 3.2.2, an agent controlling the amplitude of a sinusoidal variation of the pitch angles according to the helix strategy will be trained. Therefore the equation described in (3.2) will be modified like this:

$$\varphi(t) = \varphi_{base} + \sum_{i=0}^{N_a} A_{\varphi,i} \cos(2\pi f_{\phi,i} t + \Psi + \Psi_0) \tag{3.4}$$

Ψ_0 is the initial offset of the blade. The frequency $f_{\Phi,0}$ is chosen according to Frederik et al. so that the Strouhal Number $\mathrm{St} = f_{\Phi,0} D / V_0 = 0.25$. In difference to the setup by Frederik et al., the angular velocity of the rotor will not be constant, but is determined by (2.2). The generator torque is controlled by a greedy controller. Two different configurations of the agent will be compared. The first agent has ten nodes per layer and the state of the environment includes angular velocity and azimuth, while the second agent has 400 nodes per layer and the state additionally includes

the streamwise velocity of the velocity probes. The other parameters of both of the agents are the same as in Table 3.1.

3.2.4. Learning New Behaviour

To assess whether the RL agent is able to discover a new control strategy for wind farm control, an agent controlling the generator torque is tested, again in the environment specified by subsection 3.2.2. Three different agents will be compared. The first agent has a discount rate of $\gamma = 0.95$ and a number of actions per episode $N_{A,e} = 500$, which corresponds to setup used in the preliminary studies. The second agent has the same $N_{A,e}$ but a discount rate of $\gamma = 0.99$. The third agent also has a discount rate of $\gamma = 0.99$ but the number of actions per episode is $N_{A,e} = 1500$. The parameters differing from Table 3.1 are gathered in Table 3.3.

Table 3.3.: Parameters of agent for learning new behaviour

Parameter	Case 1	Case 2	Case 3
γ	0.95	0.99	0.99
$N_{A,e}$	500	500	1500
Width of actor network	400	400	400

This choice of parameters is motivated by a closer look at the meaning of reward and return in the context of this setup. The number of actions in the time it takes for information to travel from the first turbine to the second turbine can be estimated to be $N_{t,T} = d_T/(v_0 \cdot \Delta t_A) = 60$. This is a lower bound on the estimate, since information might travel slower due to a reduced velocity in the wake. The reward is calculated as follows:

$$\mathsf{R}(t) = \frac{P_0(t) + P_1(t) + P_2(t)}{\mathrm{MW}}. \tag{3.5}$$

Combining this with (2.22), the return of an action at the first turbine should be calculated like this:

$$\mathsf{G}_{t,0} = \sum_{t'=t}^{\mathsf{T}} \sum_{i=0}^{N_T-1} P_i(t' + iN_{t,T})\gamma^{t'+iN_{t,T}-t}. \tag{3.6}$$

Considering the values of this setup, a discount rate of 0.95 leads to a static discount of the seconds turbine power of around 95 percent, whereas a discount rate of 0.99 discounts the power of the second turbine only by 45 percent. When trying to achieve a combined control a high discount rate therefore seems to be beneficial. However, this also leads to little discounting of the power after information of the first turbine has reached the second turbine and therefore makes the return more noisy, as information that is probably less influenced by the action of the first turbine is added

to the return. Therefore a balance between these two influences has to be found.

4. Results

4.1. Validation

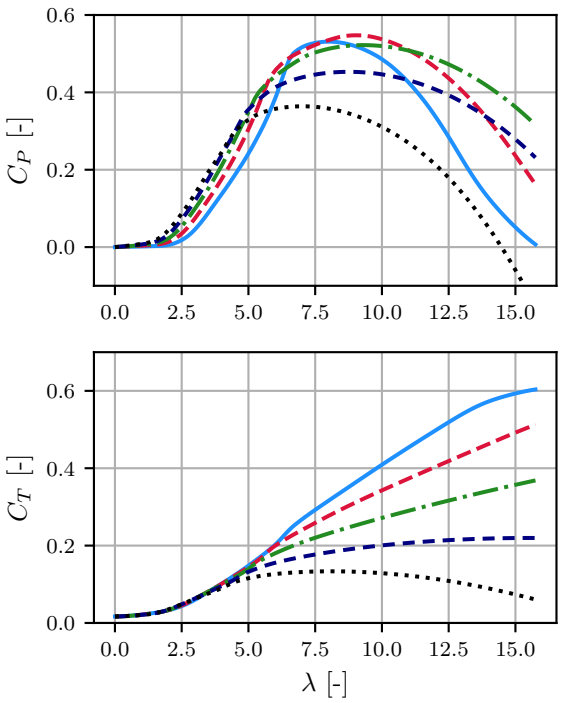

Figure 4.1.: $\Phi =$ —— -5 - - - -2.5 -·- 0.0 - - - 2.5 ····· 5. C_P and C_T as functions of tip-speed ratio and pitch angle for the NREL 5MW turbine.

As a first step, the implementation of the BEM environment is validated. Figure 4.1 shows C_P and C_T computed as functions of the tip-speed ratio λ and the collective pitch angle Φ. The results appear reasonable compared to results given in [22], where a smaller turbine is used. For the NREL 5MW data is not available. Furthermore, values obtained through interpolation of the precomputed table were compared to calculated values for 100 tip-speed ratios. The difference was found to be of $\mathcal{O}\left(10^{-5}\right)$ or smaller, while the interpolation is 170 times faster. Figure 4.1 shows that the power coefficient has a maximum between $\lambda = 6$ and $\lambda = 10.0$, depending on the pitch-angle. For $\Phi = 0$, the maximum C_P of 0.522 occurs at $\lambda = 9.23$. However, the thrust coefficient grows with tip-speed ratio for most pitch angles. The thrust influences the wake deficit and lifetime of wind turbines, therefore a reduction in λ below the value maximizing C_P can be advisable, as is commonly done in commercial turbines.

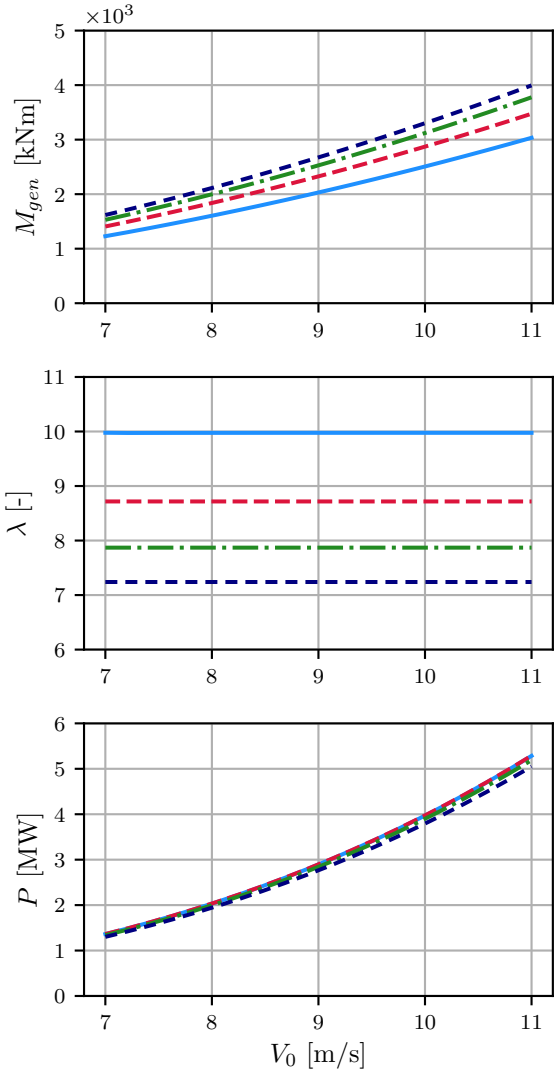

Figure 4.2.: $\kappa =$ 1.0 1.5 2.0 2.5 $[1 \times 10^6\,\mathrm{kgm}^2]$. Controller curve of greedy controller for different values of the proportionality constant κ

To validate the greedy controller, the controller curve is computed in the BEM environment. It is shown in Figure 4.2, for a range of values of κ. The choice of κ has little influence on the power in comparison to the wind velocity V_0. However, the tip-speed ratio is greatly influenced by the choice of κ. As discussed above, tip-speed ratio influences C_T and C_P. Thus a higher κ can reduce the thrust exerted on the turbine, with only a small reduction in power. Comparing the controller curve to the curve given in [29], a good agreement in P and M_{gen} is found for $\kappa = 2.5 \times 10^6\,\mathrm{kgm}^2$. However, since the lifetime of the turbine is not of concern, the κ maximizing C_P, $\kappa = 1.268 \times 10^6\,\mathrm{kgm}^2$, is chosen.

The LBM-ALM with a constant angular velocity of the rotor has been validated by Asmuth et al. in [4]. Therefore only the combination of the greedy controller with the LBM-ALM will be validated here.

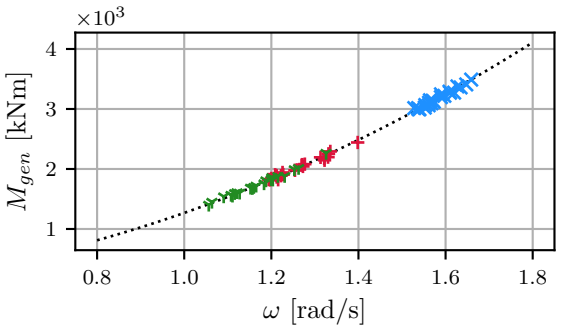

Figure 4.3.: ····· $\kappa\omega^2$ × Turbine 0 + Turbine 1 Y Turbine 2. Generator torque over angular velocity of three turbines controlled by greedy controller. Sampled every 1000th time step.

Figure 4.3 shows M_{gen} over ω for three turbines in a setup as described in subsection 3.2.2. The data gathered from the turbines agrees well with the expected value given by $\kappa\omega^2$. Deviations from the line are due to the turbulent inflow and the inertia of the turbine. The figure shows, that the turbine can follow the changing conditions closely. Furthermore, the influence of the wake loss can be seen. While the first turbine operates near the values expected at rated wind speed, the second turbine has lower angular velocity and torque and thus generates less power. The third turbine has even lower values, although the difference between the second and third turbine is not as big as between the first and the second.
Thus the environments and the greedy controller exhibit the expected behaviour and the coupling of them is working as well.

4.2. Parameter Study

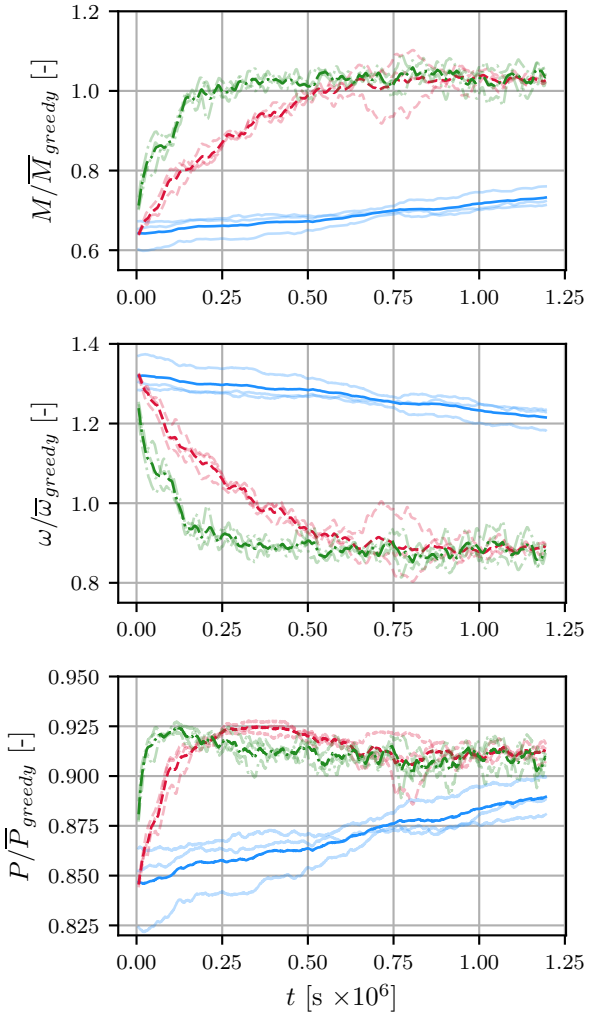

Figure 4.4.: $\sigma_A =$ — 0.01 - - - 0.05 -·-· 0.1. Sensitivity of training to σ_A. Thin lines are individual trainings, thick lines are the average of all trainings.

To gain insight into the behaviour of the RL-agent and the influence of some of the parameters, a parameter study is conducted using the BEM environment. To reduce the effect of the random initialization of the networks, for each tested value of a parameter, three independent agents are trained. The evolution of generator torque, angular velocity and power throughout training for a fixed number of time steps is compared, since these values are used as action, state and reward, respectively.

First the influence of the variance of the action, σ_A is studied. The results of the training for agents with variance of 0.01, 0.05 and 0.1 are shown in Figure 4.4. The comparison of the three values shows a clear trend. A higher variance leads to a faster increase in power. This can be expected, since a

larger variance in the action leads to trying a wider range of actions. Therefore finding a coarse estimate of a beneficial behaviour is more likely. Furthermore, it is visible, that the two agents with a higher variance reach similar behaviour. This increases the confidence, that this behaviour represents a local optimum. However, they both reach a maximum of generated power in the first half of the training, but are not able to sustain that value throughout the rest of the training. Comparing the different agents with the same set of parameters show that stochastic variation is present in training of the agents. After around two thirds of training time, one of agents with medium variance shows a significant drop in power, while another performs significantly better than average. However, towards the end of the training, the agents with medium variance perform more similar, more stable and slightly better than the agent with a higher variance. It can therefore be assumed that the choice of variance has to be a balance between a fast increase in the beginning and unstable behaviour in the end. It can be expected that the training of the agent controlling three turbines in the LBM-ALM environment will take significantly more time steps and the computational cost of a single time step is also much higher. Therefore, a faster increase is favoured and the variance is chosen to be $\sigma_A = 0.1$.

Next, the influence of an l_2-regularization of the policy network is examined. The results of the trainings without regularization and with regularization coefficients of 0.1 and 0.2 are shown in Figure 4.5. The sensitivity is significantly smaller to this parameter than to an increase in σ_A. While almost no difference is visible in the beginning for trainings with regularization, no regularization allows for a faster increase in power. However, on average, these agents drop the most afterwards. The agents also differ the most in power towards the end of the training. Remarkably, this is not the case for the generator torque, which differs most for $\varepsilon = 0.1$. A look back at Figure 4.1 shows, that near the optimum, C_P is not very sensitive to changes in tip-speed ratio, which explains this discrepancy in action and reward. The optimal tip-speed ratio corresponds to an angular velocity of $\omega = 1.58\,\mathrm{m/s}$. Figure 4.5 shows that most of the agents operate at angular velocities below the optimum. The aforementioned agent sets a lower generator torque, therefore the angular velocity increases and is actually closer to the optimum. The difference in the case without regularization is caused by an increased torque, which has a bigger effect on C_P. In the last third of the training, the agents with $\varepsilon_{p,l_2} = 0.1$ consistently perform best. Therefore this value is chosen.

Finally, the regularization of the value network is tested as well. The same coefficients as for the regularization of the policy network are tested. The sensitivity to a change in this parameter is similar to that of the policy regularization, as shown in Figure 4.5. A noticeable difference between the trainings is a delayed drop for trainings without regularization. However, this is only a delay. Later on the reductions in power are of similar magnitude as for the other cases, while the differences in generator torque are even larger than for the other agents. The second half of the training shows no clear trend for the best performance. The differences are small and the none of the averaged values is consistently better than the other. Since none of the tested values offers a clear advantage,

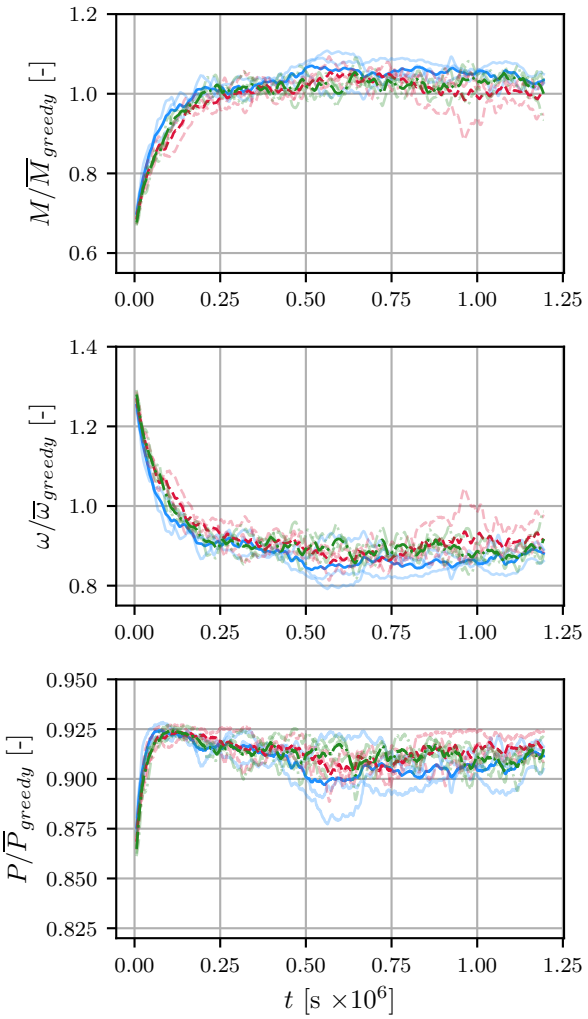

Figure 4.5.: $\varepsilon_{p,l_2} =$ —— 0.0 - - - 0.1 -·-· 0.2. Sensitivity of training to ε_{p,l_2}. Thin lines are individual trainings, thick lines are the average of all trainings.

the same value as for the regularization of the policy network is chosen.
The parameter study revealed a high sensitivity to a change in variance of the action and showed that an l_2-regularization benefits stability of the trained network while too much influence of the regularization might inhibit the optimization. In general, all of the studied trainings showed that the initial value of the generator torque is significantly lower than the optimal value. This is done by design, since a initial value that is too high leads to slowing down the turbine and a breakdown of M_{aero}, which ultimately makes a restart of the turbine necessary. Furthermore, almost all cases showed that P reached a maximum value and then decreased. This was caused by a generator torque that is consistently set higher than what would be expected from a comparison with the controller curve of the greedy controller in Figure 4.2. It can also already be noted that a lot of simulated time is necessary for the training of the agent. It can be expected that the necessary time

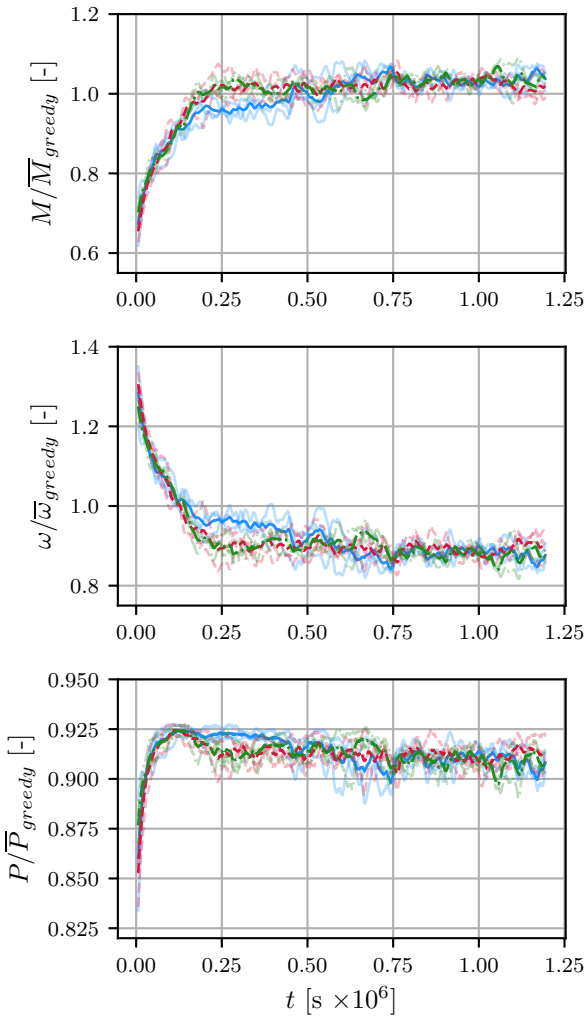

Figure 4.6.: $\varepsilon_{v,l_2} =$ — 0.0 - - - 0.1 -·-· 0.2. Sensitivity of training to ε_{v,l_2}. Thin lines are individual trainings, thick lines are the average of all trainings.

will be at least similar, but probably more for the training of an agent controlling three turbines.

4.3. Optimizing Parameters of a Dynamic Behaviour

4.3.1. A big network

Training

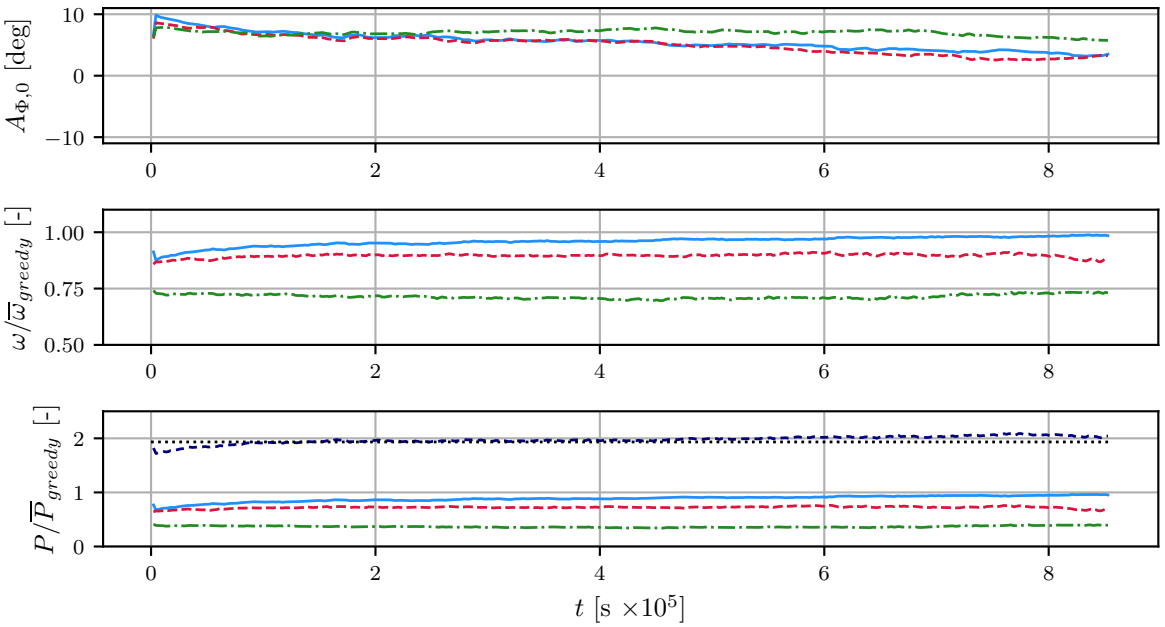

Figure 4.7.: Turbine 0 Turbine 1 Turbine 3 Total Greedy. Training of an agent controlling the amplitude of a sinusoidal variation of the pitch according to the helix approach. Only one environment shown. All values are a rolling average with a window of width $4359\,\mathrm{s}$. $\overline{P}_{greedy}$ and $\overline{\omega}_{greedy}$ are mean power and angular velocity of a single turbine controlled by a greedy controller.

To test the ability of the RL-agent to optimize the parameters of a given control strategy, two agents as described in subsection 3.2.3 were trained in the LBM-ALM environment with a small wind farm of three turbines. The time series of the training of the agent with the large network is shown in Figure 4.7. It shows the amplitude of the sinusoidal signal used to control the pitch angle, the angular velocity and the power of the three turbines as well as the combined power of the park. As a benchmark the mean power generated by a park controlled by the greedy controller is also given. In the beginning the agent performs considerably worse than the benchmark. It is also visible that the amplitude changes quickly at that point, while the changes afterwards is more gradual. The figure shows a strong correlation between increase in angular velocity and generated power for each turbine, which is to be expected since the generator torque is controlled by a greedy controller. The graph shows a steady increase in total power until about $7.5 \times 10^5\,\mathrm{s}$. At this point the power begins to decline. Again, the agent cannot remain at a local optimum. This motivates a closer look at the results and also to apply domain knowledge to judge the results and not simply accept them

as optimal.

A more detailed look into the evolution of the control strategy is offered by Figure 4.8. It shows

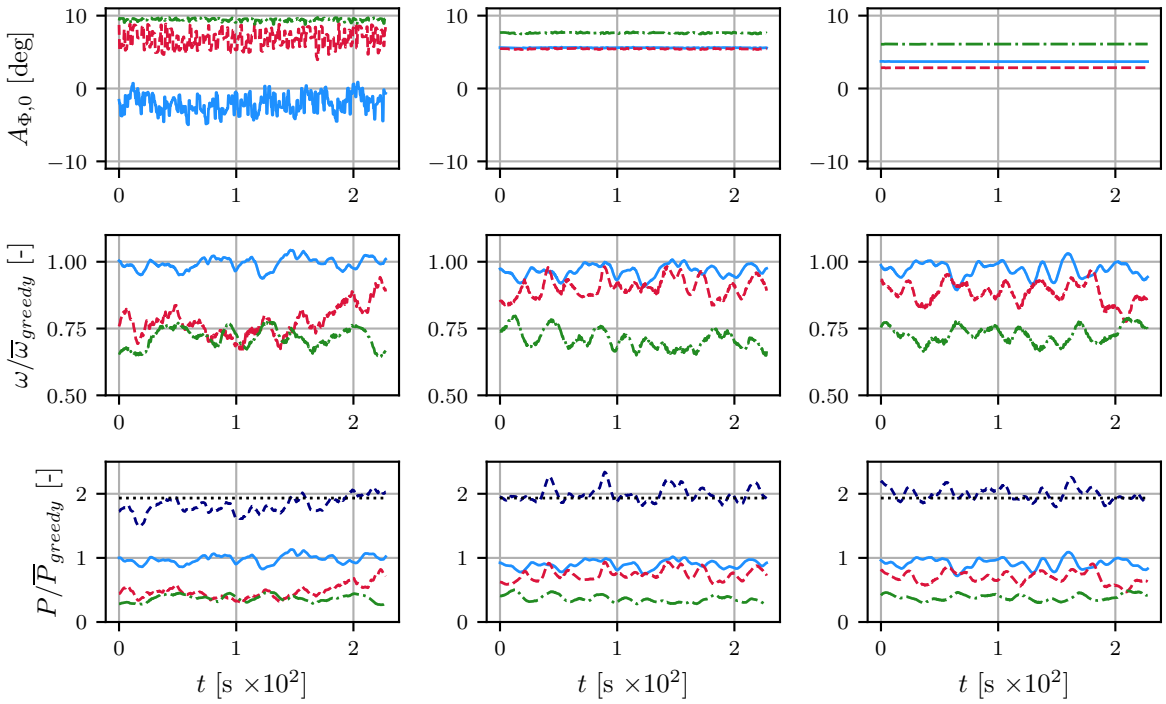

Figure 4.8.: Turbine 0 Turbine 1 Turbine 3 Total Greedy. Control strategy at three points during training. Left at the beginning of the training, center after half of training time, right after the training is finalized.

the same quantities as the figure before, for one flow-through-time. The results in the left column are obtained with the strategy after the first update, the results in the central column with the strategy after half of the training time and the right column displays the final strategy. It shows that the initial agent reacted to the fluctuating inputs of the network, that is the sampled velocities and angular velocities of the turbines, whereas the agent in the final state sets constant amplitudes. The agent after half the training time still shows some reaction, most visible in the amplitude of the last turbines pitch. That this quantity evolves the slowest can be explained by the fact, that power production is the lowest at the last turbine. Therefore the same change in relative power or C_P has a lower impact on the reward. Thus the control of the last turbine evolves the slowest. The figure also shows, that the power in the last turbine has changed little over the course of the training, which is also visible in Figure 4.7. Instead, the increase in power can be attributed to the second turbine, while the first turbine decreased in power. The strategy obtained for the last turbine is highly questionable, since curtailment of the last turbine presumably offers no benefits. Therefore a greedy strategy, that is an amplitude of zero degrees would be expected. It cannot be ruled out that this strategy would emerge with longer training. However, the computed time to reach this

strategy can roughly be estimated to be $5.75 \times 10^5\,\mathrm{s}$, when taking the mean change in amplitude over the last $1 \times 10^5\,\mathrm{s}$. Furthermore it shows that the amplitude of the pitch of the first and the second turbine are similar, about 4° and 3°, respectively. This suggests that this a good balance between optimal C_P and the effects allowing for a higher power production by the following turbine.

Analysis of the control strategy

Table 4.1.: Mean, relative difference in mean and relative difference in standard deviation of power and aerodynamic moment of the helix approach after optimization in comparison to the greedy-control case.

	P			M_{aero}		
	mean	rel. mean	rel. std	mean	rel. mean	rel. std
Total	$10.4\,\mathrm{MW}$	$+6.75\,\%$	$+4.78\,\%$	-	-	-
Turbine 0	$4.81\,\mathrm{MW}$	$-5.04\,\%$	$-2.78\,\%$	$3.08 \times 10^3\,\mathrm{kNm}$	$-3.39\,\%$	$-1.14\,\%$
Turbine 1	$3.67\,\mathrm{MW}$	$+43.4\,\%$	$+17.6\,\%$	$2.57 \times 10^3\,\mathrm{kNm}$	$+27.3\,\%$	$-0.331\,\%$
Turbine 2	$1.97\,\mathrm{MW}$	$-8.95\,\%$	$-12.2\,\%$	$1.7 \times 10^3\,\mathrm{kNm}$	$-6.08\,\%$	$-1.53\,\%$

To also provide quantitative results, Table 4.1 gathers mean power and aerodynamic moment as well as relative changes in mean and standard deviation of power in aerodynamic moment in comparison to a park that is controlled by a greedy controller. It shows that the overall gain in power is almost seven percent. It also confirms the previous observation, that this gain in power is due to an increase in power production by the second turbine, which increased by more then 40 percent in comparison to a greedy-controlled park. The third turbine decreased in power by almost 9 percent. Since the strategy of both previous turbines is very similar, one could expect a similar increase in power as for the second turbine. Comparing the results for the two first turbines to those obtained by Frederik et al., it can be seen that the power at the first turbine is reduced more, but that the increase in generated power at the second turbine is greater and that the overall increase in power also is considerably higher, namely 11 percent compared to 7.5 percent found by Frederik et al. [10]. Looking beyond a simple increase in power production, quality of generated power and turbine loads are of very high importance. One measure of quality of power is low fluctuations, which is why the standard deviation of power is also given in Table 4.1. It shows, that the fluctuations of the total power increased, due to a big increase in the fluctuations at the second turbine. The second aspect is the loads of the turbine, which is also a very active area of research. To that end, it will be pointed out, that aerodynamic torque has decreased at the first turbine and that the fluctuations have decreased at all the turbines. Therefore, like power production, the aerodynamic torque is more evenly distributed amongst the turbines, which likely has a beneficial influence on turbine lifetime. The influence on the thrust force were already assessed in [10], where only small

differences compared to a greedy control were found.

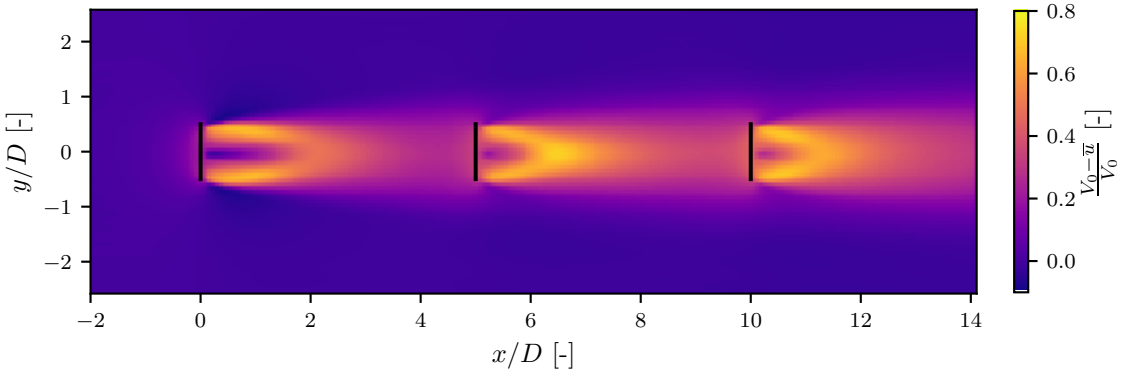

Figure 4.9.: —— Turbine. Mean velocity deficit compared to mean wind speed with optimized helix control.

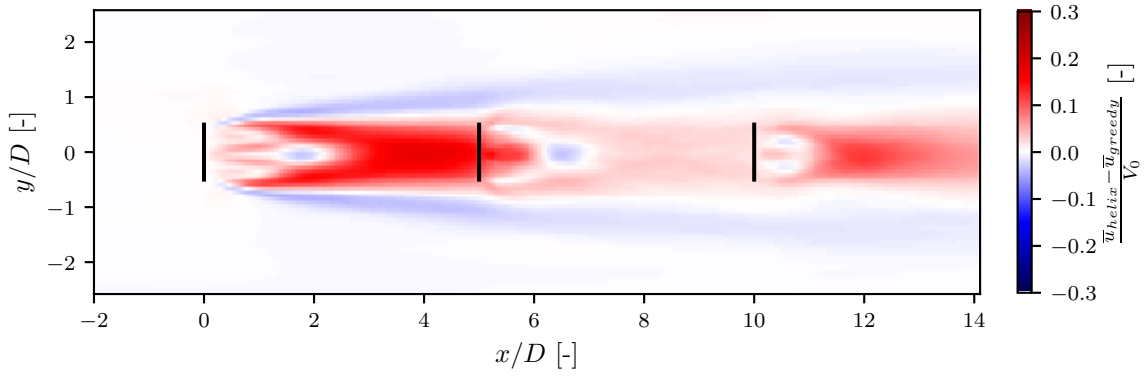

Figure 4.10.: —— Turbine. Difference in mean velocity between optimized helix control and greedy control.

To assess the causes for the increase in power, Figure 4.9 and Figure 4.10 show the velocity deficit in comparison to the mean wind speed and the difference in velocity in comparison to the greedy-control case. As expected, a wake deficit of up to 80 percent of inflow velocity is visible after each turbine. Differences are visible in the shape of the wakes behind the three turbines. The deficit is strongest at the tips of the blades in all cases, but at the second and the third turbine these minima meet in the center of the wake shortly behind the turbine, while they stay separated behind the first turbine. Comparing the fluid field in front of the turbine shows that the deficit in front of the third turbine is larger than in front of the second turbine. While some of the lower power production probably has to be attributed to the suboptimal control strategy, power production of the third turbine can probably not reach the amount of the second turbine since the inflow velocity is smaller. In the comparison to the greedy-control case, the increased speed in the wake is brightly visible, with difference of $1.5\,\mathrm{m/s}$ at the core of the wake. One can also see an area of lower mean speed on the edge of the deficit. This means that the wake in the helix approach is more spread

out, which is the motivation behind this approach. Behind the second turbine wind speeds are also higher, but the difference is not as big as in the first turbines wake. This is an important observation in the general behaviour of the helix approach, since Frederik et al. only considered two turbines in their study [10]. If the increase in velocity can not be found after the second turbine, application to a wind farm is not as beneficial as suggested by the initial study. The wake after the third turbine has again a higher velocity than in the greedy case, which again points to the fact, that more power could be generated by the third turbine.

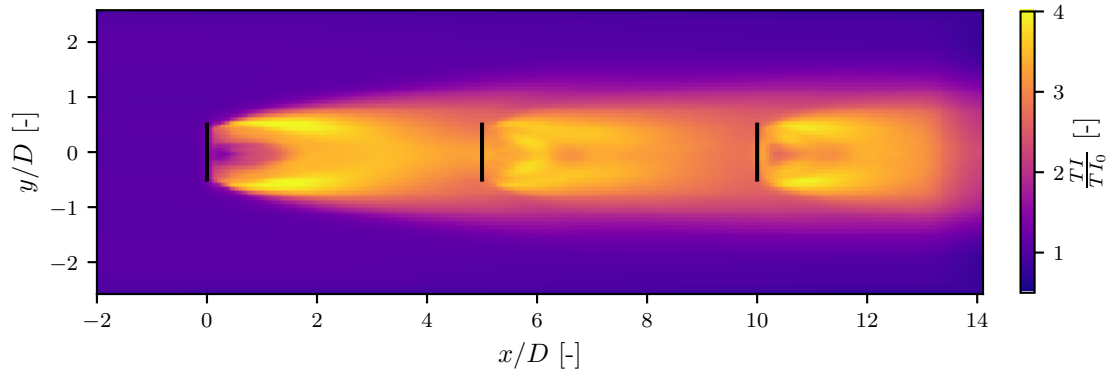

Figure 4.11.: —— Turbine. Turbulence intensity of optimized helix control.

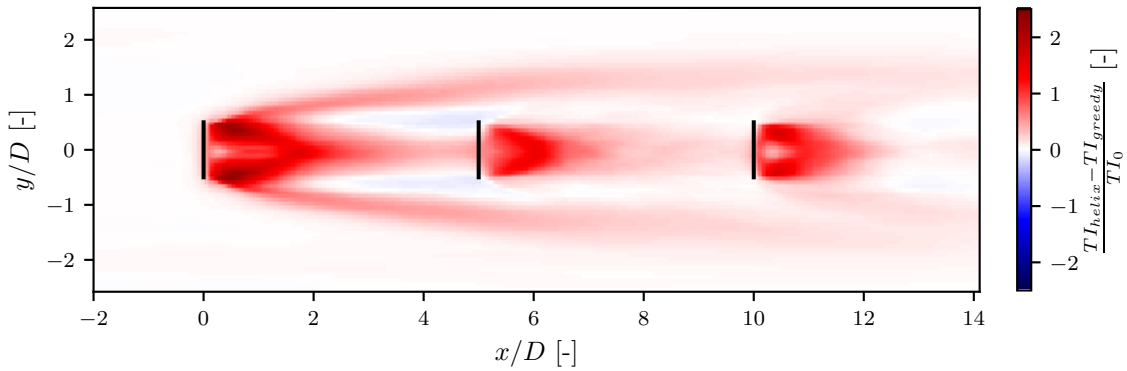

Figure 4.12.: —— Turbine. Difference in turbulence intensity between optimized helix control and greedy control.

The previous figures showed, that the second turbine caused a larger wake deficit compared to the inflow velocity than if it was greedy-controlled. This is not the case after the first turbine, where the deficit is significantly lower. A possible explanation for this can be found in the secondary statistics of the fluid, that is its fluctuations. Therefore Figure 4.11 shows the turbulence intensity of the helix control case and Figure 4.12 the difference of the helix control case to the greedy control case. The first figure shows that the turbulence intensity is up to four times higher in the wake. It also shows,

that areas, that have a high velocity deficit, are surrounded by an area of high turbulence intensity. This is expected, since these are areas of high velocity gradients. Another observation that can be made is that the area with increased turbulence intensity grows in diameter up to about half a turbine distance after the second turbine and stays constant after that. A high turbulence intensity at the edges of the wake is desirable since this increases entrainment of the surrounding fluid with high momentum and thus wake recovery. The comparison to the greedy controller shows, that this is achieved by the helix approach, most prominently after the first turbine.

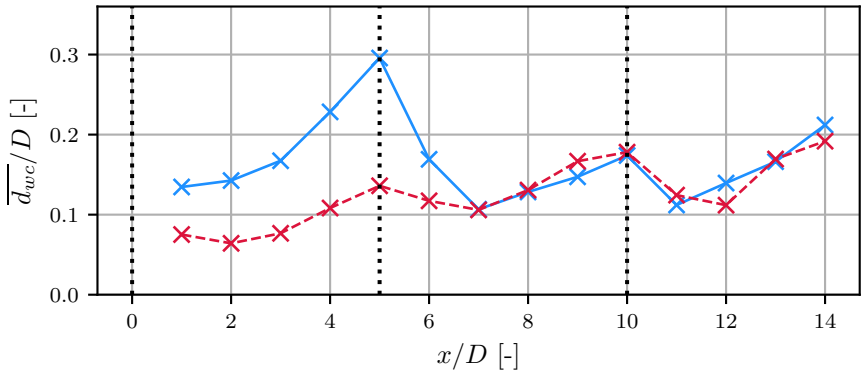

Figure 4.13.: Helix Greedy Turbines. Mean distance of wake center to plane center. Crosses mark the location of the planes in which the wake center is calculated.

In order to analyze the influence of the helix approach on the dynamics of the wake, the wake centers were calculated using the samwich toolbox [1] provided by NREL. It contains different methodologies to determine the wake center. In this work the wake center is defined as the center of a least-square fit of a Gaussian bell curve to the velocity deficit. Figure 4.13 shows the distance of the wake center d_{wc} to the center of the cross stream plane averaged over 100 snapshots of the fluid field at 14 planes downstream of the first turbine with a distance of a turbine diameter. It shows, that the wake is steered further away from the center after the first turbine, while this is not the case after the second turbine, where greedy and helix control lay on top of each other. After the third turbine the helix control increases the mean wake distance again. Areas, where $\overline{d_{wc}}$ is larger correspond to areas with lower mean wake deficit, which shows, that the wake is not expanded in comparison to the greedy-control case, but skewed away from the center. This also explains, why a band of high turbulence intensity was visible in Figure 4.12. This is the area only reached by the wake of the helix control case. Remarkably, the helix approach is not able to the skew the wake after the second turbine. This is most likely why the third turbine cannot increase its power production. Several aspects can contribute to this. First, the turbulence intensity of the inflow of the second turbine increased in comparison to the inflow of the first turbine. This likely reduces the moment exerted by the turbine on the wake. Furthermore, it is questionable, whether the frequency

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

at which the blade pitch is altered is appropriate at the second turbine. Frederik et al. showed that the wake deficit is sensitive to the Strouhal number in [10]. Assuming a velocity deficit of 30 percent of the inflow velocity, the Strouhal number would increase from 0.25 to 0.35, which, according to Frederik et al., lead to an increase in the wake deficit of about 10 percent. Thus this is a possible point for future optimization, but does not account for the entire effect. However, the decreased velocity also has an effect on tangential forces exerted on the wake, since the relative velocity at the blade and the angle of attack change. Both these changes decrease the tangential force and thus the moment exerted on the wake.
The training data showed, that the training requires a lot of computational time and that optimization is slow. However, the results did continually improve for most of the training, but, as was seen in the parameter studies as well, optimal behaviour is lost again after more training. Therefore, results of training can not be guaranteed to be optimal. The analysis of the optimized helix control showed, that due to the wake steering an increase in power production by the second turbine is possible, with little decrease in production of the first turbine. However, it also showed, that steering of the second wake was not achieved and therefore benefits for the third turbine are small, in this case even resulted in a decrease of power. Further testing of this strategy is required, for example in sheared inflow and in a park with more turbines.

4.3.2. A small network

Training

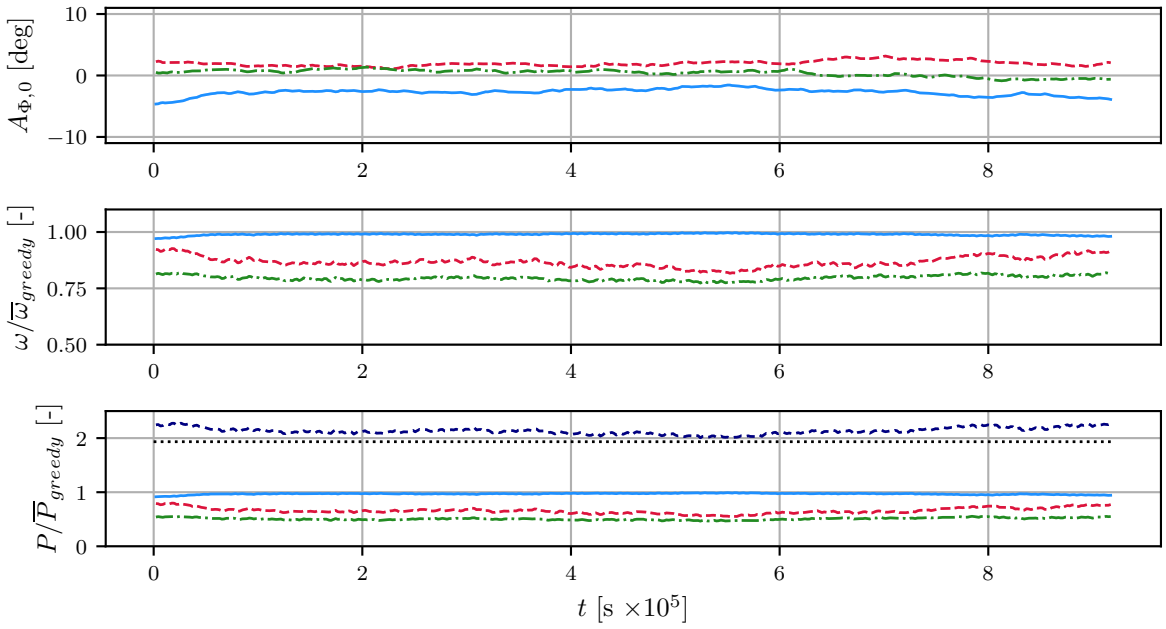

Figure 4.14.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Training of an agent with 10 nodes per layer, controlling the amplitude of a sinusoidal variation of the pitch according to the helix approach. Only one environment shown. All values are a rolling average with a window of width $4359\,\mathrm{s}$.

Since the training of the agent with 400 nodes per layer proved to be computationally very expensive, a second agent with a smaller network was trained as well, since a smaller network has fewer parameters to optimize. The time series of the amplitude of the pitch angles, the angular velocities and generated power is shown in Figure 4.14. It shows that the training does not reach a stable reward or action. However, the same reward and control strategy is reached twice, therefore it is assumed to be the local optimum and training is not continued further. In comparison to the agent with the larger network, training was not significantly shorter. The plots also show, that the control strategy did not change as much as during the training of the other agent and that this agent performed continuously better than the greedy-control reference.
For a more detailed look at the evolution of the control strategy, Figure 4.15 shows the results of simulations with the agent after twelve updates, after half of the updates and in the final state. As in the previous case, in the beginning the controlled variable $A_{\Phi,0}$ varies due to a fluctuating state of the environment. After half the training, no fluctuations can be seen any more and agent prescribes a steady value for the amplitudes. The fully trained agent also sets constant values, which also only changed by a small amount in comparison to the agent after half the training. In comparison to the

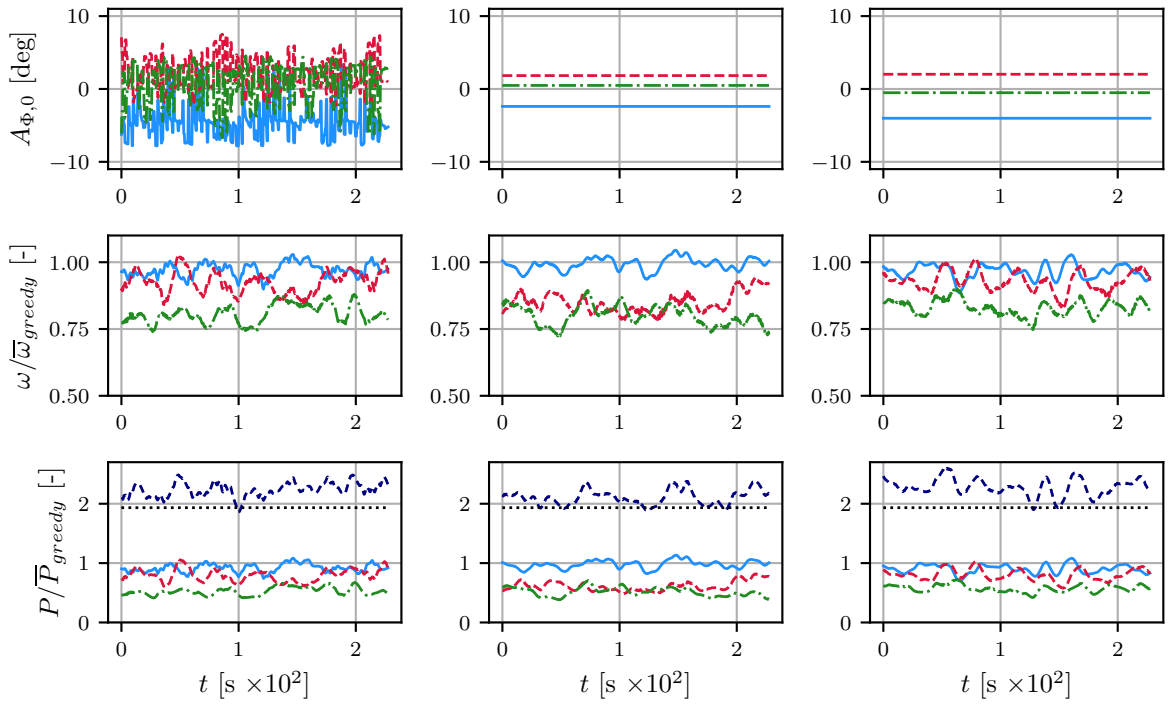

Figure 4.15.: Turbine 0 Turbine 1 Turbine 3 Total Greedy. Control strategy at three points during training of an agent with 10 nodes per layer, controlling the parameters of a helix strategy. Left column at the beginning of the training, center column after half of training time, right column after the training is finalized.

mean power by a greedy controlled park, the total power is always greater or equal. The strategy developed by this agent differs in two key points from that of the agent with a wider network. First, the last turbine's amplitude is close to zero, which agrees with the expected value. Second, the amplitude of the first turbine has a negative sign, which results in a phase shift of $180°$. However, the absolute values of the amplitudes of the first and second turbine are similar between the two agents, the first agent set the amplitude of the first turbine to $3.75°$, whereas this agent sets it to $4.0°$. The amplitude of the second turbine differs a little more, the agent with the wide network set it to $2.85°$ and the other to $2°$.

Analysis of the control strategy

The results on mean power and mean aerodynamic torque are shown in Table 4.2. The overall power generated increased by $18.2\,\%$. This increase in total power is due to an increase of power generated by the second and the third turbine by $56.8\,\%$ and $29\,\%$, respectively. This is a bigger increase of power production by the second turbine than was reached by the other agent. The increase in power produced by the first two turbines is also twice as high as found by Frederik et

Table 4.2.: Mean, relative difference in mean and relative difference in standard deviation of power and aerodynamic moment of the helix approach after optimization in comparison to the greedy-control case.

	P			M_{aero}		
	mean	rel. mean	rel. std	mean	rel. mean	rel. std
Total	$11.6\,\mathrm{MW}$	$+18.2\,\%$	$+15.9\,\%$	-	-	-
Turbine 0	$4.76\,\mathrm{MW}$	$-5.9\,\%$	$-3.9\,\%$	$3.06 \times 10^3\,\mathrm{kNm}$	$-3.97\,\%$	$-1.57\,\%$
Turbine 1	$4.01\,\mathrm{MW}$	$+56.8\,\%$	$+23.8\,\%$	$2.73 \times 10^3\,\mathrm{kNm}$	$+35.1\,\%$	$-2.49\,\%$
Turbine 2	$2.8\,\mathrm{MW}$	$+29.0\,\%$	$+13.2\,\%$	$2.14 \times 10^3\,\mathrm{kNm}$	$+18.7\,\%$	$+2.37\,\%$

al. [10]. Furthermore, in the previous case the power produced by the third turbine decreased, while this time an increase is measured. That power production by the third turbine is different is to be expected since the observation was made previously, that the last turbine did not perform optimally when controlled by the other agent. However, the further increase in power produced by the second turbine is not so easily explained. Possible reasons are the phase shift between first and second turbine or the increased amplitude of the pitch oscillations of the second turbine. Power production of the first turbine decreased by almost six percent. The aerodynamic torque of the first turbine decreased, while it increased at the other two. The dynamic load, as represented by standard deviation of M_{aero}, decreased for the first two turbines compared to the greedy-controlled park. It only increased at the last turbine. The decrease in dynamic loads is favourable, since dynamic loads have a negative effect on lifetime. The overall quality of power was decreased, as indicated by the standard deviation of total power. However, an increase in total power production in comparison to the decreased steadiness is probably still favourable.

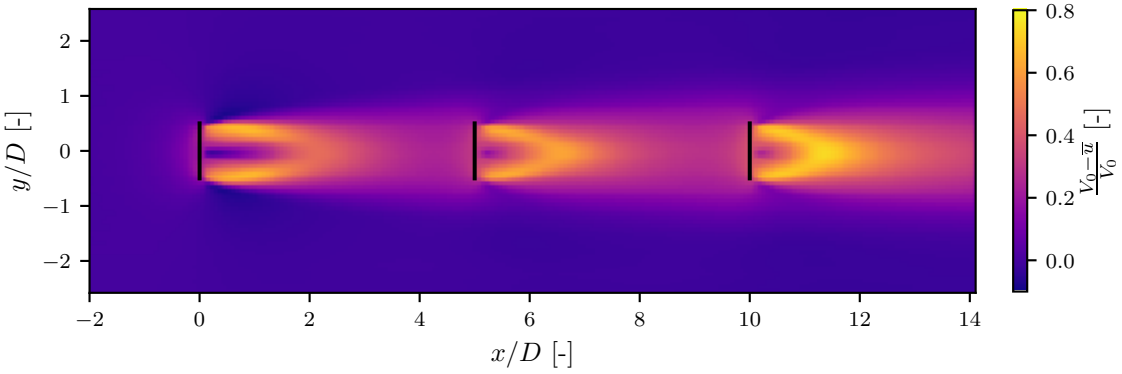

Figure 4.16.: —— Turbine. Mean velocity deficit compared to mean wind speed with optimized helix control.

To investigate the effects of this control strategy further, the mean wake deficit of a simulation with the agent with a small network is shown in Figure 4.16. It shows, that the maximum wake deficit

occurs after the third turbine as it is expected for optimal behaviour. Furthermore it shows that the wake deficit in front of the second and the third turbine is small, which is also expected for optimal behaviour. The comparison to the previous agent shows, that the wake deficit after the second turbine is significantly smaller, even though the power production is increased. This also causes the inflow velocity of the third turbine to be higher, which also allows for a higher power production there.

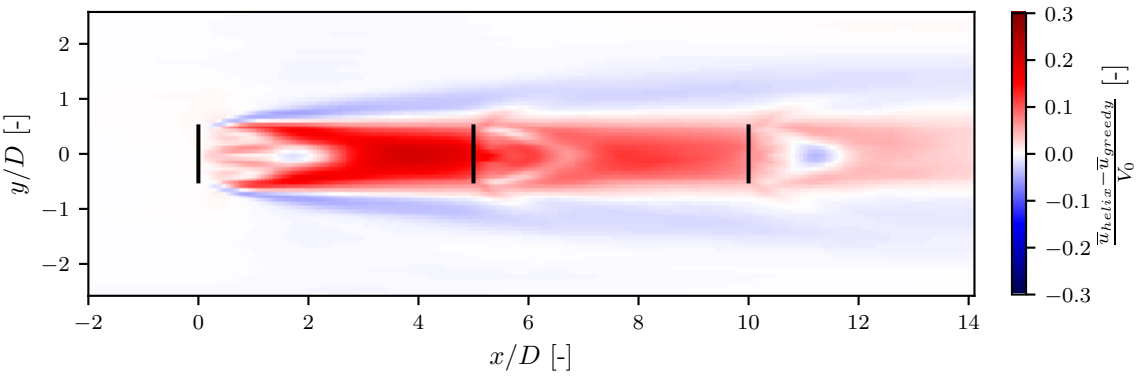

Figure 4.17.: —— Turbine. Mean velocity when turbines are controlled with optimized helix control compared to greedy controlled case.

The comparison to the greedy controller in Figure 4.17 shows a similar picture. In an area about the width of one diameter the velocity behind the first and second turbine is significantly higher, further away from the center a higher velocity deficit is visible. This was also the case when the turbines were controlled by the agent with a wider network. The increase in spread of the wake is similar to the other agent. However, the wake deficits near the core of the wake differ. Behind the second turbine only a small conic shell has a higher deficit in comparison to the greedy controller, while the rest of the seconds turbine's wake has a significantly lower deficit. Behind the third turbine the velocity deficit of the turbine controlled by this agent is higher than when the park is controlled by the other agent, which is probably due to a higher oscillation amplitude.
To show the second order statics of the flow, Figure 4.18 shows the turbulence intensity of a park controlled by this agent. It shows that the turbulence intensity is increased in the wake, but also that an area of low increase exists behind the second and third turbine. It shows also that the highest increase in turbulence intensity is after the first turbine. The maximum of turbulence intensity in each wake decreases in stream direction. After the first, but especially after the second turbine the coss-sectional area of high turbulence intensity quickly decreases downstream. After the third turbine this is not the case, this can most likely be attributed to low pitch oscillation, but effects due to the end of domain can play a role as well. The distribution of turbulence intensity after the first turbine is very similar to that of the park controlled by the agent with a large network. This is not the case after the second and third turbine. After the second turbine the area of lower increase of

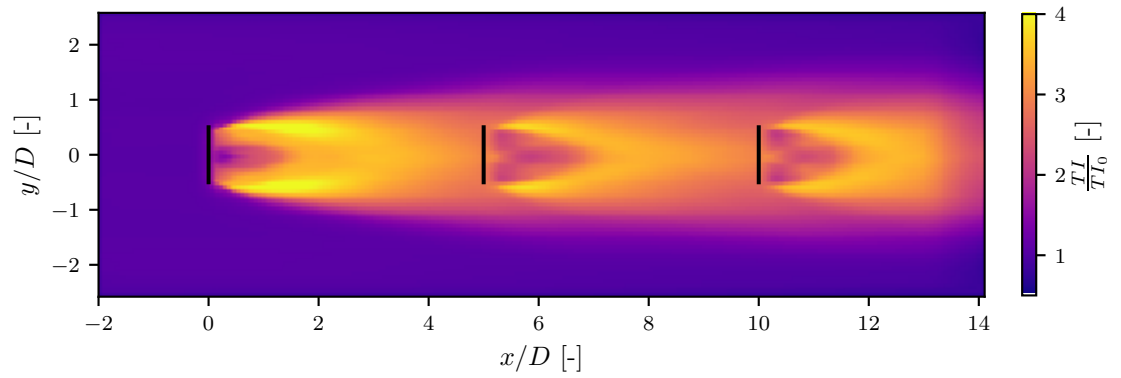

Figure 4.18.: —— Turbine. Turbulence intensity of park controlled with optimized helix control compared to turbulence intensity at inlet.

TI does not exist in the park controlled by the other agent. After the third turbine it is significantly smaller. Also the decreasing diameter of the area of high intensity is not visible. Therefore it must be caused by the phase shift or the increased oscillation amplitude.

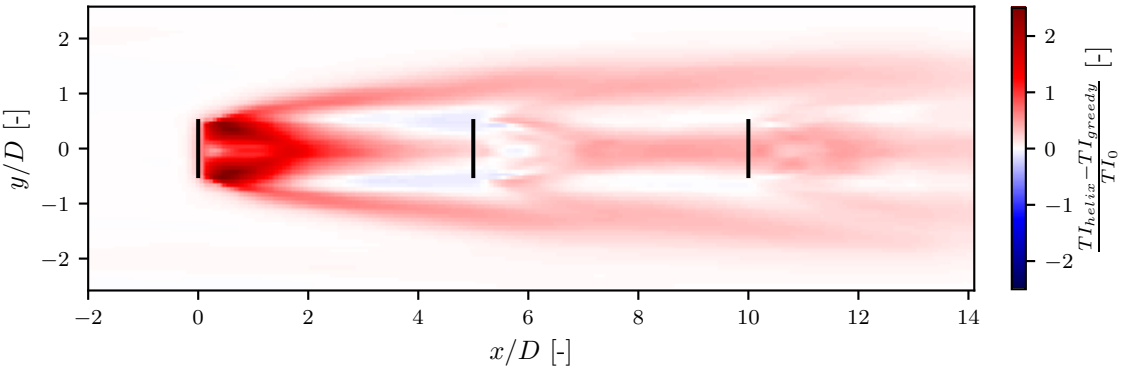

Figure 4.19.: —— Turbine. Turbulence intensity of park controlled by optimized helix control compared to greedy-controlled park.

In Figure 4.19 the difference in turbulence intensity in comparison to a greedy-controlled park is shown. It shows, that the turbulence intensity is increased throughout the wake with the exception of a narrow annular ring with the diameter of the turbine. This shell is visible in each wake and begins around $1.5D$ downstream of the turbines. The difference in TI is of the same magnitude in the wake of the second and third turbine. The figure also shows that an area of increased turbulence intensity exists connecting the second turbine to the outer area of turbulence intensity. Furthermore this outer area has a higher turbulence intensity compared to the park controlled by the agent with the wider network. This suggests that the wake might be steered outside the center like the wake of the first turbine. This was not the case when controlled by the other agent. This would offer an explanation of the increased velocity and consequently higher power production at

the third turbine.

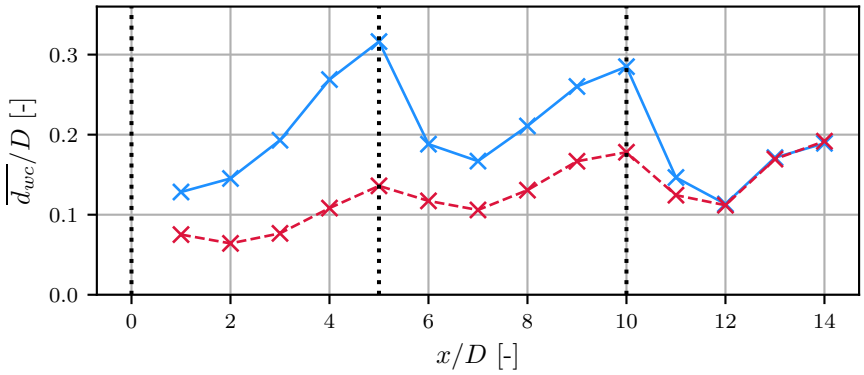

Figure 4.20.: Helix Greedy. Turbines. Mean distance of wake center to center of cross-plane at 14 cross-planes.

To verify this last hypothesis, the mean distance of the wake center to the center of the cross plane is shown in Figure 4.20. It shows again, that wake of the first turbine is steered further away from the center than if the turbine had no oscillation. The same is true for the second turbine. This confirms the previous hypothesis and explains the increased power production of the third turbine. This leads to the question why the oscillations cause the steering in this case but not in the park controlled by the other agent. Since the wake of the first turbine is the same in either case, two possible explanations remain. It is either caused by the decreased amplitude of oscillations by the second turbine or the phase shift of the oscillations at the first turbine. To test this, simulations with a controller with the same amplitudes were run, one with phase shift and one without. The mean wake distance to the cross-wise center are shown in Figure 4.21. It clearly shows that the increased in $\overline{d_{wc}}$ after the second turbine is caused by the phase shift.

Analysis of the helix control strategy for two turbines

To further investigate the influence of the phase shift, a look back at the governing physics has to be taken. Frederik et al. [10] showed, that the oscillation of the pitch angles with the helix approach results in moments in vertical and cross-stream direction, M_{tilt} and M_{yaw}, respectively. The resulting moment vector will be called $\mathbf{M}_{osc}$. They oscillate with the excitation frequency f_e and with an amplitude A_M :

$$\mathbf{M}_{osc} = \begin{bmatrix} M_{tilt}(t) \\ M_{yaw}(t) \end{bmatrix} = A_M \cdot \begin{bmatrix} sin(2\pi f_e t) \\ \cos(2\pi f_e t) \end{bmatrix} \tag{4.1}$$

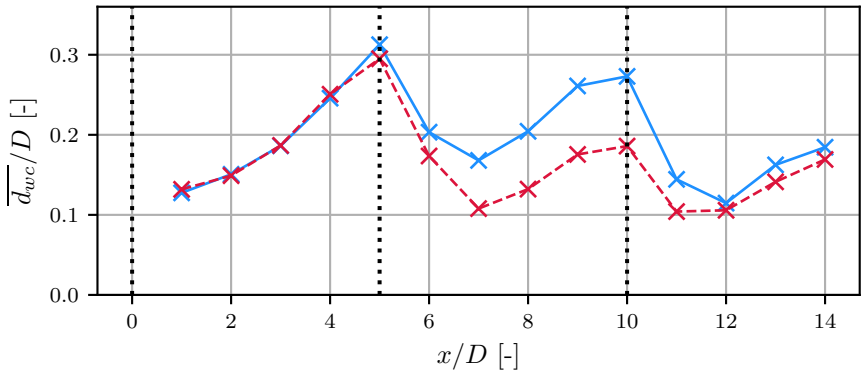

Figure 4.21.: — Phase shift --- No phase shift ····· Turbines. Two controllers with helix control and values optimized by second agent.

The moment resulting on the wake after the second turbine will be called $\mathbf{M}_{helix,1}$. It is the sum of $\mathbf{M}_{osc,1}$ due to oscillations by the second turbine and the moment transported by the wake from the first turbine $\mathbf{M}_{helix,0}$. It is equal in magnitude to $\mathbf{M}_{osc,0}$ but rotated by $\phi_{travel} = 2\pi f_e T_{travel}$, since it is the moment exerted by the turbine time T_{travel} ago. With $A_{M,0}$ and $A_{M,1}$ as the amplitudes of the moments exerted by the first and second turbine, respectively, and $\mathbf{I}$ as the identity matrix and $\mathbf{R}(\phi)$ as the rotation matrix about angle ϕ, the moment of the helix after the second turbine is:

$$\mathbf{M}_{helix,1} = \mathbf{M}_{osc,1} + \mathbf{M}_{helix,0} = (A_{M,1}\mathbf{I} + A_{M,0} \cdot \mathbf{R}(\phi_{travel})) \cdot \begin{bmatrix} \sin(2\pi f_e t) \\ \cos(2\pi f_e t) \end{bmatrix}. \tag{4.2}$$

The magnitude of the resulting moment $\mathbf{M}_{helix,1}$ is:

$$\|\mathbf{M}_{helix,1}\| = \sqrt{A_{M,1}^2 + A_{M,0}^2 + 2A_{M,0}A_{M,1}\cos(\phi_{travel})}. \tag{4.3}$$

It shows that the previous turbine influences it in two ways. The term $A_{M,0}^2$ can not be negative and must therefore increase the moment. Only the second term, which contains the amplitude and the cosine of ϕ_{travel} can result in a decrease in $\mathbf{M}_{helix}$. Assuming that $A_{M,0}$ is proportional to $A_{\Phi,0}$, which is the amplitude of the pitch oscillations, the negative amplitude results in an increase in $\mathbf{M}_{helix}$ if $\cos(\phi_{travel}) < 0$. With St and mean inflow velocity V_0, ϕ_{travel} is:

$$\phi_{travel} = 2\pi \mathrm{St} \frac{V_0}{V_{helix}} \frac{d_{Turbines}}{D}, \tag{4.4}$$

with V_{helix} as the transportation velocity of the helix. In the case studied, $\phi_{travel} = 5\pi/2 \cdot V_0/V_{helix}$. Since the wake cannot be faster than the inflow velocity and considering the results from Figure 4.16, $3 \geq V_0/V_{helix} \geq 1$ and therefore $\cos(\phi_{travel}) \leq 0$. Thus a negative amplitude and the resulting phase shift must be beneficial.

Comparing $\mathbf{M}_{helix,1}$ and $\mathbf{M}_{helix,0}$, it can be shown that the two moments have a phase shift of ϕ_0:

$$\tan(\phi_0) = \left(\frac{A_{M,0}\cos(\phi_{travel})}{A_{M,1} + A_{M,0}\cos(\phi_{travel})} - 1 \right) \tan(\phi_{travel}) \tag{4.5}$$

To eliminate this phase shift in the helix, a phase shift in the oscillations at the second turbine can be introduced. Adding a phase shift of ϕ_{travel} to $\mathbf{M}_{osc,1}$ results in:

$$\mathbf{M}_{helix,1} = (A_{M,1} + A_{M,0})\,\mathbf{R}(\phi_{travel}) \cdot \begin{bmatrix} \sin(2\pi f_e t) \\ \cos(2\pi f_e t) \end{bmatrix}. \tag{4.6}$$

The magnitude of this new moment is easily shown to be:

$$\|\mathbf{M}_{helix,1}\| = A_{M,1} + A_{M,0} \tag{4.7}$$

By design, the helix before and after the turbine do not have a phase shift.
To implement this phase shift, first ϕ_{travel} has to be found. From (4.4) all quantities except the transportation velocity V_{helix} are known. To measure this velocity, the phase angle of the wake center in each y-z-plane of the wake behind the first turbine is calculated. The mean of ϕ_{total} of 200 time steps is calculated. The velocity is calculated by $V_{helix} = 2\pi f_e x/\phi_{total}$. The velocity is computed from the angles between $2.5D$ and $3.8D$, to avoid disturbances due to the turbines. In the case studied $V_{helix}/V_0 = 0.818$. The mean total angle and the angle predicted by V_{helix} are shown in Figure 4.22. The figure shows good agreement between the two quantities in the wake after the first turbine, but significant deviation after the second turbine, which is to be expected due to the phase shift predicted by (4.5). The measured velocity predicts a rotation of $\phi_{total} = 3.06\pi$. This is very close to

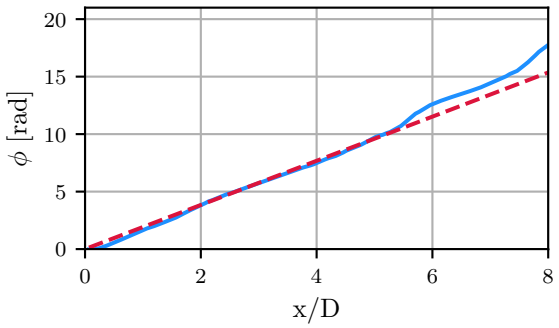

Figure 4.22.: $\overline{\phi_{total}}$ $2\pi f_e x/V_{helix}$

3π which would be the same as a phase shift of π as was performed by the agent. In Table 4.3 the mean , difference in mean relative to a greedy controller and difference standard deviation relative to a greedy controller are shown for power and M_{aero}. A comparison to Table 4.2 shows that the

two are almost equivalent, differences are in the range of five percent for most quantities.

Table 4.3.: Mean and relative difference in mean and standard deviation to a greedy-control case of power and aerodynamic moment of a park controlled with the phase shifted helix strategy.

	P			M_{aero}		
	mean	rel. mean	rel. std	mean	rel. mean	rel. std
Total	$11.5\,\mathrm{MW}$	$+17.5\,\%$	$+14.3\,\%$	-	-	-
Turbine 0	$4.76\,\mathrm{MW}$	$-5.88\,\%$	$-5.3\,\%$	$3.06 \times 10^3\,\mathrm{kNm}$	$-3.96\,\%$	$-2.11\,\%$
Turbine 1	$3.95\,\mathrm{MW}$	$+54.5\,\%$	$+28.6\,\%$	$2.7 \times 10^3\,\mathrm{kNm}$	$+33.8\,\%$	$-0.523\,\%$
Turbine 2	$2.79\,\mathrm{MW}$	$+28.6\,\%$	$+12.8\,\%$	$2.14 \times 10^3\,\mathrm{kNm}$	$+18.4\,\%$	$+1.71\,\%$

In conclusion, the phase shift in the helix arriving at the second turbine explains why a helix exists after the second turbine, when the park is controlled by the second agent, but not when it is controlled by the first agent. It also explains the increase in power produced by the second turbine when controlled by the second agent in comparison to the control strategy developed by the first agent, since the smaller wake deficit after the second turbine leads to a smaller induced velocity at the second turbine. From a more general perspective, the results presented in this chapter showed, that the agent is able to improve its behaviour. Furthermore it showed how reinforcement learning can be a valuable tool for scientific research, as it might find strategies not thought of by individual researchers as well as the possibility to try many different strategies at once. Not only can it help discover new control strategies but in this, hint at new physical phenomena not yet considered. On the other hand, it also showed some of the limiting factors. The training, even of an agent with a small network, takes a lot of computational effort and optimality can not be guaranteed. In the field of control strategies for wind parks, it was shown that the helix control has great potential to improve total generated power. It could also be shown that this strategy can be extended to bigger wind parks by phase shifting the helices of the turbines. To further the understanding of the phenomena caused by a helical wake, a thorough assessment of the loads will have to be made as well as simulations and field tests of bigger wind parks. Additionally the influence of a sheared boundary layer has not yet been studied, which will likely have a crucial influence, as the shear also influences the wake meandering.

4.4. Learning a New Behaviour

4.4.1. Training with a long episode and high discount factor

Since it could be shown, that the agent can improve the static parameters of a control strategy, the next step is to attempt to train an agent to learn a new behaviour. As described in 3.2.4, three agents controlling the generator torque of the three turbines are trained. First, an agent with a long episode and a high discount factor will be examined. The long episode is chosen to allow the agent to develop lower frequency strategies. As was noted in subsection 3.2.4, a high discount factor is chosen from an analysis of the physical timescales of the wind park.

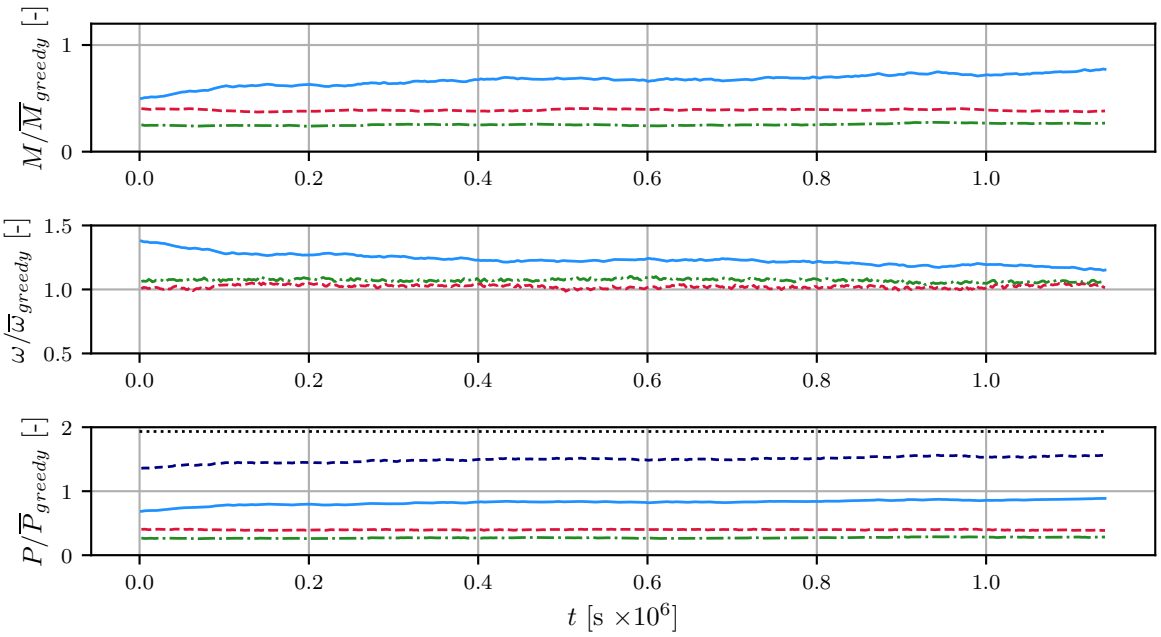

Figure 4.23.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Training of agent with $\gamma = 0.99$ and for episodes with $N_{a,e} = 1500$.

The timeseries of the generator torques, the angular velocities and the generated power of the three turbines as well as the total generated power are shown in Figure 4.23. It shows that after 1.2×10^6 s the performance by the agent is still significantly worse than that of a greedy controller. Therefore the training was stopped at this point. Furthermore, no improvement in total power is visible for 1×10^5 s. The figure shows, that the torque of the first turbine is steadily increased throughout the training, while the torque of the second and third turbine show little change.
To analyze the evolution of the control strategy, despite it not being successful, Figure 4.24 shows generator torque, angular velocity and generated power by the turbines when controlled by the agent after 12 updates, after half of training time and after the full training time. The comparison

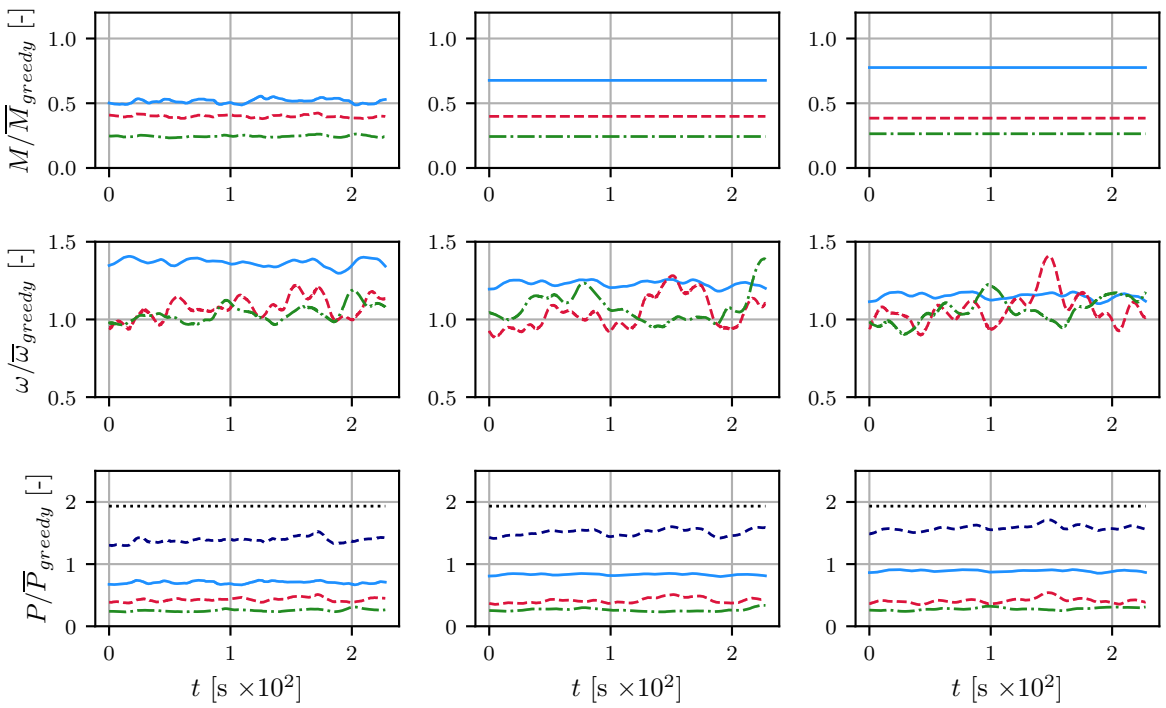

Figure 4.24.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Evolution of control strategy of agent with $\gamma = 0.99$ and for episodes with $N_{a,e} = 1500$. Left column is control strategy in the beginning of training, central column after half of the training and right column after training has finished.

of the control strategies shows that the agent evolves to set a constant generator torque at all three turbines. This is already visible after half of the training time. Furthermore, as was seen in Figure 4.23, the generator torque of the first turbine increases, while the torque at the other two turbines does not change after half of the training. Since the power of the first turbine is greatest, it appears reasonable that control of that turbine is improved fastest. A static generator torque differs from greedy control, but considering that the inflow has a uniform mean value, such a strategy still can be reasonable. However, since the total generated power only reached about three quarter of the total power of a greedy-controlled park, this strategy will not be investigated further. The major drawback of this agent was the slow evolution. One of the possible reasons for this is the low number of updates of the agent per simulated time du to long episodes. Furthermore, the reason for the long episode was the ability to develop lower frequency behaviour, which did not occur. Therefore another agent with a shorter episode was trained.

4.4.2. Training with a short episode and high discount factor

To increase the number of updates per simulated time, an agent with a number of actions per episode of 500 was trained.

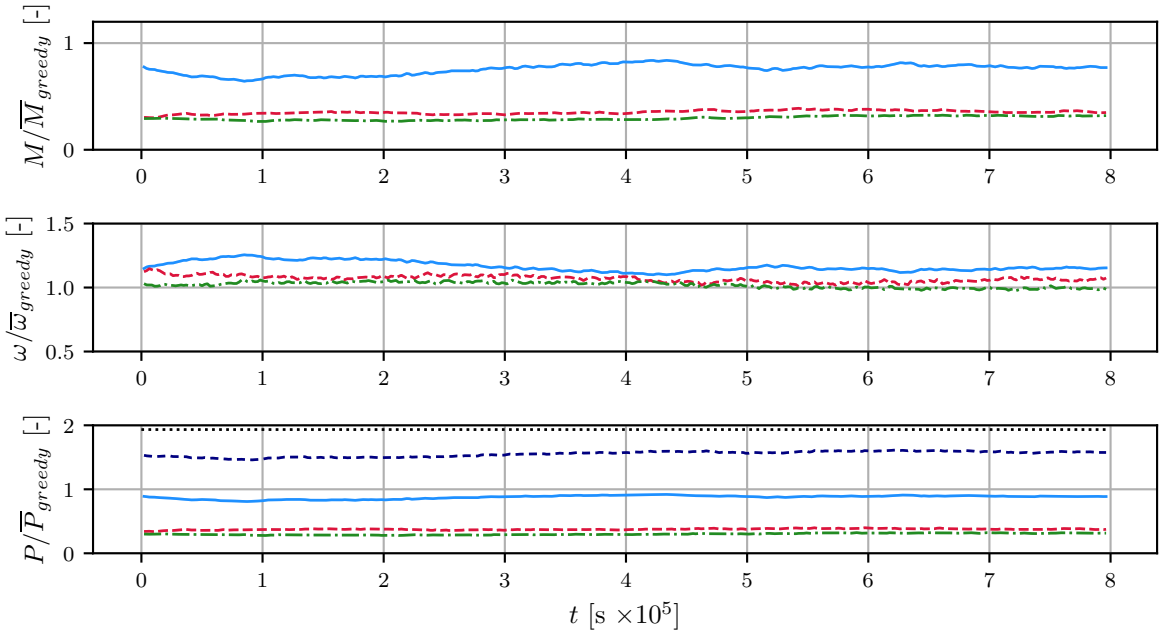

Figure 4.25.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Training of agent with $\gamma = 0.99$ and for episodes with $N_{a,e} = 500$.

The time series of the generated power, angular velocity and generator torque are shown in Figure 4.25. It shows that total generated power, after an initial decrease, increased for about half of the training. After that, total power did not increase, but the mean of the controlled variables, the generator torques, still changed. Therefore training was continued further, but the agent returned to the same state multiple times. Thus training was stopped after 8×10^5 s. The comparison to the previous agent with a longer episode shows, that the behaviour of the agent changes faster. Yet it is not able to develop a strategy that performs as good as a greedy controller. While the first turbine performs almost as good as a single greedy controlled turbine, the second and the third turbine perform considerably worse. The plot also shows that little changes occur in the control strategy for these two turbines. Again, this can be explained by the larger power produced at the first turbine. To find out more about the developed strategy and how it changed throughout the training, Figure 4.26 shows generator torques, angular velocities and generated powers per turbine as well as total generated power for parks controlled by agents at the beginning of the training, after half the training and after training has been completed. It shows that this agent also developed the strategy to set constant generator torques. It also shows that the first turbine performs almost as good as

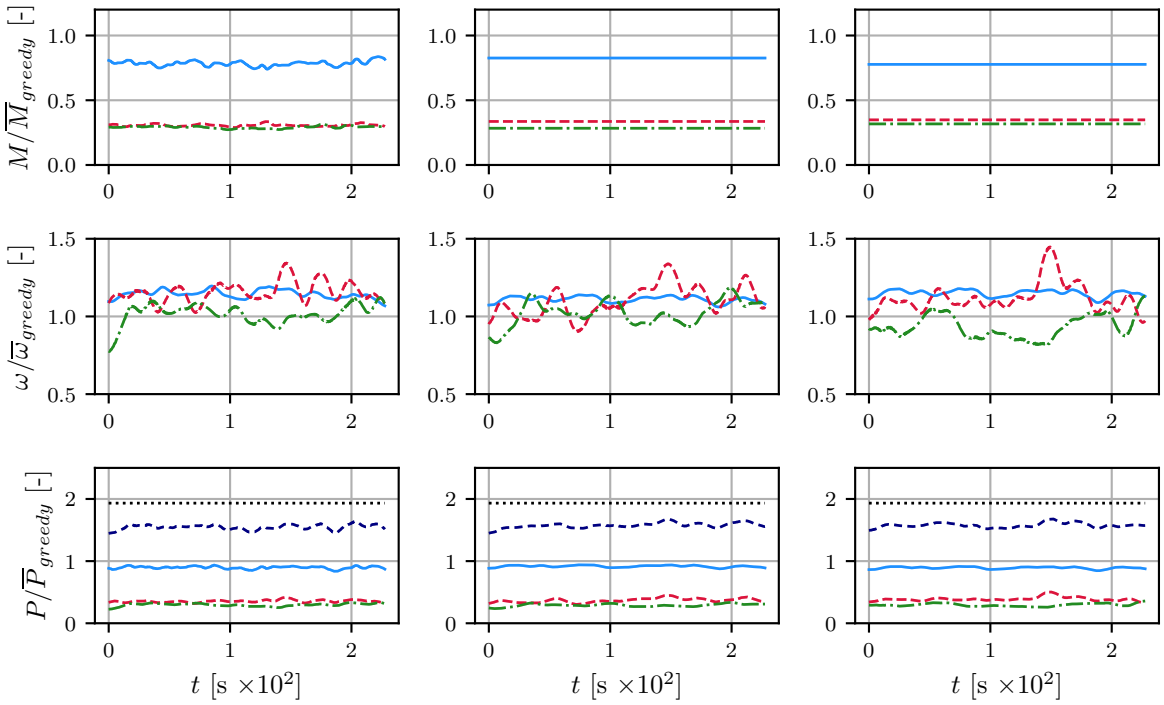

Figure 4.26.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Evolution of control strategy of agent with $\gamma = 0.99$ and for episodes with $N_{a,e} = 500$. Left column is control strategy in the beginning of training, central column after half of the training and right column after training has finished.

a greedy controlled turbine. In comparison to the greedy controller the agent sets a lower generator torque at all of the turbines, therefore they have higher angular velocities. Thus the thrust forces exerted by the turbines are also higher, resulting in a higher wake deficit. Consequentially the downstream turbines perform worse. However it is not clear why the agent does not set higher torques to reduce thrust. As was explained in subsection 3.2.4, the high discount rate leads to consideration of rewards that occur far in the future. This leads to a noisy signal, where influences of later actions also play a role. In combination with a turbulent flow that is by definition noisy, this might inhibit optimization as no strong correlation between return, which is the discounted sum of total generated power, and generator torque at one point in time exists. To reduce the noise in the signal an agent is trained with a lower discount rate.

4.4.3. Training with a short episode and low discount factor

While a discount rate of $\gamma = 0.99$ allows for the return to include the influence the action at the first turbine has with power produced by the second turbine after that information has been carried there, it might also lead to incorporation of too many time steps into the return, thus making it impossible for the agent to improve its behaviour. Therefore an agent with a discount rate of $\gamma = 0.95$ was trained as well.

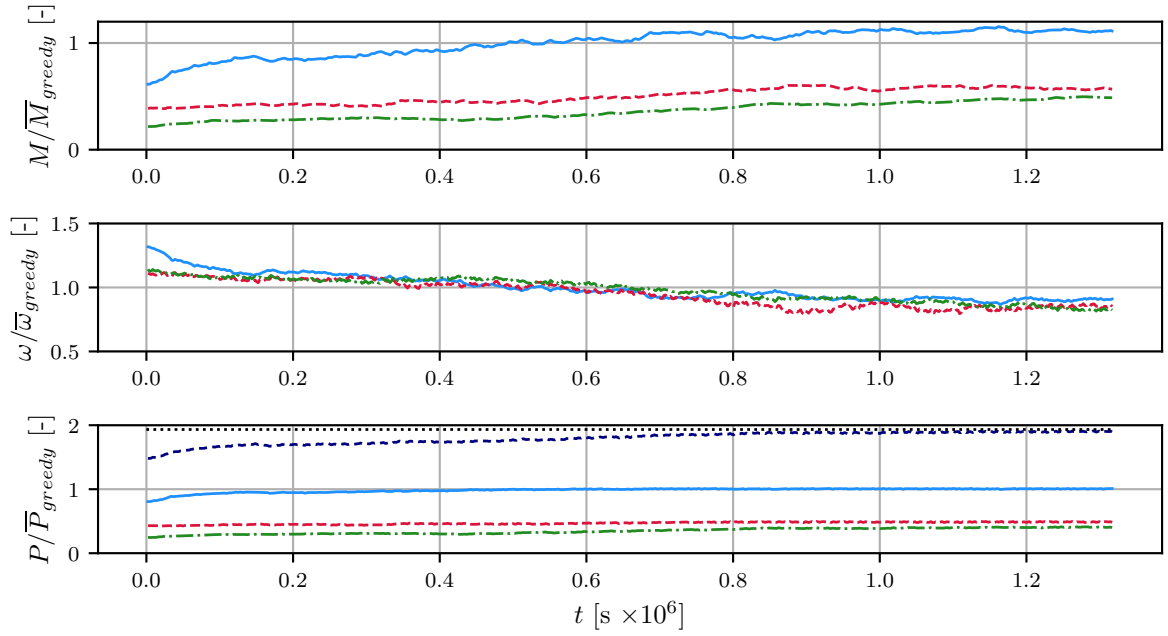

Figure 4.27.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Training of agent with $\gamma = 0.95$ and for episodes with $N_{a,e} = 500$.

The time series of the training of that agent is shown in Figure 4.27. This agent did learn a behaviour that performs almost as good as the greedy control. While the gain in total power in the beginning of training can be attributed to an increase in power produced by the first turbine, the third turbine also improves continuously. The second turbine also improves in power with time, although the gains are smaller. The training is conducted for 1.3×10^6 s, which is longer than for the previous agents, but the strategy had not finished improving after 8×10^5 s. A look at the generator torques shows that the torque at all three turbines increased throughout the training, which in turn lead to a decrease in angular velocity at all three turbines. The overall development of power and torque looks similar to that of the first agent as shown in Figure 4.23. As length of episode and discount factor both influence the return, both these agents seem to have a good ratio of these factors. As the long episode agent did not develop a dynamic control strategy, the long episode does not offer advantages, but only increases the simulated time per update.

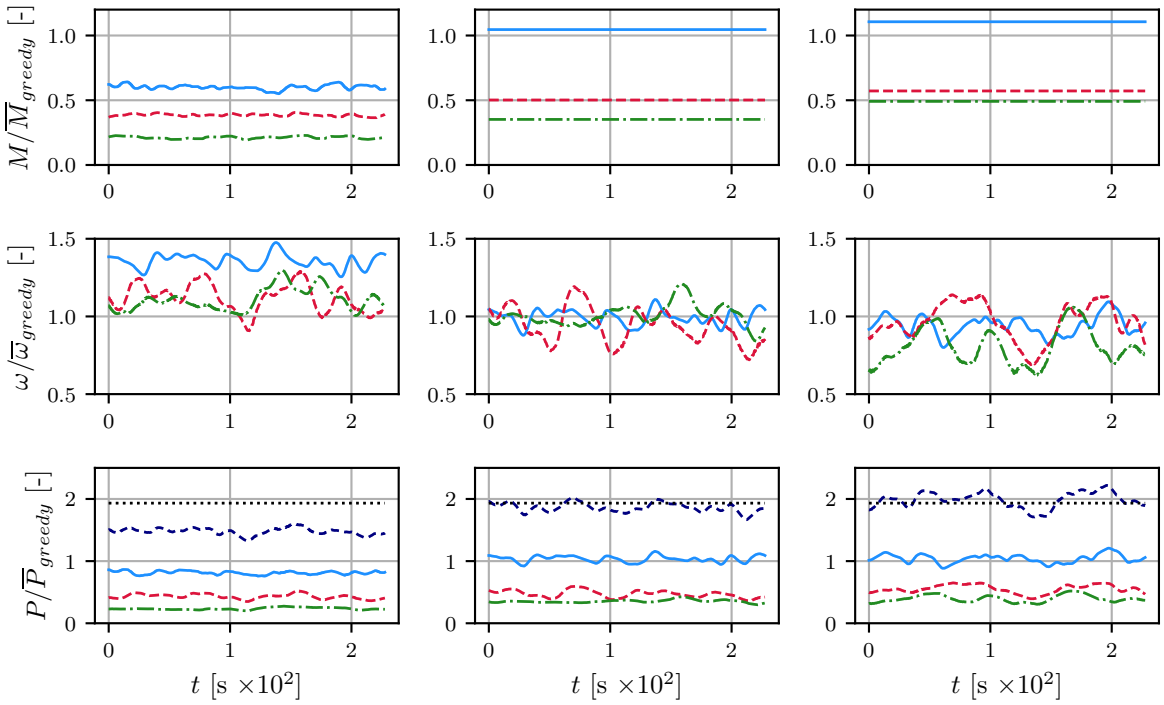

Figure 4.28.: Turbine 0 Turbine 1 Turbine 2 Total Greedy. Evolution of control strategy of agent with $\gamma = 0.95$ and for episodes with $N_{a,e} = 500$. Left column is control strategy in the beginning of training, central column after half of the training and right column after training has finished.

The evolution of the strategy is shown in Figure 4.28. It shows that this agent, like the other two previously studied, sets a constant generator torque after half of the training and does not react to the flow any more. The strategy in the second half of the training only changes in the constant torque that is set. As was visible in Figure 4.27, the torques set all increase in value. The final control strategy sets a higher torque at the first turbine than the mean torque set by a greedy controller. Furthermore it shows that, since a constant generator torque is set, angular velocities fluctuate due to the turbulent inflow. Furthermore these fluctuations are also visible in the generated power. As was mentioned in the analysis of the first generator torque controlling agent, one reason for the development of the static strategy might be that the mean inflow velocity is constant and only superimposed with fluctuations. Thus in the mean this strategy is comparable to a very slow greedy controller.

4.4.4. Analysis of flow

In Table 4.4 mean values as well as relative changes in mean and standard deviation compared to a greedy controlled case are shown for power and aerodynamic torque. The total power production

Table 4.4.: Mean, relative difference in mean and relative in standard deviation of power and aerodynamic moment of a trained torque-controlling agent in comparison to the greedy-control case.

	P			M_{aero}		
	mean	rel. mean	rel. std	mean	rel. mean	rel. std
Total	$9.65\,\mathrm{MW}$	$-1.42\,\%$	$-19.3\,\%$	-	-	-
Turbine 0	$5.1\,\mathrm{MW}$	$+0.837\,\%$	$+6.28\,\%$	$3.53 \times 10^3\,\mathrm{kNm}$	$+10.6\,\%$	$+14.8\,\%$
Turbine 1	$2.48\,\mathrm{MW}$	$-3.23\,\%$	$-22.5\,\%$	$1.82 \times 10^3\,\mathrm{kNm}$	$-9.65\,\%$	$-4.19\,\%$
Turbine 2	$2.07\,\mathrm{MW}$	$-4.55\,\%$	$-26.1\,\%$	$1.57 \times 10^3\,\mathrm{kNm}$	$-13.2\,\%$	$-6.25\,\%$

is 1.42 percent lower than that of a greedy controlled park. This is due to decreases in power produced by the second and third turbine, while first turbine produces negligibly more. But while the primary objective, the increase in total power could not be reached, the quality of overall power significantly improved with a reduction in standard deviation of 19.3 percent. This is due to a decrease in fluctuations at the second and third turbine, while fluctuations at the first turbine increased. The same is true for the aerodynamic moment. The mean as well as the standard deviation increased at the first turbine while they were considerably reduced at the second and third turbine. As especially power quality is of growing concern, a more thorough analysis of the flow might offer insights into how this reduction is achieved.

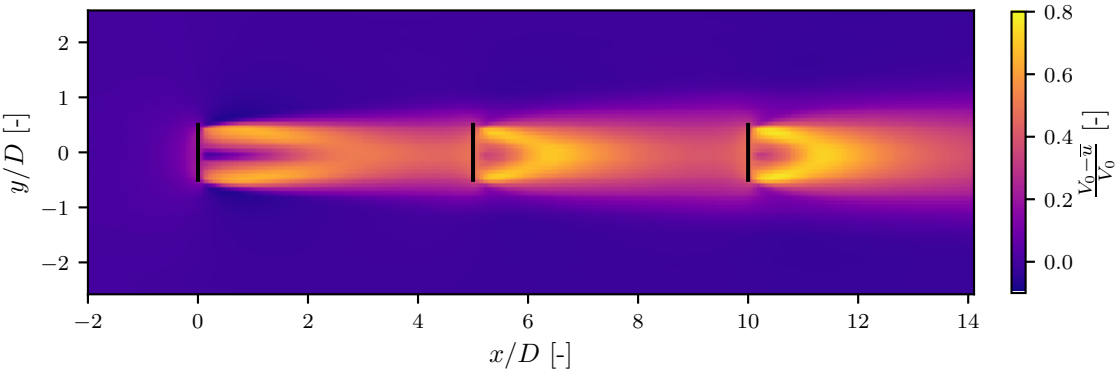

Figure 4.29.: —— Turbine. Mean wake deficit of ANN-controller.

To this end, Figure 4.29 shows the wake deficit in the domain. The deficit is largest in the wake of the third turbine and smallest after the first turbine. In comparison to the results of the helix control, the wake is not as wide. The deficit in front of the second and third turbine is larger, which explains why the generated power controlled by this controller is lower than the power generated by both of the helix controllers.

The difference in mean wake deficit of the ANN controller to the greedy controller is shown in Figure 4.30. While the wake deficit directly behind the first turbine is lower, when the park is controlled

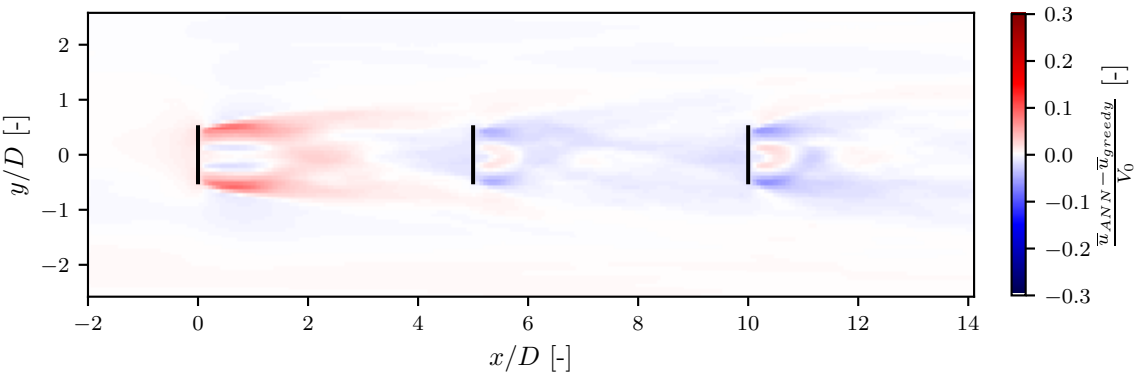

Figure 4.30.: —— Turbine. Mean wake deficit of the ANN-controller compared to the mean wake deficit of a greedy-controlled park.

by the ANN, it is higher in front of the second turbine and after the second and third turbine. Therefore the mixing of slow fluid in the wake and undisturbed fluid seems to be reduced. However, the increase in the near wake velocity after the first turbine is still of interest if turbines can not be placed as far apart as in this case. In such a case, a slow reacting greedy controller might still offer benefits for increasing power production.

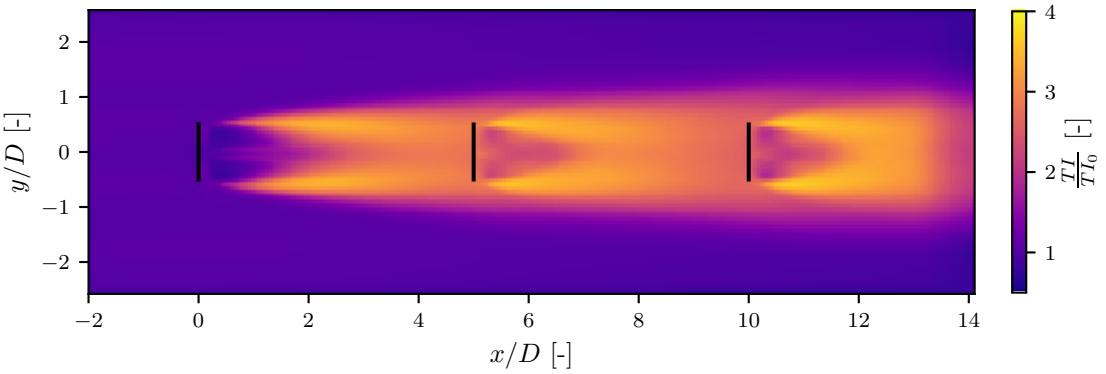

Figure 4.31.: —— Turbine. Turbulence intensity of ANN controller compared to turbulence intensity at the inlet.

In Figure 4.31 the turbulence intensity is shown. Here the difference to the helix control becomes very apparent. In the wake of the first turbine the turbulence intensity in the tip vortices of the blades is significantly lower and the wake further down is not as spread out. Instead, the tip vortices hit the blade tips of the second turbine. Overall the turbulence intensity is significantly lower, reducing the mixing of the wake with the undisturbed surrounding fluid. In the second and third wake the turbulence intensity increases and the tip vortices are more spread out.
Lastly, the difference in turbulence intensity in a park controlled by the ANN controller to a park that is greedy-controlled is shown in Figure 4.32. The turbulence intensity of the wake after the first

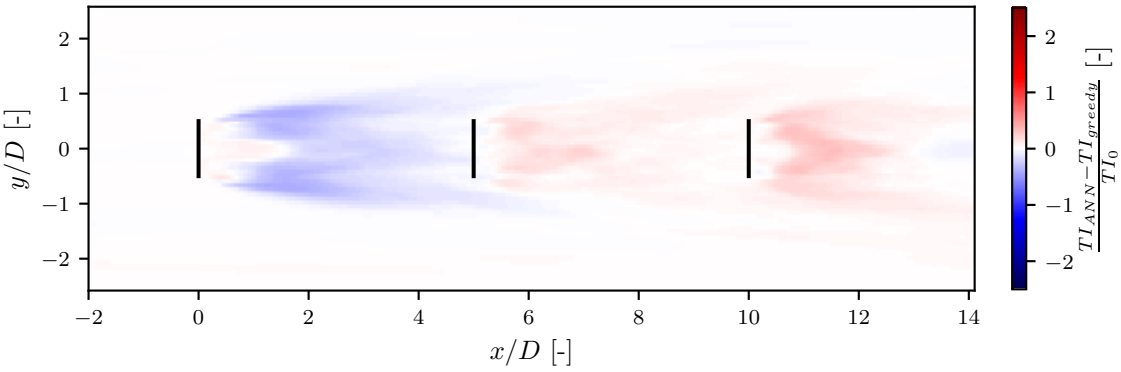

Figure 4.32.: —— Turbine. Turbulence intensity of optimized helix control.

turbine is significantly lower in the park controlled by the ANN controller. On the one hand this is probably one of the most important reasons why the wake-mixing is so significantly smaller and therefore power production by the second and third turbine decreased, this might also be a reason for a decrease in fluctuations in power. Therefore, this control strategy of a slow greedy controller might be a possible form improving power quality when parks are de-rated due to overproduction, a scenario that will become more likely in the future, if the total production capacities have increased. In the wake of the second and third turbine turbulence intensity has slightly increased, although the turbulence intensity of the inflow of the third turbine has not changed.
Overall the development of a new strategy was not successful. None of the agents were able reach the performance of a greedy controller, despite long training. However, since the coupling of the wind park to the agent worked and the learning algorithm also worked, the problems have to lay in the formulation of the problem, that is, how reward, state and action are connected to the wind park.

Analysis of the difficulties of training an RL-agent to control a wind park

Since reinforcement learning has never been applied to an active flow control problem with the properties of this case, namely a fully turbulent flow with multiple actuators separated in space and time, many aspects of this problem have not yet been explored. As was seen by the comparison of agents trained with different episode lengths, the ratio of episode length to discount factor is very important to the ability to improve behaviour. More general, a high correlation between action and return is necessary for learning. Three possible ways the current phrasing of the problem decreases this correlation can be identified.
The first can be found by observing the evolution of the control strategy of all three agents trained. While the agents with their initial policy show some reaction to input of the network, that is the velocity probes and the angular velocity of the rotors, this is not visible any more after half of the

training time. This can be explained by the fact, that the action set during the beginning of the training is, on average, much lower than the generator torque of the greedy controller for example. Therefore the return does not actually depend on the action, since a higher value of the action is always better than a lower value. Therefore all the weights will eventually be set to zero and only the biases of the last layer will be of any importance. Once that is the case, the influence of the state of the environment has been eliminated can not come back. Therefore more attention will have to be paid to the translation from the agent's action to the actual generator torque set at the turbine, so that the actions chosen in the beginning are closer to the values necessary for optimal behaviour. Multiple ways to achieve this are possible. One could use an additional layer of nodes with a linear activation function, however, this would require additional training. Another possible solution is a non-dimensionalization based on mean generator torques used by a greedy controller in preliminary studies. Additionally, the agent could also be trained via supervised learning to behave like a greedy controller before applying reinforcement learning to develop new strategies.
The second problem was already identified in subsection 3.2.4. The reward is the total generated power. However, the three different turbines will produces different amounts of power, especially in the beginning of the training. Therefore the first turbine will be improved the fastest, since increasing the efficiency of this turbine yields the greatest gain in reward. The obvious solution would be to either weigh the turbines differently or change the reward to the sum of the power coefficients. However, optimizing power coefficients would result in a greedy strategy, as this is its definition. The same problem arises when using a weighted sum. If the agent would maximize the reward, this would not be the same as maximizing the total power. Thus these two paths can not be the solution.
The closer analysis of the reward also revealed another problem already mentioned in subsection 3.2.4. The power at any moment of time the first turbine only depends on the actions of the agent at the first turbine taken at that moment and in the past. The power of the second turbine depends on the actions of the agent taken at the second turbine at this moment and in the past and on the actions taken by the agent at the first turbine, but shifted by the time it takes for that influence of the action to travel from the first turbine to the second turbine. The similar argument can be made for the third turbine. Assuming that the influence of an action of the power only lasts for one time step, the return can be phrased the following way:

$$G_t(\mathbf{a}_t, \Gamma) = P_{0,t}(\mathbf{a}_t) + P_{1,t+T_{travel}}(\mathbf{a}_t)\gamma^{T_{travel}} + P_{2,t+2T_{travel}}(\mathbf{a}_t)\gamma^{2T_{travel}} + \Gamma, \tag{4.8}$$

with Λ being all other influences. The larger γ, the larger Γ, since more time steps are included in the calculation of G_t. On the other hand, the weight of the power at the second and third turbine is also directly proportional to γ. One possible solution to both problems would be the use of independent MDPs, that control one turbine each and have different rewards. Changing the reward of the first

environment to

$$\mathrm{R}_{0,t} = P_{0,t} + P_{1,t+T_{travel}} + P_{2,t+2T_{travel}} \tag{4.9}$$

would allow for a lower discount rate and eliminate the time shifted influence of actions. The other agents would be seen as part of the environment. Furthermore, this requires no weighting of generated power. However, this would require significant changes to the framework developed for this work and was therefore not implemented.

5. Conclusion

In this work reinforcement learning was applied to control of a wind farm simulation utilizing the actuator line method and LES-LBM in order to increase the total generated power of the farm. Two approaches were tested. First, parameters of an already existing dynamic control strategy were improved. This was very successful, with an increase in total generated power by up to 18.2 percent. Additionally, further insight into the physical phenomena of the dynamic control was gained. Second, the agent controlled the turbine directly. This approach was not able to improve total generated power and possible reasons for this were identified, among them the separation in time and space of the turbines. A possible solution of this problem by using multiple agents and environments was proposed.

In a first step, the theoretical basis for RL, ANNs, turbine modelling and control as well as LBM was laid. Also a more complete deduction for a second order refinement scheme for the cumulant LBM was given, which was not available in the literature. Furthermore a literature review of advanced dynamic induction control for wind farms and of active flow control via reinforcement learning was conducted. It was found, that the coupling of AFC and RL have only been conducted for laminar, two-dimensional problems.

In the second part, the framework developed for this work as well as details of the implementation and design choices were explained. Furthermore, the setup of the simulations was explained. The wind park features three turbines with a distance of five rotor diameters and turbulent inflow with a turbulence intensity of five percent. The different agents that were trained were also explained. Two agents were trained to improve the helix control strategy, which was taken from literature. One of the agents had a network of three layers with a width of 400 nodes, the other agents network consists of three layers with a width of ten nodes. Three agents were trained to learn a new control strategy by directly controlling the generator torque of the turbines. The training different in the length of the episodes, the first agent was trained with episodes of 1500 actions, the second and the third agent with episodes containing 500 actions. The return for the first and the second agent was discounted with a discount rate of $\gamma = 0.99$, while the return for the third agent was discounted with a discount rate of $\gamma = 0.95$.

Finally the results were discussed. First the implementation of the different parts of the simulations were validated. Then preliminary studies with a simple aerodynamic model of a single turbine were conducted. They showed that the training of an agent requires $\mathcal{O}(10^6\mathrm{s})$ of simulated time. The results from the optimization of the parameters of the helix control strategy were discussed. The two agents resulted in similar behaviour with one difference. The agent with the smaller network set a negative amplitude, resulting in a phase shift of 180° of the wake helix of the first turbine. With this shift an increase of 18.2 percent in total generated power was found, while the other agent, which did not feature the phase shift, resulted in an increase of 6.75 percent. It was found that this differ-

ence in total generated power was caused by a phase shift of the helix caused by the first turbine and the helix caused by the second turbine. In the first case, the helices were aligned, resulting in an increase of the radius of the helix. In the second case, the helices are shifted by $180°$, which results in a destruction of the helix after the second turbine, decreasing power production at the second and the third turbine. Based on this finding it was proposed to shift the helix of the second turbine to align with the first turbine, which found an increase to that of the park controlled by agent with the small network. This provided further insight into the physical phenomena governing the helix approach, which had only been tested for a two-turbine park, of which only the first turbine was controlled with the helix approach. The findings in this work show how an application to a larger park can be offer significant increases in generated power.

Lastly, the results of the training of the three agents directly controlling the generator torque of the turbines were evaluated. None of the agents were able to develop a strategy that performed better than a greedy-controlled park. Still, the analysis of the flow in a park controlled by one of the agents showed, that a slower reacting greedy controller might be able to reduce fluctuations in generated power. Three possible reasons were found why the agents could not find a good strategy in the current setup. All three agents developed a static strategy and did not react to the state of the environment any more. This was attributed to an inadequate transformation of the actions of the agent to the control quantities of the turbine. Three solutions were proposed. First, the a-priori values used for the transformation could be improved by using estimates based on greedy control. Second, an additional layer of linear nodes could be used in the network. Third, the agent could be trained to behave like a greedy controller by supervised learning. The second and third problem identified are related to the reward. On the one hand, the reward, which was defined to be the sum of the generated power of the three turbines, favours improving the first turbine, since this turbine produces the most power. Second, the control of the first turbine influences the second and third turbine but only after a long period of time. During this period of time, the control has no influence on the current state of the environment. If the influence on the downstream turbines is captured by the reward, it also includes many other time steps. If it is not captured, the optimization goal reduces to a greedy strategy. To tackle both of these problems it was proposed to change the setup to the use of a separate environment and agent for each turbine as well as modifying the the reward to include only time steps which can be relevant.

Three main conclusions regarding the application of RL to AFC can be drawn from this work. First, reinforcement learning coupled to active flow control requires a lot of simulated time, to apply it to fully turbulent flows therefore is computationally very expensive. The cumulant LBM with the extension of second order refinement is a method well-suited for this task. Additionally, the hardware requirements of RL as well as LBM are similar, since both can be greatly accelerated by the use of GPUs. Second, applying RL to realistic problems requires many design choices from the great number of parameters in the learning algorithm to details in the implementation regarding

non-dimensionalization. Since this was the first application of RL to such a realistic problem, no literature was available to rely on. Third, this methodology has the potential of discovering new control strategies or improving already existing strategies as well as hinting researchers at physical phenomena, which where not previously observed.

The results regarding the helix approach showed, that it has great potential to increase the power production of small wind parks but the newly gained understanding of the physics of this approach suggest that it can easily be extended to larger parks, if the phase shift of the helices is accounted for. To further prove the feasibility of this approach for control of real-world wind parks, simulations including a sheared boundary layer will have to be conducted, as the shear greatly influences the movement of the wake. Additionally a thorough study of the loads will be necessary, as they have only been assessed through thrust and aerodynamic moment until this point.

Future works on applying RL to wind farm control will have to solve the problems discussed above. Especially the problem of time shifted actions will have to be addressed to turn RL into a more useful tool for a broader range of problems. A possible solution to this was already suggested in this chapter. It might also be necessary to gain further understanding of RL in turbulent flows by studying simpler problems, such as a single turbine. The analysis of the control strategy developed by the RL-agent is further hindered by the black-box character of the neural network. To this end it might be feasible to use different approximation functions for the policy.

Bibliography

[1] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[2] M. Abkar, A. Sharifi, and F. Porté-Agel. Wake flow in a wind farm during a diurnal cycle. *Journal of Turbulence*, 17(4):420–441, 2016.

[3] H. Asmuth, H. Olivares-Espinosa, and S. Ivanell. Actuator line simulations of wind turbine wakes using the lattice Boltzmann method. *Wind Energy Science*, 5(2):623–645, 2020.

[4] H. Asmuth et al. The Actuator Line Model in Lattice Boltzmann Frameworks: Numerical Sensitivity and Computational Performance. *Journal of Physics: Conference Series*, 1256:012022, 2019.

[5] V. Belus et al. Exploiting locality and translational invariance to design effective deep reinforcement learning control of the 1-dimensional unstable falling liquid film. *AIP Advances*, 9(12):125014, 2019.

[6] S. Boersma et al. A tutorial on control-oriented modeling and control of wind farms. In *2017 American Control Conference (ACC)*. 2017 American Control Conference (ACC), pages 1–18, 2017.

[7] S.-P. Breton et al. A survey of modelling methods for high-fidelity wind farm simulations using large eddy simulation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2091):20160097, 2017.

[8] C. Coreixas, B. Chopard, and J. Latt. Comprehensive comparison of collision models in the lattice Boltzmann framework: Theoretical investigations. *Physical Review E*, 100(3):033305, 2019.

[9] H. B. Demuth et al. *Neural Network Design*. Martin Hagan, Stillwater, OK, USA, 2nd edition, 2014.

[10] J. A. Frederik et al. The helix approach: Using dynamic individual pitch control to enhance wake mixing in wind farms. *Wind Energy*, 2020.

[11] J. A. Frederik et al. Periodic dynamic induction control of wind farms: proving the potential in simulations and wind tunnel experiments. *Wind Energy Science*, 5(1):245–257, 2020.

[12] E. Gaertner et al. IEA Wind TCP Task 37: Definition of the IEA 15-Megawatt Offshore Reference Wind Turbine. NREL/TP-5000-75698, 1603478, National Renewable Energy Laboratory, 2020.

[13] P. Garnier et al. A review on Deep Reinforcement Learning for Fluid Mechanics, 2019.

[14] B. Gaudet, R. Linares, and R. Furfaro. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, 65(7):1723–1741, 2020.

[15] M. Gehrke, C. F. Janßen, and T. Rung. Scrutinizing lattice Boltzmann methods for direct numerical simulations of turbulent channel flows. *Computers & Fluids*. Ninth International Conference on Computational Fluid Dynamics (ICCFD9), 156:247–263, 2017.

[16] M. Geier and A. Pasquali. Fourth order Galilean invariance for the lattice Boltzmann method. *Computers & Fluids*, 166:139–151, 2018.

[17] M. Geier, A. Pasquali, and M. Schönherr. Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion part I: Derivation and validation. *Journal of Computational Physics*, 348:862–888, 2017.

[18] M. Geier et al. The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 70(4):507–547, 2015.

[19] S. Ghosh et al. Contextual LSTM (CLSTM) models for Large scale NLP tasks, 2016.

[20] J. P. Goit and J. Meyers. Optimal control of energy extraction in wind-farm boundary layers. *Journal of Fluid Mechanics*, 768:5–50, 2015.

[21] S. Guadarrama et al. TF-Agents: A library for reinforcement learning in TensorFlow, 2018.

[22] M. O. L. Hansen. *Aerodynamics of Wind Turbines*. Earthscan, London ; Sterling, VA, 2nd ed edition, 2008. 181 pages.

[23] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[24] L. Hockstad and L. Hanel. Inventory of U.S. Greenhouse Gas Emissions and Sinks. EPA 430-R-20-002, Environmental Protection Agency, 2018.

[25] O. Hoegh-Guldberg et al. Impacts of 1.5°C of Global Warming on Natural and Human Systems, IPCC, 2019.

[26] A. A. M. Holtslag et al. Stable Atmospheric Boundary Layers and Diurnal Cycles: Challenges for Weather and Climate Models. *Bulletin of the American Meteorological Society*, 94(11):1691–1706, 2013.

[27] International Energy Agency. *Global Energy Review 2020: The Impacts of the Covid-19 Crisis on Global Energy Demand and CO2 Emissions*. OECD, 2020.

[28] C. F. Janßen et al. Validation of the GPU-Accelerated CFD Solver ELBE for Free Surface Flow Problems in Civil and Environmental Engineering. *Computation*, 3(3):354–385, 3, 2015.

[29] J. Jonkman et al. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.

[30] J. C. Kaimal and J. J. Finnigan. *Atmospheric Boundary Layer Flows: Their Structure and Measurement*. Oxford University Press, New York, 1994. 289 pages.

[31] S. K. Kang and Y. A. Hassan. The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows. *Journal of Computational Physics*, 232(1):100–117, 2013.

[32] A. C. Kheirabadi and R. Nagamune. A quantitative review of wind farm control with the objective of wind farm power maximization. *Journal of Wind Engineering and Industrial Aerodynamics*, 192:45–73, 2019.

[33] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, 2017.

[34] T. Krüger et al. *The Lattice Boltzmann Method: Principles and Practice*. Graduate Texts in Physics. Springer International Publishing, Cham, 2017.

[35] K. Kutscher, M. Geier, and M. Krafczyk. Multiscale simulation of turbulent flow interacting with porous media based on a massively parallel implementation of the cumulant lattice Boltzmann method. *Computers & Fluids*, 193:103733, 2019.

[36] R. Löhner. Towards overcoming the LES crisis. *International Journal of Computational Fluid Dynamics*, 33(3):87–97, 2019.

[37] J. Mann. Wind field simulation. *Probabilistic Engineering Mechanics*, 13(4):269–282, 1998.

[38] A. R. Meyer Forsting, G. R. Pirrung, and N. Ramos-García. A vortex-based tip/smearing correction for the actuator line. *Wind Energy Science*, 4(2):369–383, 2019.

[39] W. Munters and J. Meyers. Towards practical dynamic induction control of wind farms: analysis of optimally controlled wind-farm boundary layers and sinusoidal induction control of first-row turbines. *Wind Energy Science*, 3(1):409–425, 2018.

[40] K. Nilsson et al. Large-eddy simulations of the Lillgrund wind farm. *Wind Energy*, 18(3):449–467, 2015.

[41] E. Örtl. *Entwicklung der spezifischen Kohlendioxid-Emissionen des deutschen Strommix in den Jahren 1990 - 2019*. Umweltbundesamt.

[42] G. R. Pirrung et al. A coupled near and far wake model for wind turbine aerodynamics. *Wind Energy*, 19(11):2053–2069, 2016.

[43] J. Rabault and A. Kuhnle. Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Physics of Fluids*, 31(9):094105, 2019.

[44] J. Rabault et al. Deep Reinforcement Learning achieves flow control of the 2D Karman Vortex Street, 2018.

[45] J. Rabault et al. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019.

[46] K. Rohrig et al. Powering the 21st century by wind energy—Options, facts, figures. *Applied Physics Reviews*, 6(3):031303, 2019.

[47] M. Schönherr. *Towards Reliable LES-CFD Computations Based on Advanced LBM Models Utilizing (Multi-) GPGPU Hardware*. PhD Thesis, TU Braunschweig, Institut für rechnergestützte Modellierung im Bauingenieurswesen, 2015.

[48] J. Schulman et al. Trust Region Policy Optimization. In *International Conference on Machine Learning*. International Conference on Machine Learning, pages 1889–1897, 2015.

[49] J. Schulman et al. Proximal Policy Optimization Algorithms, 2017.

[50] D. Silver et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, 2017.

[51] D. Silver et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

[52] J. N. Sørensen and W. Z. Shen. Numerical Modeling of Wind Turbine Wakes. *Journal of Fluids Engineering*, 124(2):393–399, 2002.

[53] J. N. Sørensen. *General Momentum Theory for Horizontal Axis Wind Turbines*, volume 4 of *Research Topics in Wind Energy*. Springer International Publishing, Cham, 2016.

[54] M. Steinbuch et al. Optimal control of wind power plants. *Journal of Wind Engineering and Industrial Aerodynamics*, 27(1):237–246, 1988.

[55] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. 526 pages.

[56] S. Verma, G. Novati, and P. Koumoutsakos. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences*, 115(23):5849–5854, 2018.

[57] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

A. The Cumulant Lattice Boltzmann Method

A.1. Derivation of cumulants

To provide a more detailed explanation of the cumulant LBM, a more thorough derivation is given here. Note that in the following, the linear index i of the discrete populations is split into three indeces i, j, k representing the lattice directions, running from -1 to 1. First, the discrete distribution is written in a continuous form and transformed to frequency space, making it indepent of the frame of reference:

$$f(\boldsymbol{\xi}) = \sum_{i,j,k} f_{i,j,k}\delta(ic_s - \xi)\delta(jc_s - \upsilon)\delta(kc_s - \zeta) \tag{A.1}$$

$$F(\boldsymbol{\Xi}) = \int_{-\infty}^{\infty} f(\boldsymbol{\xi}) e^{-\boldsymbol{\Xi}\cdot\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi}, \tag{A.2}$$

with $\boldsymbol{\Xi} = [\Xi, \Upsilon, \mathrm{Z}]^T$ as the frequency vector. This distribution is then used in the cumulant generating function:

$$C_{\alpha,\beta,\gamma} = c_s^{-(\alpha+\beta+\gamma)} \left.\frac{\partial^\alpha \partial^\beta \partial^\gamma}{\partial \Xi^\alpha \partial \Upsilon^\beta \partial \mathrm{Z}^\gamma} \ln F(\Xi, \Upsilon, \mathrm{Z})\right|_{\Xi=\Upsilon=\mathrm{Z}=0}. \tag{A.3}$$

However, from that form, the cumulants are not directly computable. Comparing the cumulant generating function to the moment generating function

$$M_{\alpha,\beta,\gamma} = c_s^{-(\alpha+\beta+\gamma)} \left.\frac{\partial^\alpha \partial^\beta \partial^\gamma}{\partial \Xi^\alpha \partial \Upsilon^\beta \partial \mathrm{Z}^\gamma} F(\Xi, \Upsilon, \mathrm{Z})\right|_{\Xi=\Upsilon=\mathrm{Z}=0} = \sum_{i,j,k} i^\alpha j^\beta k^\gamma f_{i,j,k} \tag{A.4}$$

shows their similarity. Computing the cumulants of an arbitrary function shows how they can be expressed in terms of moments, two examples are given here:

$$C_{200} = \frac{M_{200}}{M_{000}} - \frac{M_{100}^2}{M_{000}^2} \tag{A.5}$$

$$C_{110} = \frac{M_{110}}{M_{000}} - \frac{M_{100} M_{010}}{M_{000}^2}. \tag{A.6}$$

Cumulants can also be computed from central moments, which is requires less computations [18]. Cumulants of zeroth and first order stay constant throughout the collision. The cumulants upwards from order two can now be relaxed indivdually according to

$$C^*_{\alpha,\beta,\gamma} = (1 - \hat{\omega}_{\alpha,\beta,\gamma})\, C_{\alpha,\beta,\gamma}, \tag{A.7}$$

therefore each has its own relaxation frequency. The first frequency $\hat{\omega}_1$ governs the shear viscosity and $\hat{\omega}_2$ the bulk viscosity of the fluid by the same relation as (2.51). The rest of the frequencies can be chosen freely and are usually set to one, resulting fully relaxing the cumulants, however a parametrized version of the collision operation exists, that changes some of the higher order relaxation frequencies, leading to fourth order accurate diffusion in a fully resolved simulation [17]. In underresolved simulations the parameter introduced in the parametrization can be used to influence the diffusivity and therefore acting as an implicit subgrid scale model, however so far only empirical evidence exists for this.[18] The density, velocity and the terms of velocity gradient tensor can be identified with cumulants:

$$\delta\rho = M_{000} = C_{000} \tag{A.8}$$

$$u = \frac{M_{100}}{M_{000}} = C_{100} \tag{A.9}$$

$$D_x v + D_y u = -3\hat{\omega}_1 C_{110} = k_{xy} \tag{A.10}$$

$$D_x u = -\frac{\hat{\omega}_1}{2\rho}\left(2C_{200} - C_{020} - C_{002}\right) - \frac{\hat{\omega}_2}{2\rho}\left(C_{200} + C_{020} + C_{002} - \delta\rho\right) \tag{A.11}$$

$$D_x u - D_y v = -\frac{3\hat{\omega}_1}{2}\left(C_{200} - C_{020}\right) = k_{xx-yy}. \tag{A.12}$$

Note, that in this derivation the well-conditioned cumulant is used as described in the appendix of [18] and therefore the zeroth order cumulant is not the density but its deviation from unit density $\delta\rho$. Also the additional moments k_{xy} and k_{xx-yy} are introduced for use in section A.2.

A.2. Refinement

As was stated before, LBM is usually used on a uniform grid. However, often a variation in grid spacing is desirable, as coarser grids need less memory and have a coarser timestep, whereas a finer grid is able to resolve finer turbulent structures. To balance these two interests, the domain can be partitioned into blocks with different levels of refinement, as areas of high turbulence, which need high resolution, usually occur only in known parts of the domain. The question is now how to treat the borders of the blocks. In [47] the compact interpolation scheme is proposed that is second order accurate, therefore it is consistent with the order of accuracy of the LBM in general. However, the derivation given was found to be neither exhaustive, nor fully correct, as there seem to be misprints, making it difficult to follow. Therefore a more detailed derivation is given here, that is based on [47] as well as [35], which gave a version for an incompressible form of LBM, but suffers from similar issues. In each coarse timestep, a layer of coarse nodes is interpolated from fine nodes and two layers of fine nodes are interpolated from coarse nodes. The layers overlap, so that at no point in time invalid populations are propagated outside the overlap [47]. The node

being interpolated is called the receiver node while the nodes from which is interpolated is referred to as the donor node. Each receiver node has eight donor nodes. A local coordinate system in the center of cube is defined, with the donor nodes laying in the corners at $x = -0.5, 0.5$, $y = -0.5, 0.5$ and $z = -0.5, 0.5$.
First, a interpolation function for the density, $\delta\hat{\rho}$ and the velocities $\hat{u}$, $\hat{v}$ and $\hat{w}$ is defined. Note that the density only requires to be first order accurate:

$$\delta\hat{\rho} = d_0 + d_x x + d_y y + d_z z + d_{xy} xy + d_{xz} xz + d_{yz} yz + d_{xyz} xyz \tag{A.13}$$

$$\hat{u} = a_0 + a_x x + a_y y + a_z z + a_{xx} x^2 + a_{xy} xy + a_{xz} xz + a_{yy} y^2 + a_{yz} yz + a_{zz} z^2 + a_{xyz} xyz \tag{A.14}$$

$$\hat{v} = b_0 + b_x x + b_y y + b_z z + b_{xx} x^2 + b_{xy} xy + b_{xz} xz + b_{yy} y^2 + b_{yz} yz + b_{zz} z^2 + b_{xyz} xyz \tag{A.15}$$

$$\hat{w} = c_0 + c_x x + c_y y + c_z z + c_{xx} x^2 + c_{xy} xy + c_{xz} xz + c_{yy} y^2 + c_{yz} yz + c_{zz} z^2 + c_{xyz} xyz \tag{A.16}$$

To evualate the interpolation functions, constraints have to be defined. The velocity and density in the donor node place 32 constraints, since there are eight of each, thus nine more are required for the 41 degrees of freedom. The second derivative of velocities in the center of the local coordinate system are chosen as these additional constraints. They can be computed by taking the first order central difference of the terms of the velocity gradient tensor, which will be denoted as $D_x xu$ and so on. However, the terms in the main diagonal of the velocity gradient tensor are lengthy and include $\hat{\omega}_2$. This is somewhat undesirable and it was shown in (A.10) that differences of the terms can be expressed more directly in differences of cumulants. Therefore the remaining nine constraints are chosen like this:

$$\frac{\partial^2}{\partial x^2}\hat{v} + \frac{\partial^2}{\partial y \partial x}\hat{u} = D_{xx} v + D_{xy} u \tag{A.17}$$

$$2\frac{\partial^2}{\partial x^2}\hat{u} - \frac{\partial^2}{\partial x \partial y}\hat{v} - \frac{\partial^2}{\partial x \partial z}\hat{w} = 2D_{xx} u - D_{xy} v - D_{xz} w \tag{A.18}$$

$$2\frac{\partial^2}{\partial y^2}\hat{v} - \frac{\partial^2}{\partial x \partial y}\hat{u} - \frac{\partial^2}{\partial y \partial z}\hat{w} = 2D_{yy} v - D_{xy} u - D_{yz} w \tag{A.19}$$

$$2\frac{\partial^2}{\partial z^2}\hat{w} - \frac{\partial^2}{\partial y \partial z}\hat{v} - \frac{\partial^2}{\partial x \partial z}\hat{u} = 2D_{zz} w - D_{yz} v - D_{xz} u, \tag{A.20}$$

the missing five constraints by the off diagonal can easily be extrapolated.
Thus a system of linear equations is established that can be solved for the coefficients of the inter-

polation functions:

$$a_0 = \frac{1}{32}\sum_{x,y,z} -4D_{xx}u - 2D_{xy}v - 4D_{yy}u - 2D_{xz}w + -4D_{zz}u + 4u_{xyz} + 4xyv_{xyz} + 4xzw_{xyz} \tag{A.21}$$

$$= \frac{1}{32}\sum_{x,y,z} -x\left(k_{xx-yy} + k_{xx-zz}\right) - 2yk_{xy} - 2zk_{xz} + 4u_{xyz} + 4xyv_{xyz} + 4xzw_{xyz} \tag{A.22}$$

$$a_x = \frac{1}{2}\sum_{x,y,z} xu_{xyz} \tag{A.23}$$

$$a_{xx} = \frac{1}{8}\sum_{x,y,z} x\left(k_{xx-yy} + k_{xx-zz}\right) + 4xyv_{xyz} + 4xzw_{xyz} \tag{A.24}$$

$$a_{yy} = \frac{1}{8}\sum_{x,y,z} yk_{xy} - 4xyv_{xyz} \tag{A.25}$$

$$a_{xy} = 2\sum_{x,y,z} xyu_{xyz} \tag{A.26}$$

$$a_{xyz} = 8\sum_{x,y,z} xyzu_{xyz} \tag{A.27}$$

$$d_0 = \frac{1}{8}\sum_{x,y,z} \delta\rho_{xyz}, \tag{A.28}$$

in order to reduce the number of equations only unique descriptions are given, the other coefficients can easily extrapolated. Furthermore note that the coefficients of $\delta\hat{\rho}$ are the same as for the other interpolation functions with the exception of d_0. Thus the velocities are computed to second order. To arrive at equations for the second order cumulants, (A.10) - (A.12) are solved for cumulants. A term including the divergence of the velocity is neglected, since LBM is valid in a weakly compressible range and the term is thus much smaller.To compute the cumulants, the derivatves of $\hat{u}$, $\hat{v}$ and $\hat{w}$ are used and the constant terms in the derivatives are replaced with averaged values of second order moments. Also $\hat{\omega}_1$ has to be scaled with a factor σ_{rf} to account for the change in

refinement, it is 2 when scaling from fine to coarse and $1/2$ when scaling from coarse to fine:

$$A_{011} = \frac{\partial \hat{v}}{\partial z} + \frac{\partial \hat{w}}{\partial y} - b_z - c_y \tag{A.29}$$

$$= b_{xz}x + b_{yz}y + 2b_{zz}z + b_{xyz}xy + c_{xy}x + 2c_{yy}y + c_{yz}z + c_{xyz}xz \tag{A.30}$$

$$A_{101} = \frac{\partial \hat{u}}{\partial z} + \frac{\partial \hat{w}}{\partial x} - a_z - c_x \tag{A.31}$$

$$= a_{xz}x + a_{yz}y + 2a_{zz}z + a_{xyz}xy + 2c_{xx}x + c_{xy}y + c_{xz}z + c_{xyz}yz \tag{A.32}$$

$$A_{110} = \frac{\partial \hat{u}}{\partial y} + \frac{\partial \hat{v}}{\partial x} - a_y - b_x \tag{A.33}$$

$$= a_{xy}x + 2a_{yy}y + a_{yz}z + a_{xyz}xz + 2b_{xx}x + b_{xy}y + b_{xz}z + b_{xyz}yz \tag{A.34}$$

$$B = \frac{\partial \hat{u}}{\partial x} - \frac{\partial \hat{v}}{\partial y} - a_x + b_y \tag{A.35}$$

$$= 2a_{xx}x + a_{xy}y + a_{xz}z + a_{xyz}yz - b_{xy}x - 2b_{yy}y - b_{yz}z - b_{xyz}xz \tag{A.36}$$

$$C = \frac{\partial \hat{u}}{\partial x} - \frac{\partial \hat{w}}{\partial z} - a_x + c_z \tag{A.37}$$

$$= 2a_{xx}x + a_{xy}y + a_{xz}z + a_{xyz}yz - c_{xz}x - c_{yz}y - 2c_{zz}y - c_{xyz}xy \tag{A.38}$$

$$C_{011} = -\frac{\sigma_{rf}\rho}{3\hat{\omega}_d}\left(\overline{k_{yz}} + A_{011}\right) \tag{A.39}$$

$$C_{101} = -\frac{\sigma_{rf}\rho}{3\hat{\omega}_d}\left(\overline{k_{xz}} + A_{101}\right) \tag{A.40}$$

$$C_{110} = -\frac{\sigma_{rf}\rho}{3\hat{\omega}_d}\left(\overline{k_{xy}} + A_{110}\right) \tag{A.41}$$

$$C_{200} = \frac{\delta\rho}{9} - \frac{2\sigma_{rf}\rho}{9\hat{\omega}_d}\left(\overline{k_{xx-yy}} + B + \overline{k_{xx-zz}} + C\right) \tag{A.42}$$

$$C_{020} = \frac{\delta\rho}{9} - \frac{2\sigma_{rf}\rho}{9\hat{\omega}_d}\left(-2\left(\overline{k_{xx-yy}} + B\right) + \overline{k_{xx-zz}} + C\right) \tag{A.43}$$

$$C_{002} = \frac{\delta\rho}{9} - \frac{2\sigma_{rf}\rho}{9\hat{\omega}_d}\left(\overline{k_{xx-yy}} + B - 2\left(\overline{k_{xx-zz}} + C\right)\right). \tag{A.44}$$

Now the cumulants can be transformed back to distributions, with the central moments up to order two and cumulants of order higher than two set to zero.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag der Professur für Strömungsmechanik eingereichte Diplomarbeit zum Thema

Kopplung eines künstlichen neuronalen Netzwerks mit LES-LBM zur Verbesserung einer Windpark-Steuerung

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe.

Dresden, 06. Januar 2020

Henry Korb