Wake Escape: Reinforcement Learning for Dynamic Floating Wind Turbine Repositioning

Deepika Manjula Mary Phadke | Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Wake Escape: Reinforcement Learning for Dynamic Floating Wind Turbine Repositioning

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Deepika Manjula Mary Phadke

August 9, 2026

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical Engineering (ME) for acceptance a thesis entitled

Wake Escape: Reinforcement Learning for Dynamic Floating Wind Turbine Repositioning

by

Deepika Manjula Mary Phadke

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: August 9, 2026

Supervisors:

Prof.dr.ir. Jan-Willem van Wingerden

Ir. Alexandra Ministeru

Ir. Leendert Starink

Reader:

Dr. Azita Dabiri

Abstract

As we strive to meet more of our global energy demand with renewables, floating offshore wind has gained popularity: floating turbines can access the stronger winds far out at sea and so generate more power per turbine. A key limiter of wind farm power production is wake interaction, causing estimated yield losses between five and twenty percent.

Semi-submersible floating turbines have a unique ability to reposition. They can use the yaw-induced turbine repositioning (YITuR) technique to reposition, letting downstream turbines escape their upstream neighbour's wake. Recently, the optimality of sinusoidal *dynamic* repositioning trajectories has been shown analytically, but not in a learning-based environment. This thesis is the first to prove the concept of dynamic floating wind turbine repositioning for wake escape via *reinforcement learning* (RL) in simulation.

Dynamic turbine repositioning via a Proximal Policy Optimisation (PPO) neural network control policy is a promising alternative to traditional look-up table wind farm control. Here, the problem is formulated as a partially observable Markov decision process (POMDP), using an array of two turbines. A model-free RL controller is trained using PPO, inside the Free Vortex Wake Model (FVWM), with an extension to enable repositioning motions. The optimal RL policies discovered are periodic, with frequencies between Strouhal numbers of 0.048 and 0.065 — near the platform natural-frequency Strouhal number of 0.058 — achieving array powers of 38–43% beyond the fully-waked baseline and 9–13% beyond the static yaw-only optimum.

A proof-of-concept laboratory setup for testing dynamic repositioning control was designed, built, and commissioned as part of this thesis. It consists of two scaled versions of the NREL 5 MW reference turbine, with porous-disc rotors. Porous discs retain the necessary wake characteristics under yawing, but they exacerbate the thrust drop beyond what real three-bladed rotors would do. With adaptations per the roadmap provided, laboratory testing can be a strong candidate for speeding up training for sample-hungry model-free RL controllers for dynamic repositioning.

Table of Contents

Acknowledgements **xi**

Introduction **1**

0-1 Motivation and Background Information . . . 2
0-1-1 Introduction to Wind Power and Wakes . . . 2
0-1-2 Introduction to Floating Wind Turbines . . . 2
0-1-3 Introduction to Wake Steering . . . 3
0-1-4 Control Strategies for Dancing Turbines . . . 4
0-1-5 Flow Models . . . 6
0-1-6 Introduction to Standard Wind Turbine Control via Look-up Table . . . 6
0-1-7 Introduction to Reinforcement Learning . . . 7
0-1-8 Bellman Expectation Equation . . . 8
0-1-9 Principle of Optimality . . . 9
0-2 Project Proposal . . . 12
0-2-1 Problem Statement . . . 12
0-2-2 Research Gap . . . 12
0-2-3 Research Objective and Questions . . . 13
0-2-4 Contribution and Outline . . . 14
0-3 Thesis Outline . . . 14

1 The Simulation Environment: Incorporating Turbine Repositioning into the Free Vortex Wake Model **17**

1-1 The Array of Two Turbines . . . 17
1-2 NREL 5 MW Reference Turbine . . . 19
1-3 What is the Free Vortex Wake Model . . . 19
1-3-1 On the Selection of the FVWM . . . 21
1-3-2 Solver Configuration Considerations . . . 21
1-4 Incorporating Position Changes into FVWM . . . 21
1-4-1 Low Pass Filter for Floater Dynamics . . . 22
1-4-2 Dimensional Analysis - Frequency Scaling with Strouhal Number . . . 22
1-4-3 Choice of Yaw-to-repositioning Gain . . . 23
1-4-4 Implementation Coding Mechanics . . . 24
1-5 FVWM Turbine Repositioning Extension Conclusion . . . 25

2 Reinforcement Learning Controller Formulation 27
2-1 Model-free RL Justification . . . 27
2-2 Objective Function — $J(\pi)$ . . . 28
2-3 POMDP Formulation . . . 28
2-3-1 State Space — $\mathcal{S}$ . . . 28
2-3-2 Action Space — $\mathcal{A}$ . . . 28
2-3-3 Observation Space — $\mathcal{O}$ . . . 29
2-3-4 Transition Model — $\mathcal{P}$ . . . 30
2-3-5 Reward Function — $\mathcal{R}$ . . . 30
2-3-6 Observation Function — $\mathcal{Z}$ . . . 31
2-3-7 Discount Factor — γ . . . 31
2-3-8 Initial State Distribution — μ_0 . . . 32
2-3-9 Horizon — H . . . 32
2-4 Block Diagram of the POMDP . . . 32
2-5 Policy Optimiser . . . 33
2-5-1 Policy Gradient Methods . . . 34
2-5-2 Proximal Policy Optimisation . . . 34
2-5-3 Neural Network Details . . . 35
2-6 Coding Implementation Description . . . 36
2-7 Reinforcement Learning Controller Formulation Conclusion . . . 36

3 Case Study 1: Simulation with the Free Vortex Wake Model 37
3-1 Simulator and RL Controller Training Set-up . . . 38
3-1-1 History Length . . . 38
3-1-2 Discount Factor — γ . . . 38
3-1-3 Horizon — H . . . 39
3-1-4 Normalised Performance Parameter: ($m = P$ in Case Study 1). . . . 39
3-1-5 FVWM Configuration Parameters . . . 39
3-1-6 RL Training Regime Hyper-parameters . . . 40
3-2 Orientation in the Solution Space . . . 41
3-2-1 Fully Waked Comparison Case . . . 41
3-2-2 Static Optimum . . . 41
3-2-3 Grid Search . . . 43
3-3 The Optimal RL Results . . . 44
3-3-1 Evolution of the Training Regime . . . 45
3-3-2 Headline RL Results . . . 47
3-3-3 Optimal Amplitude . . . 47
3-3-4 Optimal Frequency . . . 48
3-3-5 Optimal Phase Offset . . . 49
3-4 Training, Computation and Rollout Costs . . . 50
3-4-1 RL Training in Low-resolution FVWM . . . 50
3-4-2 High-resolution FVWM Training Time Extrapolation . . . 51
3-4-3 Estimation of Training Time in Experimental and Real-World Scenarios . . . 52
3-5 Case Study 1 Conclusion . . . 54

4 Laboratory Setup **57**
4-1 General Experimental Design Details 58
4-2 Wind Shaper 58
4-3 Reference $5\,\mathrm{MW}$ NREL Turbine 58
4-4 Dimensional Analysis — Frequency Scaling with Strouhal Number 59
4-4-1 Laboratory Turbine Repositioning Distance 59
4-4-2 Actuator Kinematics 59
4-4-3 Flow Constraints on the Geometric Scaling 60
4-4-4 Parameter Sweep To Choose Optimal Design 60
4-4-5 Resulting Scaling Factor 61
4-5 Scaling Factor 61
4-6 Rotor Design 62
4-6-1 Porous Disc Literature Review 62
4-6-2 Porous Disc Design 62
4-7 Load and Torque Requirements 63
4-7-1 Actuator Torque 63
4-7-2 Structural Integrity of the Rotor and Tower 64
4-8 Summary of Design Specifications 64
4-9 Experimental Plan 65
4-10 Experimental Setup Conclusion 66

5 Case Study 2: Laboratory Experiments **67**
5-1 Test Plan 67
5-2 Strain Gauge Calibration 68
5-3 Commanded Versus Achieved Signals 69
5-4 Strain Results: Is There An Intelligible Increase in Array Power? 71
5-5 Case Study 2 Conclusion 75

6 Conclusion **77**
6-1 Research Questions 77
6-1-1 RQ 1: POMDP Formulation 78
6-1-2 RQ 2: RL Controller Feasibility 78
6-1-3 RQ 3: Learned Control Policies 79
6-1-4 RQ 4: Power Gains from Dynamic Repositioning 79
6-1-5 RQ 5: Effect of History Length as a Surrogate for Unobservable States . . . 80
6-1-6 RQ 6: Laboratory Setup Specifications 80
6-2 Contributions Delivered 81
6-2-1 Contribution 1: POMDP and RL Formulation 81
6-2-2 Contribution 2: Optimal Signal Characterisation 81
6-2-3 Contribution 3: Computational Complexity and Training Time 81
6-2-4 Contribution 4: FVWM Extension 82
6-2-5 Contribution 5: Laboratory Setup Specification 82
6-2-6 Contribution 6: Laboratory Commissioning and Empirical Characterisation . . 82
6-3 Closing Statement 82

A Appendix A: Surveys of Real Life Wind Farm Wake Steering Test Campaigns, and Existing RL Control of Wind Farms **83**

A-1 Survey of the Most Recent Real Life Wind Farm Wake Steering Test Campaigns . . 83

A-2 Survey of Existing RL Control for Wind Farms 85

Bibliography **91**

AI Declaration **101**

Glossary **103**

List of Acronyms 103

List of Figures

1 Visible wakes at Horns Rev offshore wind farm, Denmark. Photo by Christian Steiness, Vattenfall, reproduced from [1]. 1

2 The four types of floating offshore wind turbine platforms. The semi-submersible on the right is the type used in this thesis. Adapted from [2] with permission. 2

3 The semi-submersible floating turbine's nacelle and blades enable roll, pitch and yaw motions, and the floating base undergoes heave, surge and sway motions. 3

4 "Turbine repositioning mechanism. Nacelle yaw angle γ and platform yaw angle θ_z are defined with respect to the platform-fixed frame and the inertia frame, respectively," Niu et al. [3]. (The image and caption in quotation marks are both from [3].) The paler shaded turbine and mooring lines show the original turbine position, and the dark lines and turbine show its new position due to YITuR. 4

5 Timeline of major FOWT repositioning research advances. 5

1-1 Depicting the range of motion and layout of the two-turbine array. 18

1-2 Example turbine motions with only the upstream turbine's wake shown. 18

1-3 Free vortex wake model particle release and advection representations. 20

1-4 Bode plot of filter design. 24

2-1 Block diagram of POMDP controller. 33

2-2 Actor-critic MLP architecture. The flattened observation vector is fed to two separate networks: the actor (policy), which outputs the mean of the Gaussian distribution over the yaw action, and the critic, which outputs a scalar state-value estimate. Each network has two hidden layers of 64 tanh units; no layers are shared. 36

3-1 Logical structure of this chapter. The dashed arrow indicates the iterative process by which configuration changes prompted re-running of the baseline and grid search cases. 38

3-2 Fully waked case: power output over time. Turbine 1 produces optimal power, but Turbine 2 produces almost no power. The wind velocity deficit caused by the wake is shown at the bottom of the image for context, where the black lines at x/D = 0 and 5 represent the turbines. 42

3-3 Parameter sweep of static yaw angles to find the optimum (the optimum is the same as that found with the slightly lower resolution FVWM configuration by [4]). 42

3-4 Static optimum: power output over non-dimensional time. Turbine 1 sacrifices its optimal power, so that Turbine 2 can produce more power than in its fully-waked case. This results in higher total array power. The wind velocity deficit caused by the wake is shown below for context, where the black lines at x/D=0 and 5 representing the turbines. There is significantly higher wind speed (white region) in front of Turbine 2 than there was for the fully-waked case shown in Figure 3-2. 43

3-5 Power contour for the $32.5\,\mathrm{deg}$ amplitude case, run in the high-resolution FVWM configuration detailed in Table 3-1. 44

3-6 The variety in the top 22 optimal frequencies, amplitudes and phase offsets from the grid search. 45

3-7 Reward increasing until a plateau is reached (optimal solutions exist in this plateau). Note that the magnitude of the reward is not directly comparable to the power gain calculations elsewhere in this chapter because for this figure, these values are directly from the low-resolution training runs. 46

3-8 Yaw trajectories at the four training checkpoints marked in Figure 3-7. 46

3-9 Violin box-plot of history length sensitivity analysis: the shaded region is widest where the most results are concentrated. This is a visual aid to the standard box plot. For context, the zero-yaw baseline array power is 0.247 and the static T1 yaw-only optimum from [4] is 0.313. 47

3-10 Highlighting the fact that the optimal RL policies lie in a similar frequency, amplitude and phase offset range to the grid search optima, but, RL optimal policies prefer slightly differing trajectories for T1 versus T2. The grid search constrains T1 and T2 to follow the same sinusoid. 48

3-11 Optimal yaw angles across the evaluated history lengths. 48

3-12 Optimal turbine repositioning amplitudes across the evaluated history lengths. 49

3-13 Optimal forcing frequencies across history lengths and their relationship with power output. 49

3-14 Relationship between yaw angle phase offset and power output. 50

3-15 Optimal phase offset box plots obtained for yaw angle and turbine repositioning across the evaluated history lengths. 50

3-16 C_{train} across the models. 51

4-1 The laboratory setup. 57

4-2 From L-R: a rotating 3-bladed MoWiTo turbine [5], a uniform porous disc [6] and a radially varying porous disc [7]. 62

4-3 Rotor design drawing. 63

5-1 Strain gauge calibration data snippet. Wind =5m/s, pull=movement away from wind source, push=movement towards wind source. 68

5-2 Commanded versus achieved signal tracking for the forced-motion actuators. 69

5-3 Amplitude tracking analysis: for (a-c), the decline in performance is for high-frequency high-amplitude test cases 21-24. 70

5-4 Turbine aft-strain normalised by wind capture baseline. 72

5-5 Net array thrust losses are due to T1's exacerbated losses under yawing. This figure shows all cases except Test 19 do not 'break even' in terms of the losses at T1 being sufficiently compensated by gains at T2. 74

5-6 Fore-aft T1 strain signal energy across turbines. 74

5-7 Side-side T1 strain signal energy across turbines. 75

List of Tables

1 Summary of contributions corresponding to Figure 5 timeline references. 5

2 Comparison of wind farm wake modelling tools. 6

1-1 Key specifications of the NREL $5\,\mathrm{MW}$ reference wind turbine [8]. 19

1-2 Free-vortex wake model configuration parameters, following van den Broek et al. (2022) [4, 9]. All parameters are non-dimensionalised by rotor diameter and inflow speed. 22

1-3 Characteristic full-scale and non-dimensional equivalent frequencies of the floater case reported by Starink et al. [10] (mooring line radius $R = 775\,\mathrm{m}$, rotor diameter $D = 125\,\mathrm{m}$, and inflow velocity $U = 8.2\,\mathrm{m\,s^{-1}}$). Computed using FVWM simulation clock $\Delta t = 0.3$, non-dimensional, where the equivalent full-scale sample time is $\Delta t \cdot D/U \approx 4.57\,\mathrm{s}$. 23

3-1 Free-Vortex Wake Model (FVWM) configurations used for the low- and high-resolution simulations, alongside the reference static-optimum configuration of van den Broek et al. [4]. All configurations share identical temporal and spatial domain settings, differing only in wake discretisation resolution. 40

3-2 PPO hyperparameters used for all history-length configurations. 41

3-3 FVWM (3D) state-vector size, low- vs. high-resolution configuration. 52

3-4 Similarity scale factors, full scale to laboratory. 53

3-5 Strouhal-matched frequency targets, St $\in [0.048, 0.065]$. 53

3-6 Cheapest training regime by control resolution, 10^6-step budget. 54

4-1 Subset of results of the diameter sweep. Rows shaded red exceed the edge distance spacing constraint. 61

4-2 Design specifications summary. 65

5-1 Full forced-motion test matrix (24 points, each repeated 10 times). Cell entries are test_id, one per ψ level. 68

5-2 Fore-aft strain ratio (where the wind-on baseline case = 1.0) per test point means and confidence intervals vs. baseline. Each test point was repeated 10 times and the confidence interval (CI) is quoted as either the 99 or 99.9% CI. Legend: ***99.9% CI, **99% CI, ns = not significant. Note: Test 1 is omitted as its raw data files were corrupted between graph analysis and creation of this table. 73

A-1 Comparative summary of wake steering field tests in commercial wind farms. LUT = Look-Up Table; FLORIS = FLOw Redirection and Induction in Steady State; SCADA = Supervisory Control and Data Acquisition; D = rotor diameter. 84

A-2 Reinforcement Learning approaches for wind turbine and farm control (including some especially useful general machine learning approaches that do not use RL). 86

Acknowledgements

I would like to thank my supervisors Jan-Willem van Wingerden, Alexandra Ministeru and Leendert Starink for their advice and support. It was lovely to speak with you and learn from you in our regular meetings. Thank you for the opportunity to work on control topics for wind energy. To Ralph Willekes and Will van Geest, thank you for sharing your expertise and helping me build the laboratory setup. My thanks also goes to the following people within the Delft Center for Systems and Control: to Sachin Umans for his invaluable advice on porous discs and prototyping of alternate methods when the actuators were not working in the lab, to Azita Dabiri for her initial reinforcement learning advice that inspired my further research, to my friend Anouk de Koning for discussing our codes often to brainstorm and error check, to Shane van Rhijn for very kindly swapping our lab bookings multiple times when things went wrong, to Yichen Liu for his advice on reinforcement learning and for providing feedback on my report, to Ivo van Straalen for helping me when the programmable logic controller crashed, and finally to Jonas Gutknecht, Marcus Becker, Tristan Crumpton and Daan van der Hoek for their advice on dynamic yawing and various wind turbine wake models.

Special thanks go to my mother and Vijaykaka, although neither of them likes this word 'thank you'. To Mum for emotional and financial support, and for instilling in me a life-long love of learning. To Vijaykaka for inspiring me and teaching me the mathematical skills required for this master's, from primary school, into high school and on to control theory, when my own Baba was no longer able to give me these lessons.

To my friends, family, loved ones, and colleagues in Delft, Pune, Thane, Nasik, Cherbourg, Dorking, Melbourne, and at home in Adelaide, you have all helped me to or through this master's programme with conversation, parties, commiseration, sports sessions, and a whole lot of laughter — thank you.

Delft, University of Technology
August 9, 2026

Deepika Manjula Mary Phadke

“Without that passion and urge, there is a gradual oozing out of hope and vitality, a settling down on lower levels of existence, a slow merging into non-existence. We become prisoners of the past and some part of its immobility sticks to us.”

— *Jawaharlal Nehru, The Discovery of India*

Introduction

We must be more sustainable for the future of our world. As stated in Art. 2(1)(a) of the Paris Agreement [11], we are aiming to "limit the temperature increase to 1.5°C above pre-industrial levels". To that end, renewable energy sources are becoming increasingly popular. Wind energy is one such source. As we strive to meet ever more of our global energy demands with renewable sources, floating offshore wind has become popular: floating turbines can access the stronger winds far out at sea and therefore generate more power per turbine.

A key limiter of wind farm power production is wake interaction. In fact, an estimated five to twenty percent of yield is lost to wake effects [12]. Due to space constraints, the turbines in many farms are so close together that upstream turbine wakes extend to downstream turbines. This is a problem because the wind speed inside the wake is lower than the free-stream velocity, which reduces downstream turbine power production. This problem is clearly visible in the (in)famous image of turbine wakes at Horns Rev offshore wind farm in Figure 1. A solution to this problem is to redirect, or 'steer', the wakes of upstream turbines away from downstream turbines. Of the wake mitigation techniques, power de-rating and yaw/induction control have been prominent for the last ten years, but recently, with the advent of floating wind turbines, turbine repositioning has come into focus [13, 10] (relocation is not possible with fixed turbines). In other words, there is a new way to mitigate wake effects for floating turbines.

Figure 1: Visible wakes at Horns Rev offshore wind farm, Denmark. Photo by Christian Steiness, Vattenfall, reproduced from [1].

Implementing control solutions to mitigate the wake effect is challenging due to the complex underlying dynamics. Data driven controllers can alleviate the need to compute accurate solutions to these complex dynamics in real-time. Reinforcement learning (RL) is coming into focus as a method for improving models of, and controllers for, real-world systems. Until now RL's grand successes have been in applications to game-theory and robotics, but a significant body of research is developing around its application to real world system modelling [14, 15]. RL can be used to both develop new controllers, and, better identify the system environment. Through their trial and error interactions directly with the unsimplified environment, RL agents might uncover complex, cooperative wake-steering strategies between turbines that would otherwise be left undiscovered.

0-1 Motivation and Background Information

The focus of this thesis is floating turbine repositioning via reinforcement learning control, but before diving into this specific topic in detail, let us first discuss the pre-requisite background knowledge: wind power and wakes, the floating wind turbine system, the basics of wake steering, control strategies to effect repositioning (which is a special case of wake steering), existing wake simulation tools, general methods for wind farm control, and the fundamentals of reinforcement learning.

0-1-1 Introduction to Wind Power and Wakes

The most commercially successful wind turbines are the horizontal axis wind (HAWT) turbine, pictured in Figure 2. Globally, HAWT wind farms produce around 2300 TWh of electricity, amounting to about 8–10% of the global energy market (this is based on 2025 figures from [16] and [17]). This figure could be even higher if it were not for wake interactions causing yield losses between 5–20% [18]. These wake losses occur because turbines are placed close enough together, due to space constraints, that the low velocity region of air aft of the turbine extends to the air-intake region of the next turbine downstream. This reduces the power the downstream turbine can produce. The industry is focussed on mitigating the effect of these wakes. Further information on wake mitigation is included in Section 0-1-3.

0-1-2 Introduction to Floating Wind Turbines

The purpose of building offshore wind farms is to access the stronger winds out at sea. The floating variety of offshore turbines can be installed in even deeper waters where fixed-bottom turbines are economically infeasible, thus broadening the area where wind energy can be harnessed. The main types of floating offshore wind turbines (FOWT) are shown in Figure 2. This thesis focusses on one particular type of floating turbine: the semi-submersible, which is essentially a regular turbine mounted on a floater which is in turn anchored to the seabed via mooring lines. A semi-submersible floating turbine's motion is typically characterised in six degrees of freedom: roll, pitch, yaw, heave, surge and sway. This is depicted in Figure 3. More details on how this type of design can enable whole-of-platform repositioning is discussed in subsection 0-1-4, "Control Strategies for Dancing Turbines".

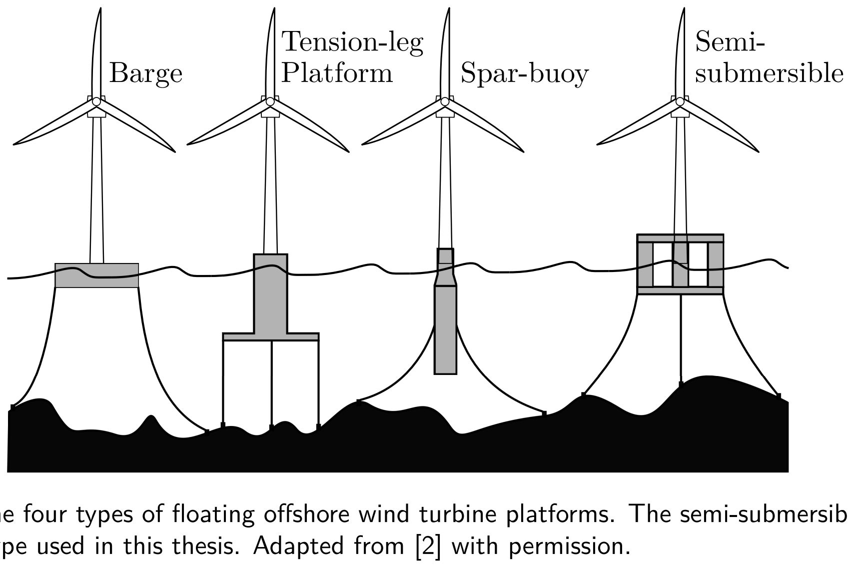


Figure 2: The four types of floating offshore wind turbine platforms. The semi-submersible on the right is the type used in this thesis. Adapted from [2] with permission.

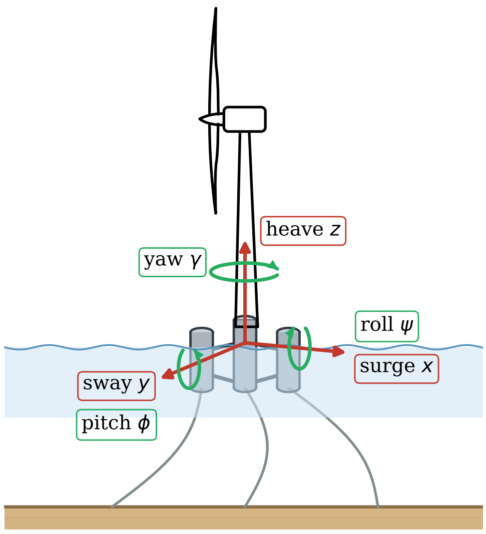


Figure 3: The semi-submersible floating turbine's nacelle and blades enable roll, pitch and yaw motions, and the floating base undergoes heave, surge and sway motions.

0-1-3 Introduction to Wake Steering

Yaw and induction control as it is understood today is largely built on the early works of 2015 and 2016 [19, 20, 21, 22], which in turn had advanced the early pitch and torque for induction control techniques initially proposed in [23] in 1988. Essentially, the turbine is yawed to misalign it with the direction of the oncoming wind. This steers the wake away from downstream turbines. This research area is well developed, and after many successful high-fidelity computer simulation experiment campaigns, field testing began in 2019. These real-world testing campaigns also proved fruitful, and yaw-based wake steering is currently being implemented on some operational farms. Current state-of-the-art advances in wake steering with yaw and induction control lie in using data driven techniques and machine learning to produce optimal controllers, as well as dynamic induction control to produce a helical wake [5, 24, 25], and floating turbine repositioning [10, 26].

The campaigns summarised in Table A-1 in Appendix A-1 are some of the most widely cited or newest real-world validations of wake steering control. All use static, model-dependent controllers, not learned policies. They establish the practical performance ceiling against which all new controller results are in interpreted by the industry. In all cases, the unobservable states in the atmosphere remain uncorrected and unestimated, and closed-loop feedback not implemented. They all identify model validation and inability to adapt to real-time wind condition changes as the largest adoption barriers: this is one prompt for research into RL for improved modelling and real-time control.

Since the aforementioned studies all highlight the use of open-loop control and a steady state wake model as their major limitations, as a comparison point, the closed-loop high fidelity simulation by [27] will now be discussed. In this study using highly varying wind fields, similar gains were found: energy yield gains of 1.4% compared to a baseline greedy controller, and instantaneous energy gains up to 11% for wake-loss-heavy situations, but the applicability of the work to the real world stills fall short in terms of the fact that they use a steady state wake model (FLORIS [28]). Still, it is a valuable paper as it addresses the problem wherein the mismatch between steady-state predictions and time-varying real-world results due to inaccuracies in the model, get amplified with open-loop control. Such closed-loop techniques are yet to be tested in real-world farms.

0-1-4 Control Strategies for Dancing Turbines

Turbine repositioning is a special case of wake-steering and is still a new field of research. Figure 5 and its accompanying table, Table 1, depict the history of this field, the current focii and headline developments: the field is currently focused on researching distributed control, platform oscillation reduction, use of time-varying wakes in simulations, and deployable model predictive control (MPC) for repositioning.

In 2014, Rodrigues et al. [29] were the first to propose a framework for optimal farm layout based on *moving* turbines. They showed improved energy production was possible, but did not venture into the mechanisms to actually achieve turbine repositioning. This research was quickly followed up by Fleming et al. in 2015 [19] using high-fidelity simulations to prove that this does indeed increase power production under fully-waked conditions with two turbines. From there, in 2016 Han and Nagamune [30] were the first to demonstrate that aerodynamic thrust can be used as the control input to effect repositioning. The use of aerodynamic thrust for repositioning is a form of what is termed 'passive' relocation. 'Active' relocation refers to relocation with thrusters or physical changes to the mooring lines. Passive relocation inherently means that an optimisation problem is solved in order to allow us to develop the control law. The change in aerodynamic thrust for repositioning is achieved using Yaw and Induction-based Turbine Repositioning (YITuR): the term most widely used in industry. This YITuR works by misaligning the turbine nacelle with the direction of oncoming wind, thereby causing the aerodynamic thrust force to change and push the turbine to a new location. This method also relies on the floating turbine mooring lines not being too taut. An image to illustrate this YITuR is provided in Figure 4.

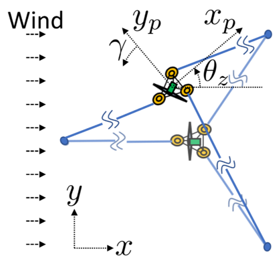


Figure 4: "Turbine repositioning mechanism. Nacelle yaw angle γ and platform yaw angle θ_z are defined with respect to the platform-fixed frame and the inertia frame, respectively," Niu et al. [3]. (The image and caption in quotation marks are both from [3].) The paler shaded turbine and mooring lines show the original turbine position, and the dark lines and turbine show its new position due to YITuR.

By 2019, in their review of current promising methods for mitigating wake effects, Kheirabadi et al. [15] cited turbine repositioning as one of three promising methods. In fact all of the recently developed repositioning control algorithms (both in the review by [15] and afterwards) are based on YITuR. In [13], the current state-of-the-art baseline repositioning controller for implementation on real turbines and farms is proposed. As they highlight in their article, the current research efforts are focussed on developing control-oriented dynamic models, mitigating platform oscillations, farm-level and distributed control architectures, control co-design, fatigue analysis of wake effects on the turbine, improving high fidelity simulations, and expanding experimental validation of the concept. The results of [15] reinforce the point that the adoption of turbine repositioning in real farms is hindered by the lack of simulation tools that allow movement of the turbine and accurately map the dynamics. The aforementioned current research focii aim to address these deficiencies.

The only papers yet to truly prove the viability and usefulness of constant movement optimality for FOWT under mooring constraints are by Starink et al. [10, 43]. To better illustrate the

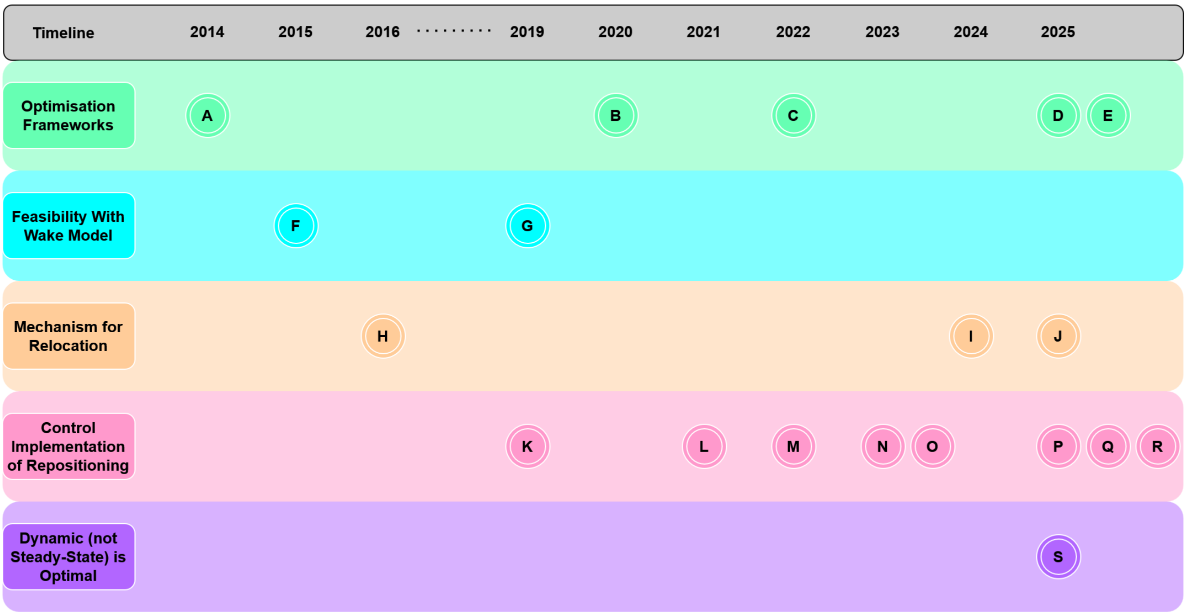


Figure 5: Timeline of major FOWT repositioning research advances.

Reference	Headline Contribution
A, [29] Rodrigues et al. 2015	Genesis of this idea and initial feasibility proof
B, [31] Kheirabadi et al. 2020	Shows YITuR Steady-State (SS) feasibility is dependent on farm layout
C, [32] Froese et al. 2022	Proposes YITuR SS via iterative gradient-based framework
D, [33] Huang et al. 2025	SS Layout and operation optimisation framework
E, [34] Wei et al. 2025	Time-varying wind framework
F, [19] Fleming et al. 2015	CL wake control framework
G, [35] Kheirabadi et al. 2019	YITuR controller with SS wake & 2D wind field
H, [30] Han et al. 2016	Pioneered aerodynamic thrust for repositioning
I, [36] Alkarem et al. 2024	Mooring orientation adaptation for active relocation
J, [37] Phadnis et al. 2025	Individual Pitch Control (IPC) for Turbine Repositioning
K, [38] Han et al. 2020	Real time LQI controller to attain steady-state target
L, [39] Kheirabadi et al. 2021	Distributed MPC for repositioning | neural-net for improved floater dynamics
M, [40] Gao et al. 2022	Baseline parametric controller for direct implementation in real systems
N, [41] Jard et al. 2023	Robust MPC for repositioning
O, [3] Niu et al. 2023	LQR IPC reducing platform oscillations during repositioning $\Rightarrow$ increased power
P, [13] Niu et al. 2024	Turbine & Farm Repositioning Controller for $\downarrow$ oscillations and $\uparrow$ power
Q, [26] Mei et al. 2025	Physics model for model-based DRL: yaw control including repositioning
R, [42] Jard et al. 2025	MPC with dynamic wind field, but static optimal target position
S, [10] Starink et al. 2025	Proof that time-varying repositioning control signals outperform static

Table 1: Summary of contributions corresponding to Figure 5 timeline references.

difference between dynamic turbine repositioning and standard steady-state target attainment, this research group term their approach ‘Dancing Turbines’. Importantly, their research shows that periodic results are optimal, but they do constrain their investigations to perfectly sinusoidal trajectories. It may therefore be beneficial to widen the search for optimal trajectories beyond sinusoidal constraints. They also answer the call for new turbine control methods to be tested in 3-D time-varying dynamic wake simulations (they use the quasi-dynamic medium fidelity FLORIDyn [44, 45] software in [10] and the high-fidelity QBlade simulation environment [46] employed in [43]), but have not yet added experimental validation to their suite of test environments. That is to say, one can be confident that research of dynamic turbine repositioning strategies is fruitful, and there is most definitely scope to widen the search to non-sinusoidal trajectories tested in experimental settings.

0-1-5 Flow Models

As alluded to in the previous sections, choosing an appropriate wake simulation software for testing wake steering control algorithms is important. This section acts as a brief summary targeted at identifying wake models that capture wake curl, are control-oriented, and can be run on a standard laptop computer. Detailed explanations of the underlying physics are excluded. Wake curl is a dominant mechanism in wake steering [47]. Therefore, it is important that models used to validate wake steering control efficacy accurately capture wake curl. Without proper modelling of the 3D and dynamic nature of wakes, significant underestimates of the benefits of wake steering occur [4]. The second and third focii of this review, control-oriented models with low computational intensity, have been chosen to highlight those models for which online closed-loop control algorithms can be tested. This is where wake steering research advancements are currently directed. Table 2 compares popular wake models with respect to these criteria. FLORIS [28], FLORIDYN [44], and WFSim [48] are widely cited control-oriented software tools. Unfortunately, none of them capture the curled-wake characteristic. They are engineering models, rather than detailed physics models. Conversely, large-eddy simulations (e.g. SOWFA [49], PALM [50], EllipSys3D [51, 52], Turbo-Wake [53]) are physics-based models that accurately capture 3D wake characteristics, but they are computationally intensive. This makes them unusable for real-time control. They are instead tailored to situations where fluid mechanics accuracy is paramount, and can lend themselves to use in pre-training control algorithms offline. These pre-trained algorithms can then later be tested online on one of the control-oriented tools, or ideally in a real farm. In fact, of the surveyed wake models, only the Free Vortex Wake Model developed by [4] reproduces the 3D curled wake, is control-oriented, and has a low computational burden.

Table 2: Comparison of wind farm wake modelling tools.

Tool	Dynamic	Wake curl	Control-oriented	3D	Laptop-feasible
FLORIS [28]	–	–	✓	–	✓
FLORIDyn [44]	✓	–	✓	✓	✓
Jensen/Park [54]	–	–	✓	–	✓
PyWake [55]	–	–	✓	✓	✓
WFSim [48]	✓	–	✓	–	✓
FAST.Farm	✓	✓	–	✓	–
3D FVW [4]	✓	✓	✓	✓	✓
2D FVW [4]	✓	–	✓	–	✓
QBlade [46]	✓	✓	✓	✓	–
OFF [56]	✓	–	✓	✓	✓
WindSE [57]	–	–	–	–	✓
SOWFA [49]	✓	✓	–	✓	–
PALM [50]	✓	✓	–	✓	–
EllipSys3D [51, 52]	✓	✓	–	✓	–
Turbo-Wake [53]	✓	✓	–	✓	–

0-1-6 Introduction to Standard Wind Turbine Control via Look-up Table

The final topic in this abridged introduction to floating wind farms, before moving on to a brief introduction to reinforcement learning, is the ‘look-up table’ (LUT). LUTs are the industry standard for wind farm control. As the name suggests, this is a table of pre-computed points that map measured variables such as wind speed and direction, to control actions such as yaw offsets, and to reference output powers. This is just a subset of the values typically included in a look-up table, but the key point is that this is a static reference document that is limited by the accuracy and amount of data it was trained on. It is used because it allows for fast, simple and computationally efficient open loop controllers to be deployed on the real turbines. Any deviation from this standard approach to wind farm control will be judged by manufacturers based on the level of increased power it affords (or decreased fatigue loading, or other objective) and its real-time deployment capability.

0-1-7 Introduction to Reinforcement Learning

Instead of the LUT approach to wind farm control, deep reinforcement learning can be used to develop a neural-network of relationships between measurable quantities, control actions and objectives. This is quickly becoming the next most popular controller for wind farms, but it is still in the early stages of development. Table A-2 in Appexdix A-2 is an exhaustive list of all new (up to early 2026) published works on RL control development for wind farm control. In order to support the reinforcement learning content of this thesis, this section is included as a short introduction to the relevant fundamental components of reinforcement learning: finite Markov decision processes, partially observable Markov decision processes, the Bellman expectation equation, and the principle of optimality.

Reinforcement learning is one of the three main streams of machine learning: reinforcement learning, supervised learning and unsupervised learning. Supervised learning involves training on labelled data, unsupervised learning finds hidden patterns in unlabelled data, and reinforcement learning trains agents to make decisions through trial and error to maximize a reward [58].

RL can then be split into two broad categories distinguished by how their value functions are obtained. A *model-based* agent uses the environment's transition dynamics $P(s' \mid s, a)$ and reward function directly to compute values and extract a policy, whereas a *model-free* agent has no such model and instead learns values by sampling transitions through interaction, never explicitly representing $P(s' \mid s, a)$.

Finite Markov Decision Processes

In RL, system environments are typically encoded as Markov Decision Processes (MDP). This is how the environment can be formally described. An MDP is defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu_0, H)$, where:

- $\mathcal{S}$ is the state space, the set of all possible environment states s,
- $\mathcal{A}$ is the action space, the set of all actions a available to the agent,
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, where $P(s' \mid s, a)$ is the transition probability of reaching state s' from state s after taking action a,
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, where $R(s, a, s')$ is the reward received after transitioning from s to s' via action a,
- $\gamma \in [0, 1)$ is the discount factor, controlling the present value of future rewards,
- $\mu_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution, where $\mu_0(s)$ is the probability of starting in state s,
- $H \in \mathbb{N} \cup \{\infty\}$ is the horizon, the maximum number of time steps per episode.

The agent's goal is to find a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$, where $\pi(a \mid s)$ denotes the probability of selecting action a in state s, that maximises the expected discounted return:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t, s_{t+1}) \right]. \tag{1}$$

Partially Observable Markov Decision Processes

In many real systems, there are both process and observation uncertainties: the system model and the sensor suite are imperfect. For this reason there is an extension to the MDP called the partial observability Markov decision process (POMDP). Here, the agent maintains a *belief state* — a probability distribution over possible true states, updated recursively via Bayes' rule as new observations arrive [59]. Mathematically, the POMDP extension to the MDP tuple is: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P, R, Z, \gamma, \mu_0, H)$, where $\mathcal{S}$, $\mathcal{A}$, P, R, γ, μ_0, and H are defined as in the previous paragraph, and the new terms are:

- $\mathcal{O}$, the observation space, the set of all possible observations o available to the agent,
- $Z : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \to [0,1]$, where $Z(o \mid s', a)$ is the probability of receiving observation o after transitioning to state s' via action a.

Let $\mathcal{B} = \Delta(\mathcal{S})$ denote the belief space, the set of all probability distributions over the state space $\mathcal{S}$. The belief state $b_t \in \mathcal{B}$, where $b_t(s)$ is the probability of being in state s at time t given the history of observations and actions, is updated recursively via Bayes' rule as follows:

$$b_{t+1}(s') = \frac{Z(o_{t+1} \mid s', a_t) \sum_{s \in \mathcal{S}} P(s' \mid s, a_t)\, b_t(s)}{P(o_{t+1} \mid b_t, a_t)}. \tag{2}$$

Then, otherwise, the agent's goal is still the same: to find a policy that maximises the expected discounted return as per the aforementioed Equation 1. The only difference is that the policy is now defined over the belief state $\pi : \mathcal{B} \times \mathcal{A} \to [0,1]$, where $\pi(a \mid b_t)$ denotes the probability of selecting action a given belief state b_t.

0-1-8 Bellman Expectation Equation

The key equation we use to find optimal behaviour within a given MDP is the discrete-time Bellman Expectation Equation (see Equation 3).

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[r(s,a) + \gamma \sum_{s'} P(s' \mid s, a) \sum_{a'} \pi(a' \mid s') Q^\pi(s', a') \right], \tag{3}$$

where:

- $r(s,a)$ is the expected immediate reward for taking action a in state s,
- $\pi(a' \mid s')$ is the policy, giving the probability of choosing action a' in state s',
- $Q^\pi(s', a')$ is the expected long-term return of the state-action pair (s', a') when following policy π,
- all other notation is as defined in Section 0-1-7.

0-1-9 Principle of Optimality

The Bellman Expectation Equation computes the expected long-term return of a state-action pair under a single policy, but the goal of RL is to find the *optimal* policy π^*. An optimal policy can be decomposed into two components:

- an optimal first action at state s,
- followed by an optimal policy from the successor state s'.

Depending on the problem definition, we choose to either compute the optimal state-value function, or the optimal state-action-value function. These equations are called the Bellman optimality equations and are a necessary and sufficient condition of optimality. They are detailed in Equations 4 and 5 below, respectively. For finite-horizon problems these are solved by backward induction; for the infinite-horizon discounted case they are solved as a fixed-point problem, for example via value iteration.

Both the optimal state-value function (Equation 4), and the optimal state-action-value function (Equation 5) will give the same result. The choice of which formulation to use depends on what information is available. Although $P(s' \mid s, a)$ appears in both Bellman optimality equations, q^* can be learned by sampling transitions from the environment without ever evaluating that sum explicitly, and the greedy policy $\pi^*(s) = \arg\max_a q^*(s, a)$ requires no model to evaluate. Learning v^* by sampling is possible, but extracting the greedy policy still requires $P(s' \mid s, a)$ explicitly, making q^* the natural choice for model-free RL. In model-based learning there is no particular extra ease afforded by either formulation: it merely depends on whether action-centric notation is preferable, or if there are significant constraints limiting our ability to compute v*.

$$v^*(s) = \max_a \left(r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) v^*(s') \right) \tag{4}$$

$$q^*(s, a) = r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) \max_{a'} q^*(s', a') \tag{5}$$

Equivalent control-theory formulation

Alternatively, in the control theory community, this same problem is instead formulated as the minimisation of an infinite-horizon discounted cost function such as in Equations 6 and 7 for discrete-time, and Equation 9 for continuous-time. This equivalent formulation is given below for reference and is not required to follow the remainder of this thesis.

State-value (V / J) form:

$$J^*(x) = -V^*(x) = \min_{u \in \mathcal{U}(x)} \left(g(x, u) + \alpha\, \mathbb{E}_w \left[J^*(f_{\mathrm{d}}(x, u, w)) \right] \right) \tag{6}$$

Action-value (Q-function) form:

$$Q^*(x, u) = g(x, u) + \alpha\, \mathbb{E}_w \Big[\min_{u' \in \mathcal{U}(f_{\mathrm{d}}(x,u,w))} Q^*(f_{\mathrm{d}}(x, u, w), u') \Big] \tag{7}$$

Relationship between Q and J:

$$J^*(x) = \min_{u \in \mathcal{U}(x)} Q^*(x, u) \tag{8}$$

The continuous-time Hamilton-Jacobi-Bellman form is obtained by taking the limit of Equation 6 as the time between control inputs Δt goes to zero, with the discount factor and discount rate related by $\alpha = e^{-\rho \Delta t}$:

$$\rho J^*(x) = \min_{u \in \mathcal{U}(x)} \left[g(x,u) + \nabla_x J^*(x)^\top f_\mathrm{c}(x,u) + \frac{1}{2} \operatorname{Tr}\Big(\sigma(x,u)\sigma(x,u)^\top \nabla_x^2 J^*(x) \Big) \right] \tag{9}$$

- $J^*(x)$ is the optimal cost-to-go from state x at time t
- $g(x,u)$ is the instantaneous cost
- $f_\mathrm{d}(x,u,w)$ is the discrete-time transition map, returning the successor state
- $f_\mathrm{c}(x,u)$ is the continuous-time drift, giving the deterministic rate of change $\dot{x}$
- $\sigma(x,u)$ is the diffusion matrix (which disappears in deterministic cases)
- $\nabla_x J^*(x)$ is the gradient of the value function
- $\nabla_x^2 J^*(x)$ is the Hessian of the value function
- the min is taken pointwise over $u \in \mathcal{U}(x)$
- $\mathcal{U}(x)$ is the set of admissible controls available in state x
- w is the exogenous disturbance, with $\mathbb{E}_w$ the expectation taken over it
- $\alpha \in (0,1)$ is the per-step discount factor
- $\rho > 0$ is the continuous-time discount rate

Exact solutions to the Bellman optimality equations (Equations 4 and 5) guarantee convergence to the globally optimal policy. However, this is only possible when the state and action spaces are small enough to model exactly, and the transition model $P(s' \mid s,a)$ is known. This is termed *indirect RL* [59], since the policy is recovered only after the value function has been solved for. Unfortunately, many physical systems do not satisfy these conditions: their state spaces are continuous and high-dimensional, and their transition dynamics are unknown or too complex to model exactly, rendering exact solution of Equations 4–9 intractable. In such cases, *direct RL* methods are used instead: rather than solving for v^* or q^* exactly, a parameterised policy is improved directly via gradient ascent on an estimate of the expected return, sampled from interaction with the environment. This is discussed in more detail for the specific case of Proximal Policy Optimisation used in this thesis in Section 2-5.

Putting it into Practice

Simply put, we are following this flow of events:

Reinforcement learning	**Control theory**
Initialize policy π_θ	Initialize policy π
$\downarrow$	$\downarrow$
Observe state s_t	Observe state x_t
$\downarrow$	$\downarrow$
Select action $a_t \sim \pi_\theta(s_t)$	Select control input $u_t \sim \pi(x_t)$
$\downarrow$	$\downarrow$
Apply action to system	Apply control input to system
$\downarrow$	$\downarrow$
Observe reward r_t and next state s_{t+1}	Observe cost $g(x_t, u_t)$ and next state x_{t+1}
$\downarrow$	$\downarrow$
Update policy to maximise cumulative reward	Update policy to minimise cumulative cost
$\downarrow$	$\downarrow$
Repeat until convergence	Repeat until convergence

0-2 Project Proposal

0-2-1 Problem Statement

There is often a shortfall between the realised annual energy production of a wind farm and the predictions. A major cause of this mismatch are the wake effects: upstream turbines generate a wake that extends to the next turbine downstream, reducing the amount of power it can generate because it then does not experience the true free-wind field. This is exacerbated when the wind direction is aligned with the largest axis of aligned turbines. Inherent in this problem is the fact that wake mitigation is a whole-of-farm level affair: it is not useful to consider this only for stand-alone turbines because the key point is the *interaction* between turbines. Due to space constraints it is typically not possible to place turbines even further apart, and existing wind farms cannot just be re-built; we have to work within existing physical constraints. Therefore, adaptive control methods for steering the wake away from downstream turbines are of interest. Of the adaptive methods for wake mitigation, turbine yaw-misalignment, induction control and repositioning based steering techniques are the most promising. However, all tests of these techniques suffer from the same problem: they all rely on either an over-simplified wake model, or they use high fidelity models that are too slow for testing real-time controllers. This is problematic because the primary mechanism being tested here is wake-steering, so the wake being studied should be realistic. The most under-researched combination of these areas is turbine repositioning via closed-loop control with a realistic time-varying wake model.

0-2-2 Research Gap

Since floating wind turbines can translate, they have an extra available degree of freedom for wake steering. This is an under-researched area. The use of time-varying wake models to validate or compute offline solutions for look-up tables for wake steering is also rare. Rarer still is the use of a time-varying wake model *with 3D wake curl behaviour*. Developing in parallel to this is research into RL optimal controllers for collective wind farm control. RL for wake steering is an especially nascent research topic. There is an open question as to how to accurately learn optimal dynamic wake-steering trajectories using RL, encoding the wind farm and reinforcement learning behaviour in a POMDP. A POMDP RL controller trained on a time-varying 3D wake model (such as the Free Vortex Wake Model [4]) or a laboratory setup could act as a first proof-of-concept for RL control for dynamic wake steering.

LUT controllers are the standard approach to wind farm control. The LUTs are precomputed offline and indexed by a small set of measured or estimated states and fast enough for real-time deployment, but they are only as accurate as the (typically steady-state) model used to generate them, and scale poorly as the number of relevant state dimensions grows [60]. MPC can directly optimise over a dynamic model at each timestep and adapt to time-varying conditions, but the online optimisation this requires is often too computationally expensive to run in real-time at wind-farm scale [41]. Reinforcement learning sits between these two: training is typically far more computationally expensive than either LUT generation or a single MPC solve, since it requires a very large number of environment interactions to converge, but then a trained policy can be deployed as a single forward pass through a neural network — in effect a LUT with a much higher-dimensional, learned, and non-linear mapping from state to action, without the same increased storage cost [61].

This trade-off inspires a strategy wherein the RL controller is trained offline in a high-fidelity time-varying wake simulation, and only the resulting policy is deployed on physical hardware or a real wind farm.

0-2-3 Research Objective and Questions

To guide the research direction in this thesis, and provide targets against which to compare the results, the following research objective, questions and contributions are defined. They are in order of most abstract (the research objective) to most specific (the contributions).

Research Objective: The objective of this thesis is to design, implement, and evaluate a proof-of-concept reinforcement-learning controller for dynamic repositioning of floating wind turbines via wake steering in a physics-based dynamic wake model, and then use these results to design a purpose-built laboratory setup for future testing of dynamic repositioning control techniques.

This objective requires three main components:

1. a dynamic, time-varying wake flow model capable of representing turbine repositioning motion,
2. a reinforcement learning controller formulated to exploit that model, and
3. a laboratory setup capable of physically testing the required repositioning motion.

Prior work by van den Broek et al. [4] established the Free Vortex Wake Model (FVWM) as a control-oriented dynamic wake framework, but without support for turbine repositioning motion. In this thesis, the following is documented: an extension to the FVWM to include yaw-induced repositioning (Component 1), formulation and training of a POMDP-based RL controller on the extended FVWM (Component 2), and the design, build and test of a laboratory rig to physically evaluate dynamic repositioning trajectories (Component 3).

This research objective is synthesised into six research questions as follows:

RQ 1 How can we formulate the POMDP of the wind farm for wake steering?

RQ 2 Is RL-based repositioning control feasible for online and/or offline deployment?

RQ 3 What wake-steering policies does the RL agent learn, and how do they compare against baseline strategies?

RQ 4 How much power can dynamic repositioning recover, compared to a static baseline?

RQ 5 What effect do configuration parameters have on RL controller performance, especially the length of aggregated historical state measurements as a proxy for the unobservable states?

RQ 6 What are the design specifications for a dynamic turbine repositioning laboratory rig and how does it perform?

0-2-4 Contribution and Outline

Contributions: This thesis contributes the following new information, methods and equipment:

Simulation Contributions

Contribution 1 A POMDP formulation and RL controller for dynamic wake-steering repositioning, trained on the extended FVWM.

Contribution 2 Characterisation of the optimal repositioning signals discovered by the RL controller, benchmarked against literature strategies (Starink et al. [10], van den Broek et al. [4]) and standard grid search techniques.

Contribution 3 Quantification of the training time requirements for the RL controller in the FVWM.

Contribution 4 An extension to van den Broek et al.'s Free Vortex Wake Model (FVWM) [4, 9] to simulate yaw-induced turbine repositioning motion.

Laboratory Contributions

Contribution 5 Specification, mechanical design and build, and experimental commissioning of a WindShaper-compatible laboratory rig for wake-steering repositioning control.

Contribution 6 Empirical characterisation of the tracking fidelity and repositioning limitations of the laboratory actuators, including quantified commanded-vs-achieved error in frequency, amplitude, and phase.

0-3 Thesis Outline

This thesis is organised into the following sequence of chapters:

Introduction (this chapter)
Contains the research motivation, background information that introduces readers with mechanical and electrical engineering training to the preliminary concepts required to comprehend the content of this thesis, and the project proposal including research questions and contributions.

Chapter 1: The Simulation Environment: Incorporating Turbine Repositioning into the Free Vortex Wake Model
The system (an array of two turbines), the wake simulator (van den Broek's FVWM [4]), and the method for integrating turbine repositioning motions into the FVWM are presented.

Covers: RQ 1; Contribution 4

Chapter 2: Reinforcement Learning Controller Formulation
Lays the foundations for the simulation and experimental testing campaigns — defines the POMDP tuple and the RL control strategy.

Covers: RQs 1, 3, 5; Contribution 1

Chapter 3: Case Study 1: Simulation with the Free Vortex Wake Model
The results of experiments with the RL controller from Chapter 2, in the extended FVWM environment from Chapter 1, are discussed.

Covers: RQs 2, 3, 4, 5; Contributions 1, 2, 3

Chapter 4: Laboratory Setup
Informed by the simulation environment and results in Chapters 1 and 3, the laboratory rig design is specified.

Covers: RQ 6; Contribution 5

Chapter 5: Case Study 2: Laboratory Experiments
The results of the experimental test campaign are included here.

Covers: RQs 4, 6; Contributions 5, 6

Chapter 6: Conclusion and Future Work
Here, the entirety of the thesis is summarised, and recommendations for future work are provided.

Chapter 1

The Simulation Environment: Incorporating Turbine Repositioning into the Free Vortex Wake Model

In this chapter, the simulation environment is defined. The decision to use an array of two turbines for tests is discussed in Section 3-1, the reference turbine design details are presented in Section 3-2, the existing Free Vortex Wake Model (FVWM) is recapitulated in Section 3-3, and the extension of this FVWM to include yaw-induced repositioning motion is described in Section 3-4. In toto, this system environment provides the turbine motion and time-varying wake dynamics that the RL agent formulated in Chapter 2 interacts with and learns from. The following research question is partially addressed in this chapter (the environment is introduced), and the following contribution is made:

RQ 1 — How can we formulate the POMDP of the wind farm for wake steering?

Contribution 4 — An extension to van den Broek et al.'s Free Vortex Wake Model (FVWM) [4, 9] to simulate yaw-induced turbine repositioning motion.

1-1 The Array of Two Turbines

In order to test the wake-escape abilities of turbine repositioning control, at least two turbines are required. In order to keep computational costs low, and since this is a first proof-of-concept of RL for dynamic turbine repositioning, it is prudent to start with the smallest case: two turbines. The two turbines are placed five rotor diameters apart as is the industry standard for commercial wind farms. They are both capable of ± 45 degrees yaw angles, and the equivalent associated repositioning distance; and they are constrained to translation along the y-axis but with no floater rotation or x-axis translation, as depicted in Figure 1-1. The axes in this figure are scaled by the turbine diameter, because this is the convention used in the Free Vortex Wake Model described in the following sections. A second figure, Figure 1-2, is also provided, to help the reader visualise the relationship between turbine yaw angle, position and the wake. Important to note is that when the turbine is yawed such that the left face points downwards to negative-y, the wake will deflect in a downwards curve, and the turbine under YITuR will reposition upwards (see the bottom tile of Figure 1-2). This is the phenomenon visible, and convention used, throughout this thesis.

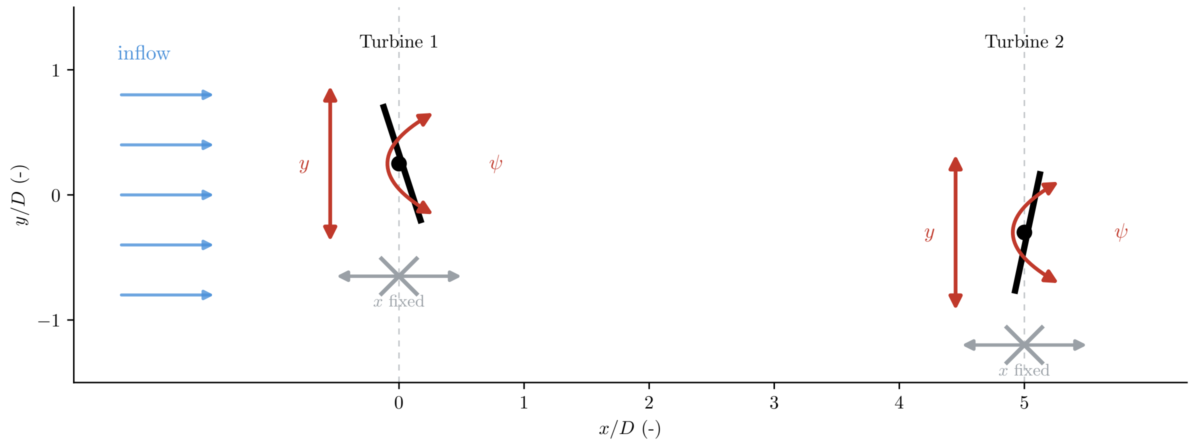


Figure 1-1: Depicting the range of motion and layout of the two-turbine array.

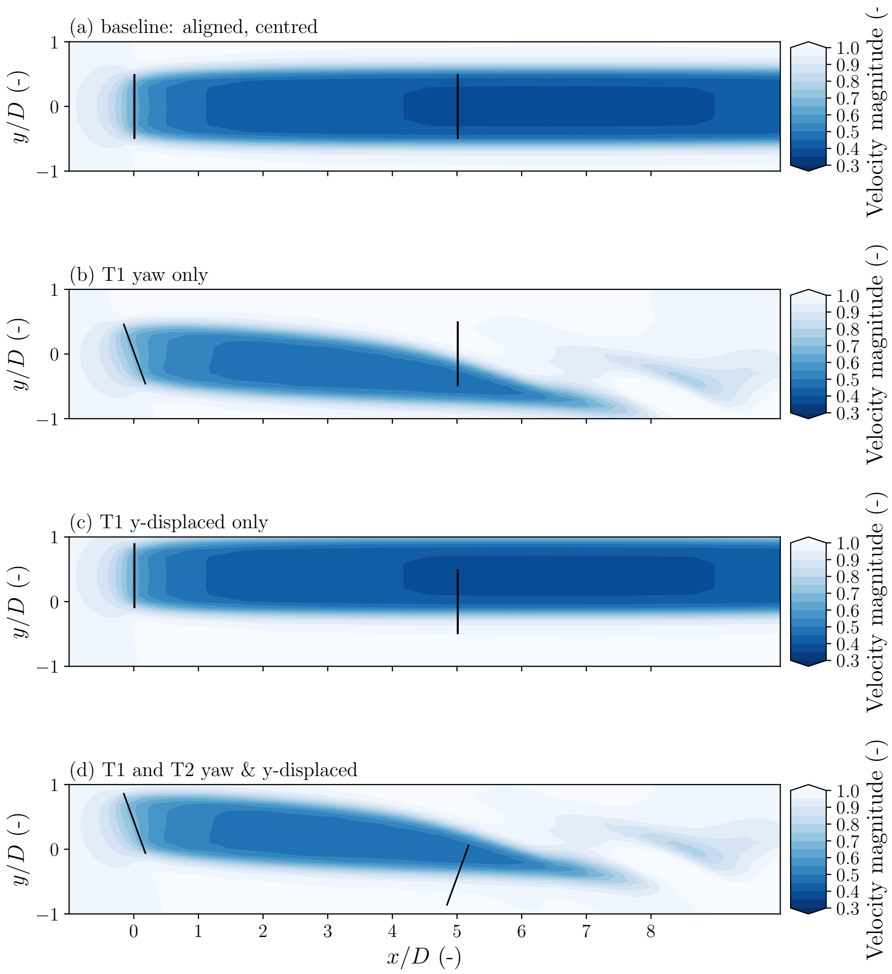


Figure 1-2: Example turbine motions with only the upstream turbine's wake shown.

1-2 NREL 5 MW Reference Turbine

The reference turbine against which all scaled models (both computer simulation and laboratory) are produced in this thesis is the NREL 5 MW Reference Turbine defined in [8].

Table 1-1: Key specifications of the NREL 5 MW reference wind turbine [8].

Parameter	**Value**
Rating	5 MW
Configuration	Upwind, 3 blades, variable-speed, collective pitch
Rotor / Hub diameter	125 m / 3 m
Hub height	90 m
Cut-in, rated, cut-out wind speed	3, 11.4, 25 m/s
Cut-in, rated rotor speed	6.9, 12.1 rpm
Rated tip speed	80 m/s

1-3 What is the Free Vortex Wake Model

The Free Vortex Wake Model (FVWM) was developed by van den Broek et al. within the Delft Center for Systems and Control at TU Delft. Their original code, underlying mathematics and experiments on steady state wake steering optimality were published in 2022 [4]. The FVWM is based on the actuator-disc representation of a wind turbine and it comes in two versions: 2-D and 3-D. Only the 3-D version preserves the wake-curl phenomenon that is crucial for not over-predicting power gains under dynamic yawing, therefore, the 3-D model is used in this thesis. It is a dynamic dimensionless model wherein a set number of vortex particles represent the wake at any given time instant, acting like individual operating points. At each successive time instant, new particles are released at the rotor of the upstream turbine and existing particles are advected downstream with the downstream-most particles removed. The 2-D particle release and advection, as well as the 3-D case with rings instead of particles, are shown in Figure 1-3.

This process is formulated as a discrete-time, non-linear state-space system,

$$\boldsymbol{q}_{k+1} = f(\boldsymbol{q}_k, \boldsymbol{m}_k), \tag{1-1}$$

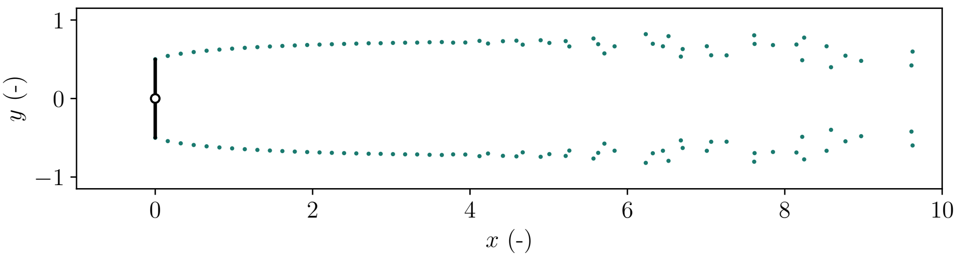
$$\boldsymbol{y}_k = g(\boldsymbol{q}_k, \boldsymbol{m}_k), \tag{1-2}$$

where the state $\boldsymbol{q}$ collects the vortex element positions $\boldsymbol{X}$, circulations $\boldsymbol{\Gamma}$, stored inflow velocities $\boldsymbol{U}$, and the previous control input $\boldsymbol{M}$, and the control vector $\boldsymbol{m}$ contains the axial induction a and yaw angle ψ of each turbine.

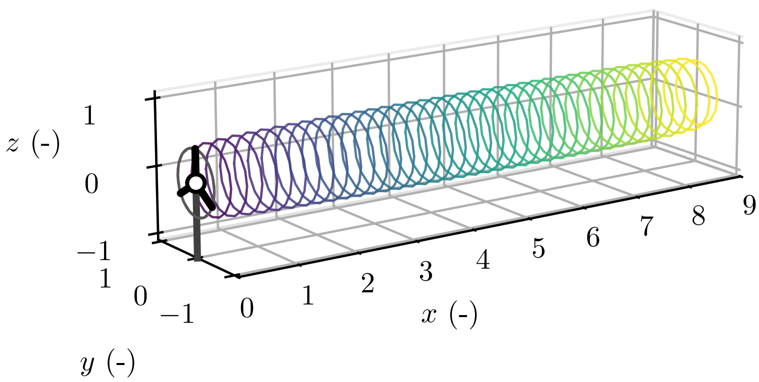
A forward-Euler update driven by the local free-stream and induced velocities is used to describe the advection of existing vortex ring positions downstream,

$$\boldsymbol{x}_i^{(b)}\big|_{k+1} = \boldsymbol{x}_i^{(b-1)}\big|_k + h\left(\boldsymbol{u}_{\infty,i}^{(b-1)} + \boldsymbol{u}_{\text{ind}}\Big(\boldsymbol{x}_i^{(b-1)}, \boldsymbol{q}\Big)\right)\Big|_k, \tag{1-3}$$

where $\boldsymbol{x}_i^{(b)}$ denotes the position of vortex element i in ring b (rings are indexed by time-step of release, with $b = 0$ the most recently shed ring), h is the fixed simulation time step, $\boldsymbol{u}_{\infty,i}^{(b-1)}$ is



(a) 2-D free vortex wake particle release and advection depiction.



(b) 3-D free vortex wake ring release and advection depiction.

Figure 1-3: Free vortex wake model particle release and advection representations.

the stored local free-stream velocity at that point, and $\boldsymbol{u}_{\text{ind}}$ is the induced velocity at any point given by the Biot–Savart contribution summed over all vortex elements in the wake.

New vorticity is released at the rotor plane each time-step according to

$$\Gamma_i^{(0)}(\boldsymbol{q}, \boldsymbol{m}) = c_t'(a)\, \tfrac{1}{2}\, (\boldsymbol{u}_r \cdot \boldsymbol{n}(\psi))^2\, h, \tag{1-4}$$

where $c_t'(a)$ is a local thrust coefficient derived from momentum theory, a is the induction factor, $\boldsymbol{u}_r$ is the disc-averaged rotor velocity, and $\boldsymbol{n}(\psi)$ is the unit vector normal to the rotor disc, pointing in the downstream direction and rotated by the yaw angle ψ.

Turbine power output, which forms the basis of the RL reward signal and part of the observation space used in the Case Study 1 in this thesis, is then computed as

$$P_i = \tfrac{1}{2} c_p'(a)\, A_r\, (\boldsymbol{u}_r \cdot \boldsymbol{n}(\psi))^3\, . \tag{1-5}$$

where $c_p'(a)$ is the local power coefficient and A_r is the rotor area.

A discrete adjoint addition to the model also exists wherein exact gradients of a receding-horizon power objective can be computed. This adjoint option is not used in this thesis because it adds excessive computational complexity. Without it, the 3-D FVWM instead serves purely as a fast, physics-based simulation environment for training and evaluating RL agents.

Downstream turbines that are not modelled with their own vortex wake are instead represented as *virtual turbines*: their rotor-disc-averaged velocity is sampled from the flow field induced by the upstream turbine(s), but they do not themselves release vorticity and therefore do not influence the wake. To account for the fact that a real turbine would extract energy from the flow and locally reduce its downstream velocity, the disc-averaged velocity sampled at the virtual turbine is scaled by its own induction factor, $\boldsymbol{u}_r^* = (1-a)\boldsymbol{u}_r$, before being used to compute its power output via Eq. (1-5) [4]. This allows the power production of a downstream turbine to be estimated without the added computational cost of simulating its wake, at the cost of neglecting any influence that turbine's own induction would have on the flow further downstream.

1-3-1 On the Selection of the FVWM

The FVWM was chosen as it is the only available wind farm wake modelling tool capable of meeting the five desired criteria for this research: the wake model must be dynamic, accurately model wake curl, be control oriented, 3-D and laptop-feasible. The extended results of the comparison study against other available models that led to this decision is available in Table 2 in the previous Section 0-1-5. The FVWM is unique in that it is both capable of capturing the 3-D wake curl behaviour that leads to realistic predictions of power gain under dynamic wake steering scenarios, *and*, it is control-oriented. It is, however, limited in that it is designed to only model the propagation of one upstream turbine's wake on a number of downstream 'virtual' turbines. A 'virtual' second turbine placed five diameters downstream of the first turbine sits within the FVWM's zone of validated accurate wake behaviour. After that point, however, the wake characteristics deteriorate into the far wake region, limiting the model's accuracy for wake mixing and accumulation beyond a second turbine. Therefore, in using the FVWM, we are limited to the two-turbine case: one upstream turbine with full-wake modelling activated, and one downstream 'virtual' turbine that only experiences the upstream turbine's wake and responds accordingly. Of course, in the future, methods such as van den Broek's own directed graph communication protocol [62] can be used to expand the FVWM for reliable use in multi-turbine tests. However, as described earlier in Section 1-1, the use of two turbines is both sufficient to understand the repositioning phenomenon, and suitable for the scope of this proof-of-concept study into collaborative dynamic turbine repositioning for wake escape with RL.

1-3-2 Solver Configuration Considerations

In the FVWM, the user can vary the parameters listed in Table 1-2. This table has been populated with a set of values from the FVWM 3-D examplar [9]. The most important consideration when changing these parameters is the relationship between core size and time step, and also the relationship between the number of rings and the time step. The values shown in this table were chosen such that no inflation of the upstream turbine's power, and the smallest possible inflation of the downstream turbine's power occurs. The number of rings and elements can be reduced for the purpose of faster computation times, but this results in even larger inflations of the downstream turbine's power.

1-4 Incorporating Position Changes into FVWM

The FVWM does not include any option for turbine translation. Therefore, one contribution of this thesis is to build this into the FVWM (Contribution 1.4). The repositioning is effected in the manner invented by Han et al. [30]: using the aerodynamic thrust to effect YITuR. To do this, a low pass filter and steady-state gain were used to relate turbine yaw angle to lateral platform repositioning. As previously metioned, the reference turbine for the calculations is the 5 MW reference floating wind turbine defined in [8]. Per the work in [10], the dynamic solutions are optimal for specific mooring line slackness: too taut or too slack and static solutions are optimal instead. In the FVWM, a position change manifests thus: for T1, vortex rings are released at each time step at the new location of the rotor; for T2, the power is evaluated at

Table 1-2: Free-vortex wake model configuration parameters, following van den Broek et al. (2022) [4, 9]. All parameters are non-dimensionalised by rotor diameter and inflow speed.

Parameter	Symbol	Value
Spatial dimension	n_d	3
Time step	h	0.3
Number of wake rings	n_r	80
Elements per ring	n_e	16
Vortex core size	σ	0.16
Modelled turbines	n_t	1
Virtual turbines	-	1
Turbine position (w.r.t. rotor centre)	(x, y, z)	(0, 0, 0)
Virtual turbine position	(x, y, z)	(5.0, 0, 0)

the new location of the rotor. Let us commence with defining the low pass filter and then move on to the description of the yaw-to-position gain which encodes the mooring line slackness:

1-4-1 Low Pass Filter for Floater Dynamics

The yaw signal is filtered through a continuous-time second-order low pass filter with transfer function:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{1-6}$$

where ω_n is the natural frequency (rad/s) and ζ is the damping ratio. The damping ratio is set to $\zeta = 0.7$. This continuous-time filter is discretised using the bilinear (Tustin) transform with sample time Δt,

$$s = \frac{2}{\Delta t}\frac{1 - z^{-1}}{1 + z^{-1}}, \tag{1-7}$$

yielding discrete coefficients (b, a) such that the filtered signal is

$$y[n] = \frac{1}{a_0}\left(\sum_{i=0}^{2} b_i\, x[n-i] - \sum_{j=1}^{2} a_j\, y[n-j]\right). \tag{1-8}$$

1-4-2 Dimensional Analysis - Frequency Scaling with Strouhal Number

To develop representative simulation and laboratory models, the 5 MW NREL reference floating wind turbine physical parameters must be scaled appropriately. The first step in this process is non-dimensionalisation. The dynamic repositioning effectiveness is directly related to the yaw and translation forcing frequency as described by Starink et al. in [10]. This can be expressed in a non-dimensional fashion by the Strouhal number:

$$\mathrm{St} = \frac{f_e D}{U_\infty} \tag{1-9}$$

where f is the actuation frequency, D a characteristic length (here, the turbine's rotor diameter), and U_∞ the characteristic free wind speed.

To preserve the relevant non-dimensional dynamics between the full-scale reference case reported by Starink et al. [10] (mooring line radius $R = 775$m, rotor diameter $D = 125$m, and inflow velocity $U = 8.2$m/s) and the non-dimensional scaled FVWM simulation setup (reference rotor diameter $D = 1, (-)$), the Strouhal number is held constant, such that the filter's natural frequency, the forcing frequency, and the Nyquist frequency can all be compared on a common non-dimensional basis. These quantities are summarised in the table below, and also shown in Figure 1-4.

Table 1-3: Characteristic full-scale and non-dimensional equivalent frequencies of the floater case reported by Starink et al. [10] (mooring line radius $R = 775\,\mathrm{m}$, rotor diameter $D = 125\,\mathrm{m}$, and inflow velocity $U = 8.2\,\mathrm{m\,s^{-1}}$). Computed using FVWM simulation clock $\Delta t = 0.3$, non-dimensional, where the equivalent full-scale sample time is $\Delta t \cdot D/U \approx 4.57\,\mathrm{s}$.

Quantity	f (Hz) (full scale)	$St = fD/U$ (–)	$\omega = 2\pi\, St$ (rad/–)
Forcing frequency, f_{forcing}	0.002273	0.0346	0.2177
Filter natural frequency, f_n	0.003916	0.0597	0.3751
Filter cutoff ($-3\,\mathrm{dB}$), f_c	0.003955	0.0603	0.3789
Nyquist frequency, f_{Nyq}	0.1093	1.667	10.47

1-4-3 Choice of Yaw-to-repositioning Gain

A gain was also defined to encode the scaling between yaw angle amplitude and the resulting repositioning distance:

$$g = \frac{\Delta y_{\text{ss}}}{D\,\psi_{\text{ss}}}, \tag{1-10}$$

where g is the non-dimensional steady-state yaw-to-displacement gain, Δy_{ss} is the steady-state lateral displacement, $D = 125$ m is the reference rotor diameter of the NREL 5 MW wind turbine, and ψ_{ss} is the steady-state yaw angle (in degrees).

Three reference gains were identified from the literature, each associated with a different mooring line length:

$$g_{\text{Han}} = 0.0028 \text{ nd/deg}, \quad L_{\text{Han}} = 835.5\,\mathrm{m} \quad [38], \tag{1-11}$$

$$g_{\text{Gao}} = 0.0318 \text{ nd/deg}, \quad L_{\text{Gao}} = 970.0\,\mathrm{m} \quad [40], \tag{1-12}$$

$$g_{\text{Starink}} = 0.03175 \text{ nd/deg}, \quad L_{\text{Starink}} = 900.0\,\mathrm{m} \quad [43]. \tag{1-13}$$

Three independent estimates of the gain at $L_{\text{trial}} = 775\,\mathrm{m}$ were calculated by scaling each reference gain by L_{trial} using the following equation (based on the assumptions that the catenary mooring stiffness scales as $k \propto 1/L$, and that the displacement gain scales approximately linearly with mooring line length, $g \propto L$):

$$g_{\text{trial}} = g_{\text{ref}} \cdot \frac{L_{\text{trial}}}{L_{\text{ref}}}, \tag{1-14}$$

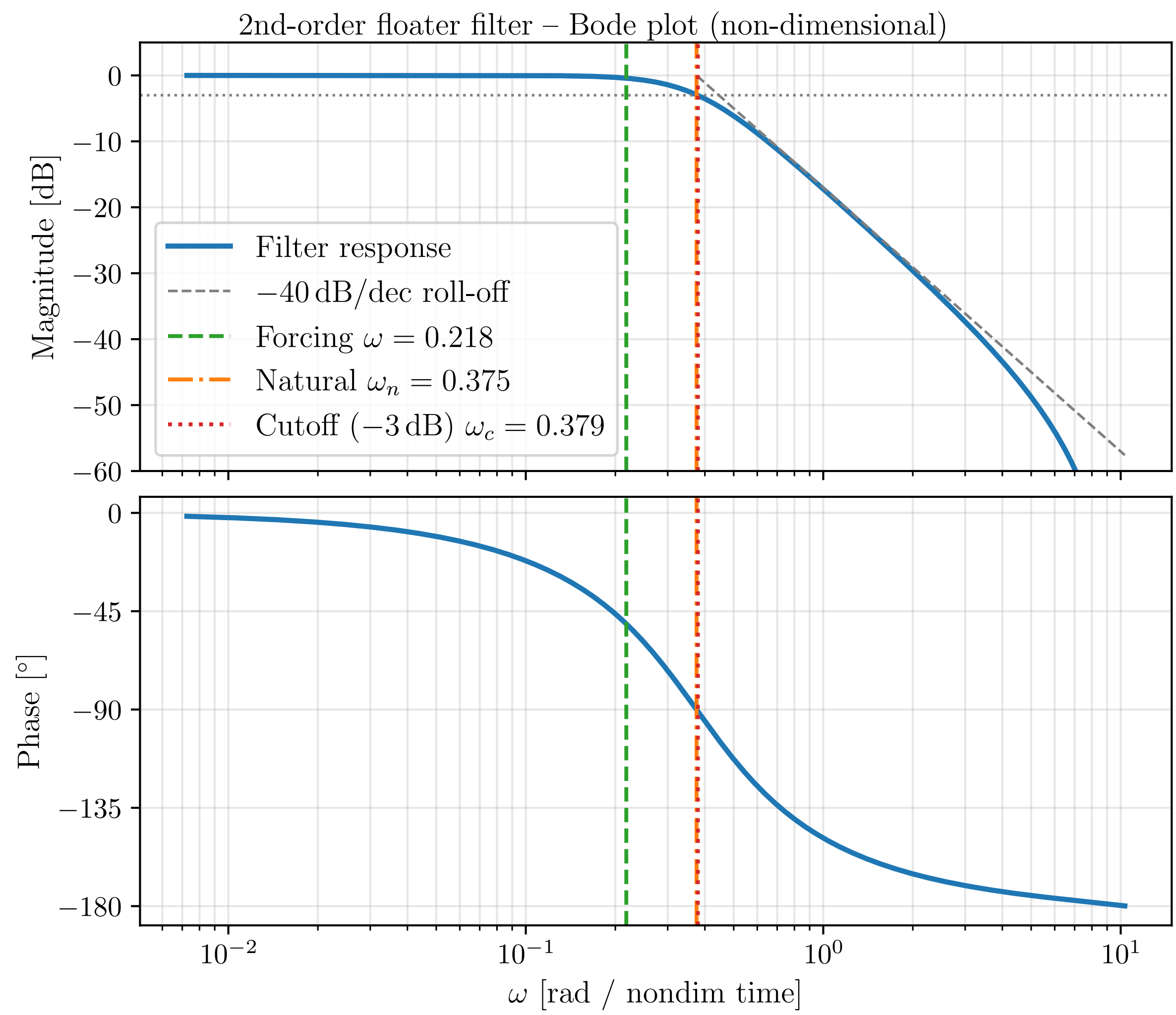


Figure 1-4: Bode plot of filter design.

A value close to the mid-point of these values was then adopted for the model: $g = 2/125 = 0.016\,\mathrm{nd/deg}$. In physical terms, this gain means that a steady yaw command of $\psi_{ss} = \pm 45°$ produces a steady-state lateral repositioning of $\Delta y_{ss} = g\,\psi_{ss}\,D \approx 0.72\,D$, or roughly 90 m for the reference NREL 5 MW turbine, which is physically reasonable given the mooring line lengths considered in the literature.

1-4-4 Implementation Coding Mechanics

The original FVWM defines the full control vector $\mathbf{m}$ as,

$$\mathbf{m} = \begin{bmatrix} a_0 & \psi_0 & a_1 & \psi_1 \end{bmatrix}^\top, \tag{1-15}$$

for a two-turbine configuration with both turbines operate at $a = 0.33$ and turbine yaw angle ψ.

In this thesis, the control vector of the FVWM is extended to take x, y, and z position values that can vary over a specified time horizon. For a single turbine, the extended per-turbine control vector becomes:

$$\mathbf{m}_i = \begin{bmatrix} a_i & \psi_i & x_i & y_i & z_i \end{bmatrix}^\top, \tag{1-16}$$

where x_i, y_i, and z_i denote the commanded turbine repositioning offsets from the turbine's nominal (baseline) position, in the global x, y, and z directions respectively. Note that x_i and z_i stay constant, and only y_i is modified based on the filtered yaw angle command. The full control vector $\mathbf{m}$ for a wind farm of N turbines is then the concatenation of the N per-turbine vectors $\mathbf{m}_i$, giving a control vector of length $5N$ rather than the original $2N$.

1-5 FVWM Turbine Repositioning Extension Conclusion

In this chapter, the two-turbine array layout and range of motion of these turbines was presented. The states and governing equations of the Free Vortex Wake Model were detailed, as was the extension developed to incorporate turbine repositioning motion (Contribution 4). The FVWM already encodes the wake as a discrete-time, non-linear state-space system, with vortex advection and circulation shedding computed at each time-step. To this, a new module was added: a low pass filter with a scaled yaw-to-displacement gain used to translate turbine yaw commands into repositioning motion, consistent with the theory of aerodynamic-thrust-based YITuR. Since the accuracy of the wake advection reduces after the second turbine in the FVWM, analysis is constrained to two turbines (T1 upstream and T2 downstream), rather than a larger farm. This is reasonable as the mechanism of wake escape via YITuR is visible with two turbines, and, this study is intended to be a proof-of-concept of the phenomenon, not a full-farm pre-production rollout validation.

Chapter 2

Reinforcement Learning Controller Formulation

In this chapter the formulation of the model-free reinforcement learning controller is established, applying the POMDP-based approach introduced in Section 0-1-7 to the dynamic turbine repositioning problem. The chapter begins by justifying the choice of a model-free approach (Section 2-1), before formally defining the problem as a POMDP (Section 2-3), and describing the policy optimiser (Section 2-5). This chapter addresses the following research questions and makes the following contribution:

RQ 1 How can we formulate the POMDP of the wind farm for wake steering?

RQ 3 What wake-steering policies does the RL agent learn, and how do they compare against baseline strategies?

RQ 5 How does partial observability of the flow field affect controller performance?

RQ 7 How can the controller's action space be constrained to guarantee safe turbine motion?

Contribution 3 A POMDP formulation and RL controller for dynamic wake-steering repositioning, trained on the extended FVWM.

2-1 Model-free RL Justification

The objective is to develop a real-time controller for an array of two wind turbines. The question then becomes: will model-free or model-based RL be appropriate. In a real wind farm, the complex system model is not accessible to the controller, which is also why classical MPC is not deployed on wind turbines. One option is to learn a surrogate dynamics model of the

FVWM via function approximation — for example, a neural network trained through system identification to reproduce the wake dynamics — and use it for model-based control. However, such a learned model introduces model bias that is difficult to bound and that compounds over the long prediction horizons imposed by wake advection delays. The alternative, adopted here, is to learn a control policy directly from interaction with the FVWM, mapping observations of yaw angle, turbine position, and wind speed to yaw actions. This model-free approach avoids model bias entirely, at the cost of sample efficiency [59]. Model-free methods are typically sample-hungry, but this is acceptable here because the FVWM is relatively cheap to evaluate and training is performed offline. Model-free RL is therefore chosen as the paradigm for this project.

2-2 Objective Function — $J(\pi)$

Let us commence by stating the end goal upfront. The agent's objective is to learn a policy π that maximises the expected discounted return,

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right], \tag{2-1}$$

where $\gamma \in [0, 1]$ is the discount factor and T is the episode horizon. These terms are described in detail in Section 2-3.

2-3 POMDP Formulation

To develop a controller for this situation, the control problem is modelled as a POMDP. We use a *partially observable* MDP because the true wake state is unobservable. Therefore, we must estimate it. The tuple representing this POMDP is presented in Equation 2-2 below:

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \mathcal{Z}, \gamma, \mu_0, H\}. \tag{2-2}$$

In each of the following subsections, the meaning of each element of this tuple is defined.

2-3-1 State Space — $\mathcal{S}$

The first element of this POMDP tuple is the state space, denoted $\mathcal{S}$. In POMDP, this is the complete extended Free Vortex Wake Model (introduced in Chapter 1). This is the 'world' that the RL controller then learns in.

2-3-2 Action Space — $\mathcal{A}$

The yaw angle is the only control action required to effect coupled yaw-repositioning wake steering. (As previously described in Section 0-1-4, the aerodynamic thrust force due to yaw angle misalignment with the oncoming wind direction enables relocation.) Therefore, the action space is:

$$\mathcal{A} = \{\psi\}. \tag{2-3}$$

Actions are specifically *not* enforced to a specific type of signal because this is exactly what we wish to determine with reinforcement learning. From literature, we could hypothesise that the optimal trajectories are sinusoidal [10], but we do not want to pre-empt the result with such restrictions during training.

Excessively large changes in action commands are curtailed: a maximum rate of yaw angle change is set to 5 degrees per time step.

2-3-3 Observation Space — $\mathcal{O}$

The wind farm is modelled as a POMDP: the true state (the full wake state of the FVWM, see Chapter 1) is not directly accessible to the controller, which receives only partial observations of it. To cope with this partial observability, the agent conditions its action on a finite history of past observations, which serves as an approximate sufficient statistic for the underlying belief over the true wake state.

Each instantaneous observation consists of three quantities per turbine: a performance measure, the yaw angle, and the lateral position. Turbine performance is denoted generically by m; the specific quantity depends on the experimental method. In the computational experiments m is the array power output P (Case Study 1, Section 3), whereas in the physical wind-tunnel experiments m is realised via tower-strain measurements ε (Case Study 2, Section 5). Both are normalised to $[0,1]$; the case-specific normalisation is detailed in the corresponding section. The formulation below is written in terms of m for generality and should be read as $m = P$ or $m = \varepsilon$ as appropriate.

The instantaneous observation at time t is

$$\tilde{\boldsymbol{o}}_t = [\, m_{T1}, m_{T2},\, \psi_{T1}, \psi_{T2},\, y_{T1}, y_{T2}\,]^\top \in \mathbb{R}^{3n_{\text{turbines}}}, \tag{2-4}$$

with each component normalised as

$$\begin{aligned} m_i &\in [0,1], && \text{normalised performance measure}, && \text{(2-5)}\\ \psi_i &\in [-1,1], && \text{normalised yaw angle}, && \text{(2-6)}\\ y_i &\in [-1,1], && \text{normalised lateral position}, && \text{(2-7)} \end{aligned}$$

for $i \in \{1, \ldots, n_{\text{turbines}}\}$. The corresponding per-timestep observation set is

$$\tilde{\mathcal{O}} = [0,1]^{n_{\text{turbines}}} \times [-1,1]^{n_{\text{turbines}}} \times [-1,1]^{n_{\text{turbines}}}, \tag{2-8}$$

which for a two-turbine array ($n_{\text{turbines}} = 2$) is $\tilde{\mathcal{O}} = [0,1]^2 \times [-1,1]^2 \times [-1,1]^2 \subset \mathbb{R}^6$.

To provide temporal context, the agent observes a stacked history of the most recent H instantaneous observations, where H is the history length:

$$\boldsymbol{o}_t = (\tilde{\boldsymbol{o}}_{t-k}, \ldots, \tilde{\boldsymbol{o}}_t)\,, \qquad k = H - 1. \tag{2-9}$$

The full observation space is therefore

$$\mathcal{O} = \tilde{\mathcal{O}}^{\,H} \subset \mathbb{R}^{H \times 3n_{\text{turbines}}}, \tag{2-10}$$

which for the two-turbine array is $\mathcal{O} \subset \mathbb{R}^{H\times 6}$. The choice of history length H varies across the case studies and its effect is discussed in Chapter 3.

2-3-4 Transition Model — $\mathcal{P}$

Even though this is a model-free RL approach, a transition model still exists. The trial-and-error process interacts with this transition model, sampling from it as if it is a black box. We do not try to learn the underlying model. The transition model describes how the underlying state $s \in \mathcal{S}$ (the full wake state defined in Section 1) evolves in response to the agent's action. In general, this evolution is subject to process noise or unmodelled disturbances w, so the transition is stochastic:

$$\mathcal{P}(s' \mid s, a) = \int \delta\big(s' - f(s, a, w)\big)\, p(w)\, dw, \tag{2-11}$$

where $f(\cdot)$ is the (case-specific) state evolution function and $p(w)$ is the distribution of the disturbance acting on the system. The disturbance term w represents sources not captured by f, such as turbulent inflow, unresolved wake turbulence and other system noise. Continuing with this generic case-study agnostic notation, the state evolution is written:

$$s_{t+1} = f(s_t, a_t, w_t). \tag{2-12}$$

Note that the transition kernel is defined over the full wake state s, and not over the observation $o = \{m, \psi, y\}$. The observation is not a Markov state: knowing (m, ψ, y) at the current instant is not enough to predict the next state. Two wake fields can share the same (m, ψ, y) yet differ in their underlying vortex structure, and these two fields will in general evolve to different successor states. The observation therefore does not, on its own, capture the information needed to describe the dynamics. This is exactly why the problem is formulated as a POMDP, and why the agent conditions its policy on a history of observations (Section 2-3-3) rather than on a single instantaneous observation: the history recovers an approximate Markov statistic that the instantaneous observation lacks.

Remark (Case Study 1 — deterministic FVWM transition): In the computational case study, the environment dynamics are given by the FVWM under uniform, deterministic inflow with deterministic wake dynamics i.e. $w_t \equiv 0$. The transition function reduces to $f_{\text{FVWM}}(s_t, a_t)$, and the integral above collapses to a Dirac delta:

$$\mathcal{P}(s' \mid s, a) = \delta\big(s' - f_{\text{FVWM}}(s, a)\big), \qquad s_{t+1} = f_{\text{FVWM}}(s_t, a_t). \tag{2-13}$$

This is the special case of the general transition model above obtained by setting $p(w) = \delta(w)$.

Remark (Case Study 2 — stochastic lab transition): In the physical wind tunnel case study, turbulent inflow, tower vibrations, motor inaccuracies and mounting imperfections, mean the deterministic assumption no longer holds, and the full stochastic form

$$\mathcal{P}(s' \mid s, a) = \int \delta\big(s' - f_{\text{lab}}(s, a, w)\big)\, p(w)\, dw \tag{2-14}$$

must be used. Further details on f_{lab} and the quantification of noise and uncertainties are discussed in Chapter 5.

2-3-5 Reward Function — $\mathcal{R}$

The reward is defined as the sum of the normalised performance parameter m_i across all turbines in the array at each time step:

$$r_t = \sum_{i=1}^{n_{\text{turbines}}} m_{i,t}, \tag{2-15}$$

where $m_{i,t} \in [0,1]$ is the normalised performance parameter for turbine i at time t, as defined in Section 2-3-3. It is deliberate that the *combined* array output is used in the reward function as this is what permits wake-steering behaviour to emerge: a reduction in upstream turbine performance is only reinforced by the agent if it is offset by a larger gain in downstream turbine performance. Again, the specific formulation of m in each experiment will be described at that specific case study section in the results (Chapters 3 and 5).

2-3-6 Observation Function — $\mathcal{Z}$

The observation function $\mathcal{Z}$ specifies the probability of receiving observation $o \in \mathcal{O}$ given the underlying state s and action a:

$$\mathcal{Z}(o \mid s, a) = p(o \mid s, a). \tag{2-16}$$

This term describes the gap between the true unobservable wake state and the belief state available to the agent. In general, $\mathcal{Z}$ has two distinct sources of uncertainty: (i) the mapping from state to the underlying performance parameter m, which is itself imperfectly modelled, and (ii) sensing noise introduced by the process of acquiring o from m (instrumentation or estimation error).

Similar to the transition model, the observation function can be stochastic or deterministic. If we let the noise-free, true values be denoted by $m^\star(s,a)$, $\psi^\star(s,a)$ and $y^\star(s,a)$, the full observation vector is modelled as

$$\boldsymbol{o} = [m^\star(s,a), \psi^\star(s,a), y^\star(s,a)]^\top + \boldsymbol{\nu}, \qquad \boldsymbol{\nu} \sim p(\boldsymbol{\nu}), \tag{2-17}$$

where $\boldsymbol{\nu}$ is the vector of observation noise acting on each component. Whether each component of $\boldsymbol{\nu}$ is zero or nonzero differs between the two case studies, as described below.

Remark (Case Study 1 — deterministic observation function): For the special case where the observations are deterministic (as is the case for the computational case study, where $m^\star(s,a) = \bar{P}(s,a)$ is computed exactly from the FVWM state via momentum theory, and $\psi^\star(s,a)$, $y^\star(s,a)$ are the exactly known commanded yaw angle and lateral position) we have $p(\boldsymbol{\nu}) = \delta(\boldsymbol{\nu})$, and therefore the deterministic observation function is,

$$\mathcal{Z}(o \mid s, a) = \delta(o - \bar{P}(s,a)). \tag{2-18}$$

Remark (Case Study 2 — stochastic observation function): In the physical wind tunnel case study, m is derived from strain gauge measurements, while ψ and y are derived from motor and positioning encoders; all three are subject to sensing noise, vibrations and turbulent inflow, so all three components of $\boldsymbol{\nu}$ are nonzero. Accordingly, the observation function is stochastic,

$$\mathcal{Z}(\boldsymbol{o} \mid s, a) = \int \delta\big(\boldsymbol{o} - [m^\star(s,a), \psi^\star(s,a), y^\star(s,a)]^\top - \boldsymbol{\nu}\big)\, p(\boldsymbol{\nu})\, d\boldsymbol{\nu}, \tag{2-19}$$

with $p(\boldsymbol{\nu})$ determined through experiment calibration.

2-3-7 Discount Factor — γ

The discount factor $\gamma \in [0,1]$ sets the relative weighting of immediate versus future reward. As a lower bound, γ should be large enough that the effective optimisation horizon, $1/(1-\gamma)$, spans at least the wake transport time between T1 and T2, but ideally several multiples of this. Let us define the wake transport time between T1 and T2 which are five diameters apart as $T_w = 5D/U_\infty$ (non-dimensional time units, with $U_\infty = 1$), and let us also define a scaling factor k_γ that ensures the agent's effective optimisation focus extends beyond the time it takes for an action's wake effect to reach the downstream turbine. That is,

$$\frac{1}{1-\gamma} \gtrsim k_\gamma \frac{T_w}{dt}, \qquad k_\gamma > 1. \tag{2-20}$$

The specific values of γ and k_γ used in experiments are given in the Chapter 3.

2-3-8 Initial State Distribution — μ_0

The agent always begins from a fixed, known initial state s_0, so μ_0 is simply:

$$\mu_0(s) = \delta(s - s_0) \quad \text{i.e.} \quad \mu_0 = \delta_{s_0}. \tag{2-21}$$

2-3-9 Horizon — H

The horizon H uses the same wake-transport-time logic as γ, but it must be considerably longer, since the full wake interaction needs to propagate well past T2 *beyond* the effective optimisation window set by γ. Accordingly, a new scaling factor, k_H, is introduced to ensure this result:

$$H\,dt \gtrsim k_H\,T_w, \qquad k_H \gg k_\gamma. \tag{2-22}$$

A second, independent lower bound comes from the dynamic yaw trajectory itself: H must span at least one full oscillation period of the slowest trajectory tested. Otherwise, the effect of a test action's frequency is not easily determinable (this is to prepare for the fact that the optimal trajectories are likely periodic [10]. Given a minimum Strouhal number $St_{\min}$ (informed by the literature, and later refined based on grid search results), we set

$$H\,dt \gtrsim \frac{1}{St_{\min}}. \tag{2-23}$$

As with T_w above, this bound uses the normalised FVWM convention $D = 1$, $U_\infty = 1$, so that $St = fD/U_\infty$ reduces to $St = f$. The horizon is set to respect both these bounds, with specific values and Strouhal ranges detailed in the Chapters 3 and 5.

2-4 Block Diagram of the POMDP

The preceding sections are summarised in the block diagram shown in Figure 2-1.

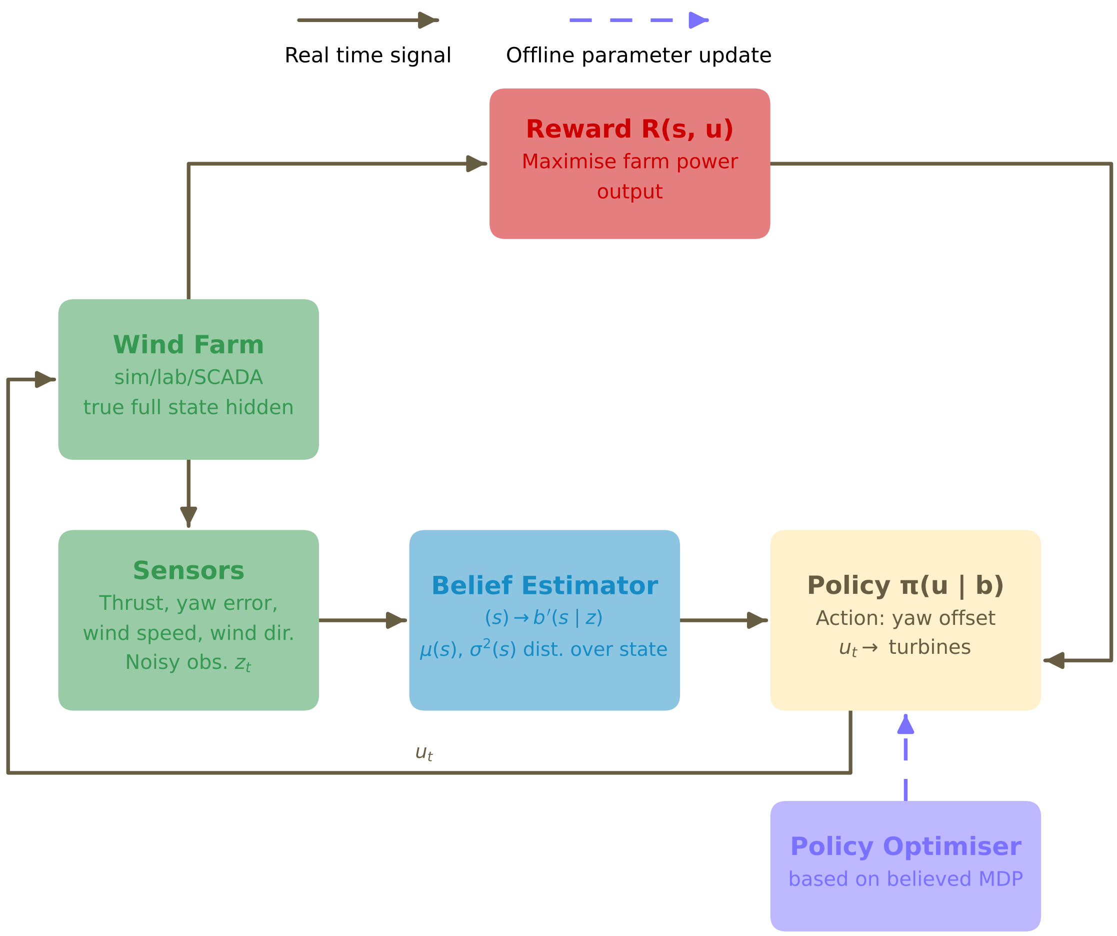


Figure 2-1: Block diagram of POMDP controller.

2-5 Policy Optimiser

The policy optimiser used in this formulation is Proximal Policy Optimisation (PPO), developed by Schulman et al. in 2017 [63]. Since the state space is continuous and high-dimensional and the transition dynamics are unknown to the controller, the Bellman optimality equations (4–9) cannot be solved exactly. Instead, a *direct RL* approach is taken: a parameterised policy is improved directly by gradient ascent on an estimate of the expected return, sampled from interaction with the environment (the FVWM, or the laboratory, in lieu of a real wind farm). The policy is represented by a neural network that maps observations to actions, and is refined through trial-and-error interaction.

PPO is chosen as the specific algorithm for four reasons. Firstly, it supports continuous action spaces, which suits the continuous yaw commands used here. Secondly, it learns a *stochastic* policy, which provides exploration during training. Thirdly, it improves training stability by limiting how much the policy is allowed to change at each update (the “clipped” objective), which avoids the destructive large updates that can occur with plain policy-gradient methods. Finally, it achieves this stability with a first-order method that is straightforward to implement. Together these make PPO a well-established default for continuous control problems of this kind. The remaining details are given in the following subsections.

2-5-1 Policy Gradient Methods

This work uses *direct* RL: rather than solving the Bellman optimality equations for a value function, the policy is represented explicitly as $\pi_\theta(a \mid o_t)$ — a neural network with parameters θ that map observations to actions. Its parameters are adjusted directly to maximise the expected discounted return. Note that, consistent with the POMDP formulation of Section 2-3-3, the policy is conditioned on the observation history o_t rather than the full state s; the results below are stated in terms of o_t, but are otherwise the standard policy-gradient results, which are conventionally written over states.

The objective to be maximised is

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \tag{2-24}$$

where τ denotes a trajectory generated by following π_θ. The parameters are updated by gradient ascent, $\theta \leftarrow \theta + \eta \nabla_\theta J(\theta)$, where the gradient is given by the policy gradient theorem:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid o_t)\, A^{\pi_\theta}(o_t, a_t)\right]. \tag{2-25}$$

Intuitively, this rule makes an action more likely when it performed better than the policy's average behaviour, and less likely when it performed worse. This comparison is captured by the *advantage function* $A^{\pi_\theta}(o, a) = Q^{\pi_\theta}(o, a) - V^{\pi_\theta}(o)$, which measures how much better action a is than average, in o. Estimating the advantage requires an estimate of the value function V^{π_θ}, which is learned separately by minimising the temporal-difference error $\delta_t = r_t + \gamma V(o_{t+1}) - V(o_t)$ against sampled experience. Methods that maintain both an explicit policy (the *actor*) and a learned value function (the *critic*) are termed *actor-critic* methods, and are the basis of PPO.

2-5-2 Proximal Policy Optimisation

Plain policy gradient is prone to instability: a single large update can move the policy far enough to collapse its performance, because a small change in θ can cause a large change in behaviour. Trust Region Policy Optimisation (TRPO) [64] prevents this by constraining each update so the new and old policies stay close, but this requires solving a constrained optimisation problem with expensive second-order methods at every step.

Proximal Policy Optimisation (PPO) [63] achieves a similar effect using only first-order methods. It is defined in terms of the probability ratio between the new and old policy for a sampled action,

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid o_t)}{\pi_{\theta_{\text{old}}}(a_t \mid o_t)}, \tag{2-26}$$

which measures how much more (or less) likely the updated policy makes that action. PPO maximises the *clipped surrogate objective*

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\,\hat{A}_t,\;\; \text{clip}(r_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\,\hat{A}_t\right)\right], \tag{2-27}$$

where $\hat{A}_t$ is an estimate of the advantage and ϵ is a small hyperparameter (typically 0.1–0.3). The clip limits the *ratio* to the range $[1-\epsilon, 1+\epsilon]$: once an update would move the new policy too far from the old one for a given action, the objective stops rewarding further movement, capping how much any single update can change the policy. Note that it is the probability ratio that is clipped — not the gradient, and not the reward.

The advantage estimate $\hat{A}_t$ is computed using Generalised Advantage Estimation (GAE) [65], a smoothed estimate that combines many future temporal-difference errors:

$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \qquad \delta_t = r_t + \gamma V(o_{t+1}) - V(o_t), \tag{2-28}$$

with $\lambda \in [0,1]$ trading off bias and variance. The full PPO objective trains the actor and critic together and adds an entropy bonus $S[\pi_\theta]$ that encourages continued exploration:

$$L^{\mathrm{CLIP+VF+S}}(\theta) = \mathbb{E}_t \left[L_t^{\mathrm{CLIP}}(\theta) - c_1 L_t^{\mathrm{VF}}(\theta) + c_2 \, S[\pi_\theta](o_t) \right], \tag{2-29}$$

where L_t^{VF} is the squared error between the critic's prediction and a target value, and c_1, c_2 weight the two additional terms [63]. The implementation used in this thesis follows the default continuous-control PPO configuration provided by StableBaselines3 [66]; the specific architecture and hyperparameters are described in Section 2-5-3 and 3-1-6.

2-5-3 Neural Network Details

Two separate neural networks are optimised during training:

1. The Critic Network: this is the value function; its weights are updated during training to minimise errors in its value (expected-return) predictions.

2. The Actor Network: this is the policy; its weights are updated to maximise the objective function introduced in Section 2-2.

Both the actor and critic are implemented as multi-layer perceptrons (MLPs) with two hidden layers of 64 units each, using tanh activation functions, as shown in Figure 2-2. The input to each network is the flattened observation vector defined in Section 2-3-3, of dimension history length$\times$ 6. The actor network outputs the mean of the (Gaussian) policy distribution over the yaw action; its standard deviation is a separate learned, state-independent parameter. The critic network outputs a scalar state-value estimate. The choice of two hidden layers of 64 units follows the standard default architecture used for continuous-control PPO in StableBaselines3 [66]. Tanh was selected over alternatives such as ReLU because it is zero-centred and bounded, which pairs naturally with the normalised, bounded observation and action spaces ($m_i \in [0,1]$, $\psi_i, y_i \in [-1,1]$) defined in Section 2-3-3.

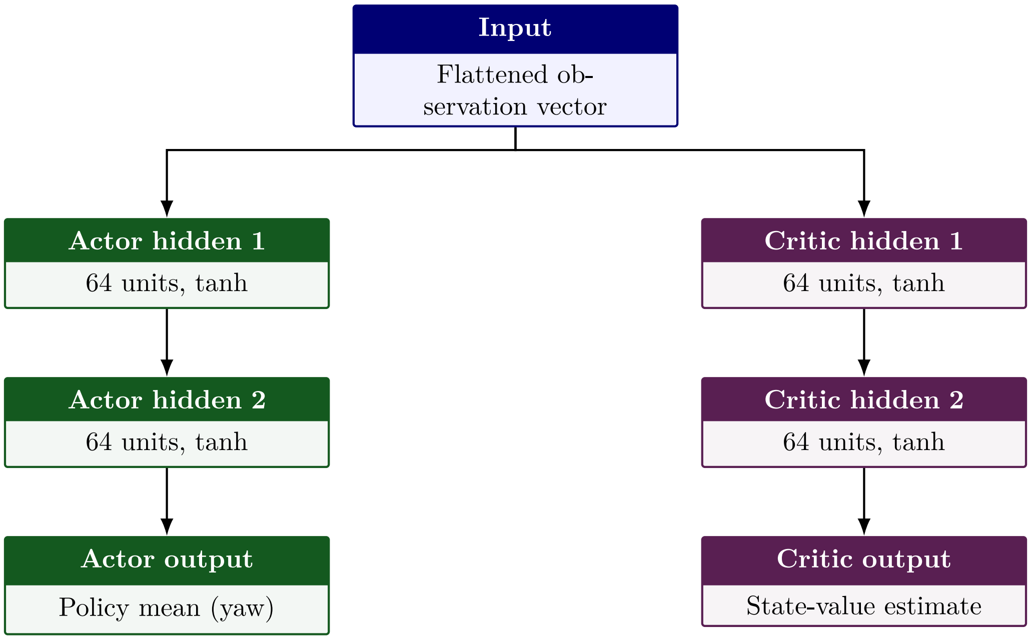


Figure 2-2: Actor-critic MLP architecture. The flattened observation vector is fed to two separate networks: the actor (policy), which outputs the mean of the Gaussian distribution over the yaw action, and the critic, which outputs a scalar state-value estimate. Each network has two hidden layers of 64 tanh units; no layers are shared.

2-6 Coding Implementation Description

This POMDP and RL controller training regime is implemented in Python using standard open-source reinforcement learning development tools. The environment (FVWM or laboratory) interface is defined using Gymnasium [67], and the RL algorithms for training and PPO are from the StableBaselines3 library [66].

2-7 Reinforcement Learning Controller Formulation Conclusion

This chapter laid the groundwork for subsequent chapters, especially the case studies in Chapters 3 and 5. All of the mathematical equations required to frame the turbine repositioning control problem as a POMDP were defined here: the transition, reward, observation, discount and horizon components, and the proximal policy optimiser used to solve it.

Chapter 3

Case Study 1: Simulation with the Free Vortex Wake Model

Reinforcement learning has recently been applied to collective wind farm control by several authors. Bizon Monroc et al. [68, 69, 70] and Kadoche et al. [71, 72, 73, 74] both formulate wind farm control as a multi-agent or graph-structured RL problem, learning yaw misalignment policies for collective wake steering. Similarly, Dong et al. [75] apply deep reinforcement learning to wake steering via yaw control, and Korb et al. [76] investigate RL for helix control. In each of these works, the available control action is restricted to those attainable by fixed-bottom turbines: yaw misalignment, axial induction and the helix. Turbine repositioning (possible only with the additional degree of freedom made available by a floating platform's mooring system) has instead been explored primarily through non-learning-based dynamic optimisation, notably by Starink et al. [10]. The FVWM developed by van den Broek et al. [4] serves as a fast control-oriented physics-based environment for training the RL controller. Using this environment, the reinforcement-learning approach used for fixed-bottom turbine collective control trained on steady-state wake models in the works above, is extended to the *repositioning* problem in a *dynamic wake* environment.

The following research questions are addressed in this chapter: 2, 3, 4 and 5. Additionally, the necessary foundations for contributions 3, 5 and 6 are also built here.

- **RQ 2** Is RL-based repositioning control feasible for online and/or offline deployment?
- **RQ 3** What wake-steering policies does the RL agent learn, and how do they compare against baseline strategies?
- **RQ 4** How much power can dynamic repositioning recover, compared to a static baseline?
- **RQ 5** What effect do configuration parameters have on RL controller performance, especially the length of aggregated historical state measurements as a proxy for the unobservable states?

Contribution 1 A POMDP formulation and RL controller for dynamic wake-steering repositioning, trained on the extended FVWM.

Contribution 2 Characterisation of the optimal repositioning signals discovered by the RL controller, benchmarked against literature strategies (Starink [10], van den Broek [4]) and standard grid search techniques.

Contribution 3 Quantification of the computational burden of training the RL controller in the FVWM.

An iterative process of conducting grid searches, running baseline cases, changing the FVWM configurations and updating RL training parameters occurred. The solution space was first characterised through baseline steady state cases and a grid search (Section 3-2), which identified the frequency band in which optimal periodic repositioning lies. This empirical finding set the RL controller's discount factor, episode horizon and the wake resolution required to represent a full forcing cycle (Section 3-1). The structure of this chapter is as follows: the simulator and controller parameters are presented first, followed by the baseline cases and grid search, finishing with the RL results and a compute-burden analysis. This flow is shown in Figure 3-1.

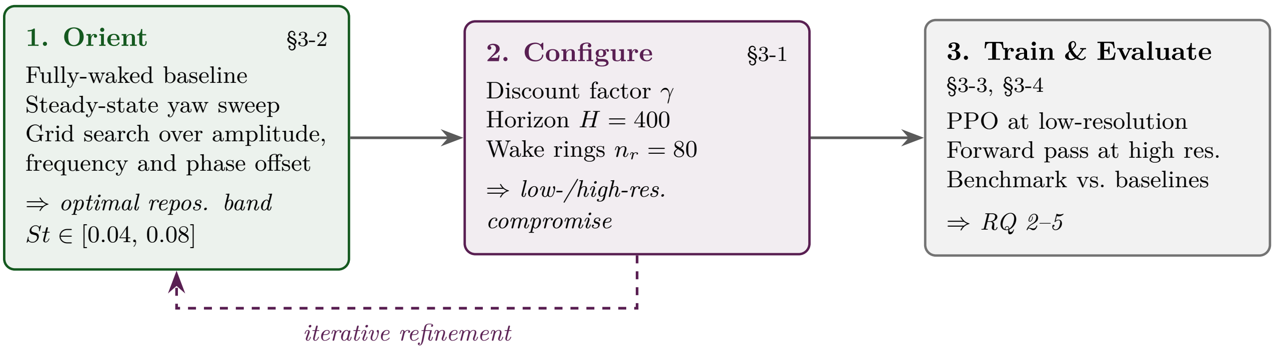


Figure 3-1: Logical structure of this chapter. The dashed arrow indicates the iterative process by which configuration changes prompted re-running of the baseline and grid search cases.

3-1 Simulator and RL Controller Training Set-up

The controller is structured according to the format outlined in Chapter 2. There are however, some parameters that are set specifically for this case study: the history length, discount factor, horizon, normalised performance parameter, and training hyperparameters. The choice of these variables is described in this section.

3-1-1 History Length

The length of the observation history provided to the controller must be sufficient for it to act as a proxy for a formal belief state. The RL results section (Section 3-3) is a sensitivity study of the influence of different history lengths, and therefore this parameter is not discussed further here.

3-1-2 Discount Factor — γ

We hypothesise that the optimal result will be periodic. Based on the grid search of sinusoidal trajectories (see Section 3-2), we expect the optimal periodic result to lie roughly in the range of

$St = 0.04$ to 0.08. We also know that the tightest constraint on the optimisation horizon is that the discount factor should allow the agent's optimisation focus to include at least one full period of the slowest candidate motion, i.e. $St_{\min} = 0.04$, corresponding to a period of $1/St_{\min} = 25$ non-dimensional time units.

Accordingly, the effective optimisation horizon is set to $1/(1-\gamma)\,dt \approx 30$ non-dimensional time units (100 steps), which comfortably spans one full period even at $St_{\min}$, while also giving $k_\gamma \approx 6$ from Eq. (2-20), safely exceeding the wake transport time T_w.

3-1-3 Horizon — H

The episode horizon H must allow the wake to propagate well past T2 *and* capture several periods of oscillation, rather than just one, so that the effect of a given trajectory's frequency on cumulative reward is observable.

Therefore, H is set $H\,dt = 120$ non-dimensional time units ($H = 400$ steps), which spans 5–10 full periods across the candidate range $St = 0.04$–0.08, giving $k_H \approx 24$ in Eq. (2-22) — well in excess of k_γ, as required — and satisfying $H\,dt \gtrsim 1/St_{\min}$ in Eq. (2-23) several times over.

3-1-4 Normalised Performance Parameter: ($m = P$ in Case Study 1).

In this case study, the normalised performance parameter m which was first introduced in Section 2-3-3, is realised as turbine power, P. However, since this is a simulation and not a real wind farm, turbine power is not directly measured and is instead computed from momentum theory. For the upstream turbine, the reference power is

$$P_{0,\mathrm{ref}} = c'_p(a_0)\, A_r\, [U_\infty(1-a_0)]^3\,, \tag{3-1}$$

while for the downstream turbine it is

$$P_{1,\mathrm{ref}} = c'_p(a_1)\, A_r\, [U_\infty(1-2a_0)(1-a_1)]^3\,, \tag{3-2}$$

where $c'_p(a)$ is the power coefficient as a function of the induction factor, A_r is the rotor swept area, U_∞ is the free-stream velocity, and a is the axial induction factor. This computed power is normalised by the maximum theoretical power achievable via momentum theory, denoted the reference power,

$$m = \bar{P} = \frac{P}{P_{\mathrm{ref}}}. \tag{3-3}$$

Accordingly, we arrive at a normalised and non-dimensional power metric that describes how much power each turbine is generating as a fraction of the theoretical optimum.

3-1-5 FVWM Configuration Parameters

Both a larger number of steps and a larger number of vortex rings is required for the upcoming RL tests, as compared to those in van den Broek's [4] static optimum validation studies with economic MPC. This has a negligible effect on the resultant powers as discussed in Section 3-2, however it significantly increases the run-times. Nevertheless, this increase to the number of steps and rings is required. The reasoning for the larger number of steps was already detailed in Sections 3-1-2 and 3-1-3. The reasoning for the larger number of rings follows a similar argument: to learn a dynamic trajectory, the simulation must retain a sufficient number of wake rings in memory to reach past the downstream turbine. Following the ring position-update in Eq. (5) of [4], and approximating mean wake convection speed as $(1-a)u_\infty$ per actuator-disc momentum theory, the forcing wavelength is estimated as follows for the likely optimal forcing frequency range $St \in [0.04, 0.08]$:

$$St = 0.04: \quad T = \frac{1}{0.04} = 25.0, \qquad \lambda \approx (1-a)\,T \approx 0.67 \times 25.0 \approx 16.75\ D \tag{3-4}$$

$$St = 0.08: \quad T = \frac{1}{0.08} = 12.5, \qquad \lambda \approx (1-a)\,T \approx 0.67 \times 12.5 \approx 8.4\ D \tag{3-5}$$

Accordingly, the streamwise wake coverage $x_{\text{cover}} \approx (1-a)\,n_r\,h_{nd}$ calculation is described here for the chosen case of 80 rings: for $n_r = 80$ and a non-dimensional timestep of 0.3: $x_{\text{cover}} \approx 16.1\ D$. As desired, this exceeds the wavelength at the high-frequency end ($St = 0.08$, $\lambda \approx 8.4$ D) with substantial margin, but unfortunately, it falls marginally short of a full wavelength at the low-frequency, worst-case end ($St = 0.04$, $\lambda \approx 16.75$ D). In the interest of not further increasing the run time, higher ring numbers were not considered, and this option was chosen despite its slight inadequacy for St=0.08.

The RL training was conducted using a lower-resolution environment with a reduced amount of vortex rings due to the computational constraints of the DelftBlue high performance computer (HPC) [77], which only allows 24-hour compute slots. Higher-resolution training, which would require two to nine days, was not feasible within this framework. (Further detailed analysis of computational costs/complexity is provided in Section 3-4.) A compromise was reached by running grid searches at high-resolution, RL training at low-resolution, and then re-running forward passes of the low-resolution-trained RL controllers back inside the high-resolution FVWM to obtain fair comparisons of the achievable power gains.

Table 3-1: Free-Vortex Wake Model (FVWM) configurations used for the low- and high-resolution simulations, alongside the reference static-optimum configuration of van den Broek et al. [4]. All configurations share identical temporal and spatial domain settings, differing only in wake discretisation resolution.

Parameter	**Low res.**	**High res.**	**Config. from [4]**
Dimension	3	3	3
Number of steps	400	400	400
Time step (-)	0.3	0.3	0.3
Number of wake rings	30	80	40
Blade elements per ring	8	16	16
Number of turbines	1	1	1
Number of virtual turbines	1	1	1
Upstream turbine position	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
'Virtual' turbine position	$(5.0, 0, 0)$	$(5.0, 0, 0)$	$(5.0, 0, 0)$
Vortex core size (-)	0.16	0.16	0.16
Approx. avg. per-run time [s]	8	196	60
Approx. avg. RL train time	9-13 hrs	9 days	2 days
Approx. avg. x_{cover}	6D	16D	8D

3-1-6 RL Training Regime Hyper-parameters

The PPO agent was trained using the hyper-parameters summarised in Table 3-2. These are the Stable-Baselines3 defaults for continuous-control tasks. A rollout length of $n_{\text{steps}} = 2048$ was chosen to span roughly five full episodes (at $n_{\text{FVWMsteps}} = 400$) before each policy update.

This means that gradient estimates are averaged over multiple independent wake-development trajectories rather than a single noisy episode. The learning rate (3×10^{-4}) and minibatch size (64) are standard PPO defaults balancing update stability against training speed. Reusing each rollout buffer for $n_{\text{epochs}} = 10$ passes improves sample efficiency, since generating new simulator rollouts is expensive; this is facilitated by PPO's clipped surrogate objective, which makes stable reuse of the same batch across multiple epochs possible [63].

Table 3-2: PPO hyperparameters used for all history-length configurations.

Hyperparameter	Value
Training timesteps	1 million
Rollout length (n_{steps})	2048
Minibatch size	64
Learning rate	3×10^{-4}
Epochs/update (n_{epochs})	10
Clip range	0.2

3-2 Orientation in the Solution Space

The chosen FVWM parameters detailed in the previous Section 3-1 differ from those that produced optimal results for van den Broek's studies [4]. Therefore, in this section, the steady state optima for this specific FVWM configuration are re-derived, to allow for fair comparisons with the RL controller. Following this rederivation of steady state optima, the results of a grid search of periodic repositioning signals is presented in order to orient the reader in the solution-space in preparation for the RL results sections of this chapter.

3-2-1 Fully Waked Comparison Case

The power results for the fully-waked case are shown in Figure 3-2. This case shows that without some form of wake steering, when turbines are aligned in the direction of the oncoming wind with the typical 5D spacing, the downstream turbine produces almost no power.

3-2-2 Static Optimum

Van den Broek et al. [4] conducted a parameter sweep to determine the optimal upstream static yaw angle (with the configuration in the third column of Table 3-1): their optima were at ± 34 degrees and resulted in a two-turbine array power of 0.313. Their sweep also showed that power capture is worst in the fully-waked zero yaw case (as expected), and that the power gains rise from this point out to the maxima at ± 34 degrees and then drop off again.

The FVWM is symmetric about the yaw axis, therefore for this study, only positive yaw angles are shown; results at negative angles are identical. A parameter sweep was run with the high-resolution configuration from Table 3-1 from 25 to 40 degrees yaw angle. The results of this parameter sweep are provided in Figure 3-3. The optimum remains the same as that found in [4]. This makes sense because no additional gains are expected due to an increased number of vortex rings since the wake is properly developed. This resultant total array power of 0.313 is the result against which the periodic results to come can be judged.

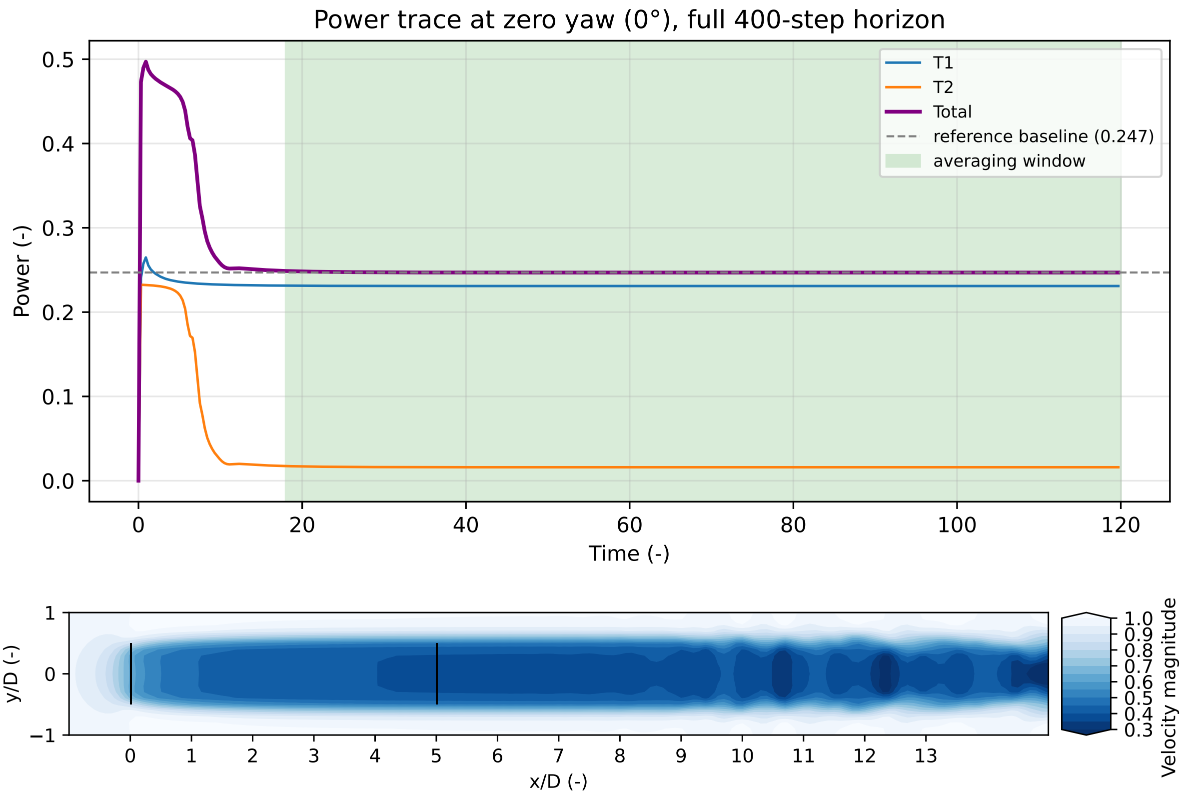


Figure 3-2: Fully waked case: power output over time. Turbine 1 produces optimal power, but Turbine 2 produces almost no power. The wind velocity deficit caused by the wake is shown at the bottom of the image for context, where the black lines at x/D = 0 and 5 represent the turbines.

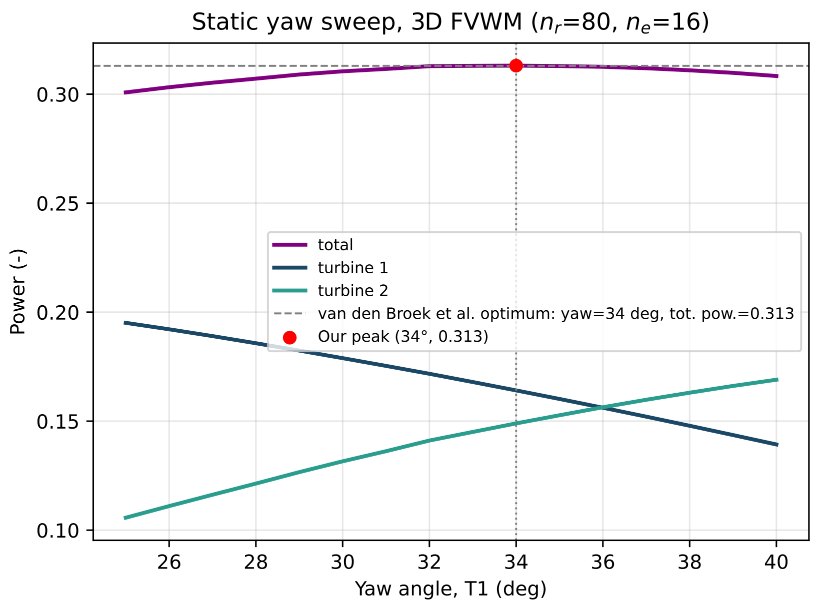


Figure 3-3: Parameter sweep of static yaw angles to find the optimum (the optimum is the same as that found with the slightly lower resolution FVWM configuration by [4]).

The fixed-bottom static yaw case is adopted as a comparison baseline because its floating-platform counterpart is inherently self-limiting: for a floating turbine, the same yaw angle that redirects the wake also displaces the platform laterally, but in the opposite sense to that redirection. A static yaw large enough to steer the wake clear of the downstream turbine therefore also translates the upstream turbine, pulling the redirected wake back into the downstream turbine's path. The downstream turbine must therefore also reposition. As such, we arrive at continual coordinated movements of T1 and T2 to allow T2 to escape T1's wake.

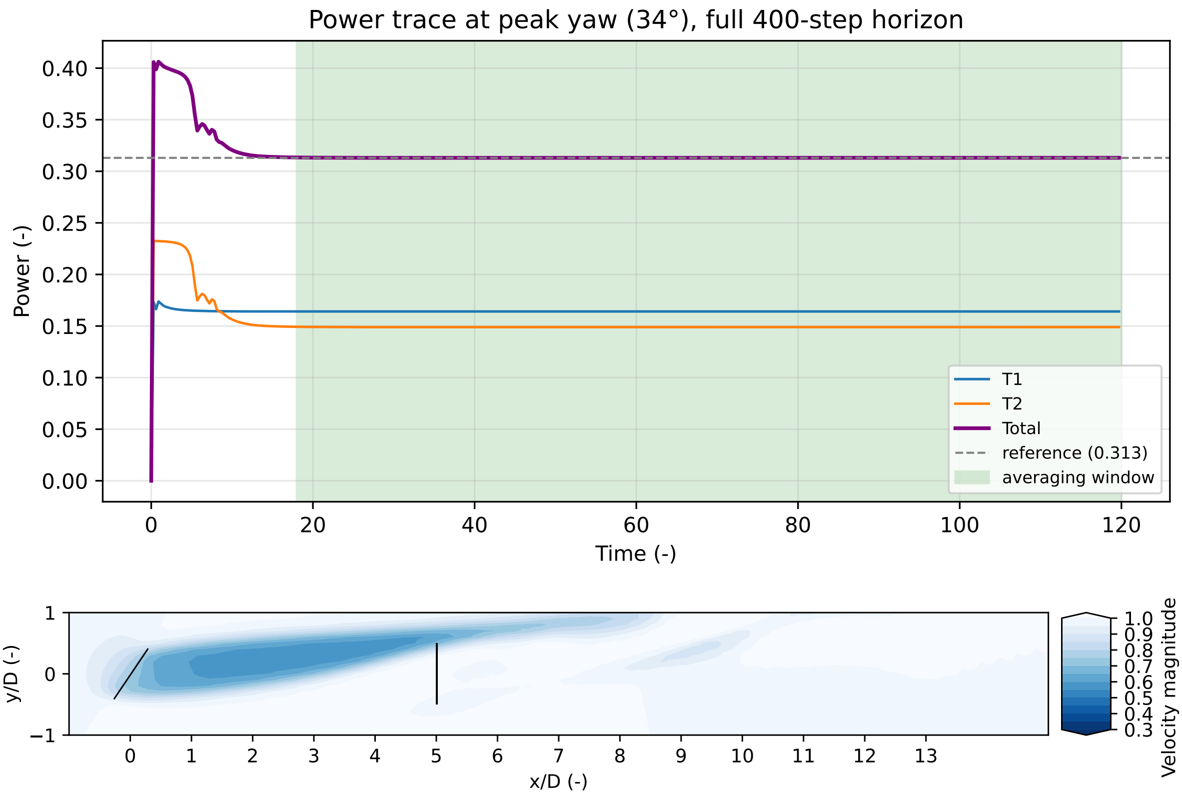


Figure 3-4: Static optimum: power output over non-dimensional time. Turbine 1 sacrifices its optimal power, so that Turbine 2 can produce more power than in its fully-waked case. This results in higher total array power. The wind velocity deficit caused by the wake is shown below for context, where the black lines at x/D=0 and 5 representing the turbines. There is significantly higher wind speed (white region) in front of Turbine 2 than there was for the fully-waked case shown in Figure 3-2.

3-2-3 Grid Search

Several parameters of the RL controller need to be carefully chosen to arrive at a viable solution. This section provides the results of a grid search with the intention that this enables us to determine likely ranges for optimal periodic results (so that informed decisions can be made about controller optimisation horizons before computationally expensive RL training commences). It also acts as a baseline against which to compare RL controller success.

The grid search was performed over the following ranges of amplitudes A, frequencies f and phase offsets between turbines ψ:

$$A = \{25,\ 27.5,\ 30,\ 32.5,\ 35\},$$
$$f \in [0.1,\ 5]\ , St_{\text{forcing}} = [0.00347,\ 0.1733], \qquad \text{in steps of } \Delta f = 0.245\, St_{\text{forcing}} = 0.00849,$$
$$\psi \in \left[-\frac{\pi}{2},\ \frac{\pi}{2}\right], \qquad \text{in steps of } \Delta\psi = \frac{\pi}{8}.$$

Both the front and rear turbines were given the same amplitude and frequency in each test. The St_{forcing} was defined by using one of the optimal wake-repositioning frequencies reported by Starink et al. ($f_{\text{ref}} = 0.002273\text{Hz}$) for a rotor radius of 775m. The corresponding Strouhal number was computed using their turbine diameter ($D = 125\text{m}$) and inflow velocity ($U_\infty = 8.2,\text{m/s}$), using the formula $\text{St}_{\text{forcing}} = \frac{f_{\text{ref}} D}{U_\infty}$.

A contour plot was developed for each amplitude in the grid search range, showing the relationship between phase offset, forcing frequency, and the non-dimensional and normalised sum of the two turbines' powers. There are many optima all resulting in an array power of 0.351. This is a 42% improvement over the fully-waked baseline and 12% over the static optimum. In the interest of succinctness, only one such contour plot for the 32.5 deg amplitude case is shown in Figure 3-5, and a second plot is provided to show the variations in sinusoids capable of achieving this optimal power range in Figure 3-6.

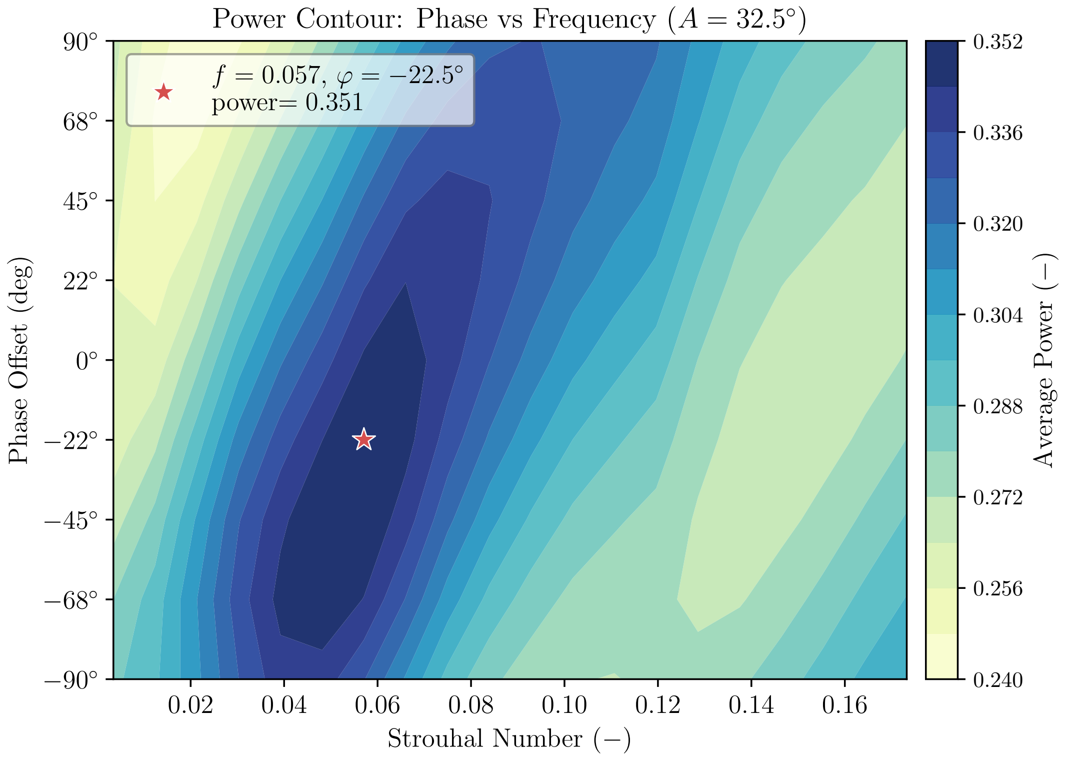


Figure 3-5: Power contour for the 32.5 deg amplitude case, run in the high-resolution FVWM configuration detailed in Table 3-1.

3-3 The Optimal RL Results

Armed with these comparison points, the RL results are now presented in this section. A range of different yaw angle amplitudes, frequencies and phase offsets are optimal. This section includes some statistical analyses of the typical variations in optimal result signal properties. However, all optimal signals are still similar in terms of the shape of the signal: an almost sinusoid with both the front and rear turbines operating at the same forcing frequency. An example of this signal shape can be seen in Figure 3-8d. The statistical analyses were carried out on a sample set of ten different optimal RL controllers for each of the following history lengths: 10, 20, 40, 60 and 100 time steps. This was in order to allow for a sensitivity analysis on the effect of history length on the learned policy. Since the state history of yaw angle, turbine position, and turbine power together act as a proxy for the unobservable states, it is essential to know what history length is required to make reasonable predictions of unobservable states. The compute burden is high for each controller (the exact compute time is discussed in Section

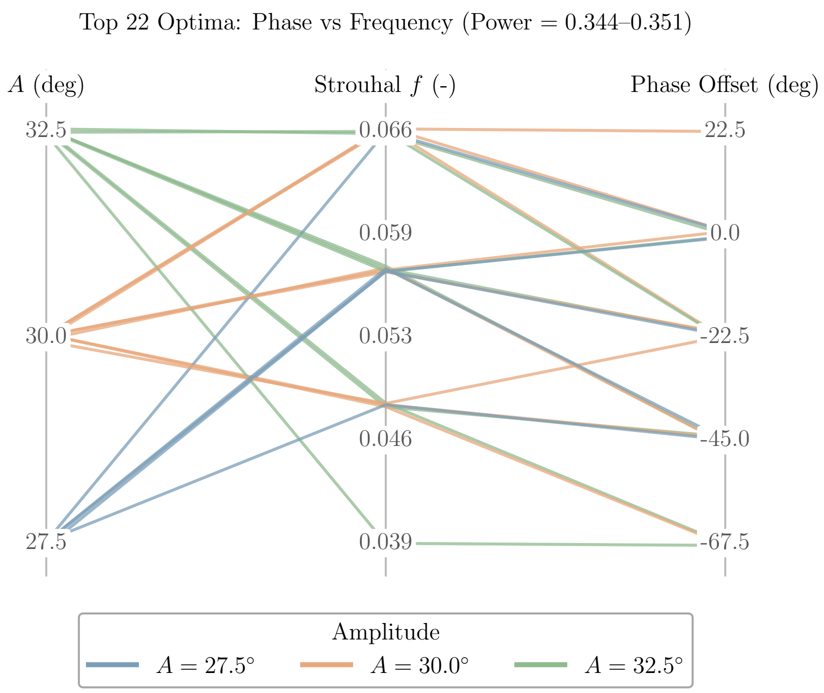


Figure 3-6: The variety in the top 22 optimal frequencies, amplitudes and phase offsets from the grid search.

3-4), therefore the analysis was restricted to ten repeat trainings at each history length. Of course, with more compute power, a higher level of confidence in the optimal ranges could be achieved. It is important to note that these results are heavily influenced by the chosen mooring line slackness as this controls what position change is possible via YITuR. The range of optimal signal properties would be different for different mooring line configurations as described in [10].

3-3-1 Evolution of the Training Regime

The aim of this section is to show how the RL controller moves from learning steady-state solutions to truly periodic solutions as training progresses. Initially, training was done on a laptop for only 2e5 steps, but convergence could not be achieved and only near steady state, or small amplitude oscillations that underperformed compared to the grid search resulted. Therefore, training was instead run on the DelftBlue [77] supercomputer for one million time steps. This was shown to be sufficient training time to reach convergence, as can be seen by the plateau in reward increase in Figure 3-7. This figure contains the reward curve for an optimal controller and four instances along the training journey are highlighted with red circles. Figure 3-8 then shows the actual "learned" trajectories of yaw and position at these four points, along with their associated power production. In fact, all trajectories in the converged optimal reward region (the plateau in Figure 3-7) are periodic with roughly the same appearance. Of course, there is still some variation in the range of optimal solutions, and this will be discussed in future sections.

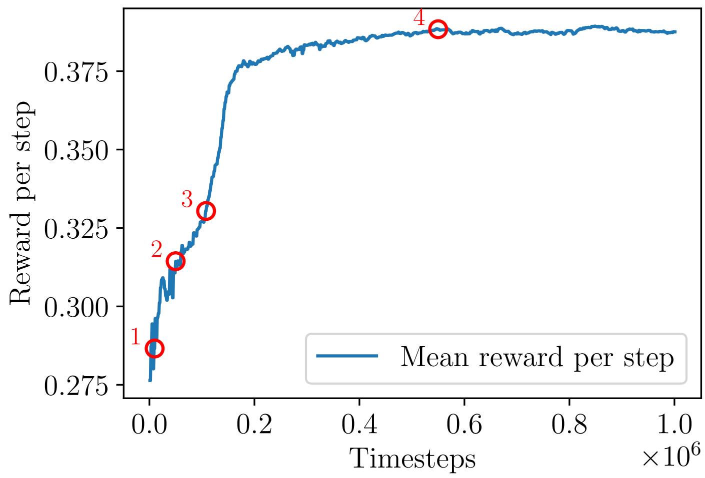
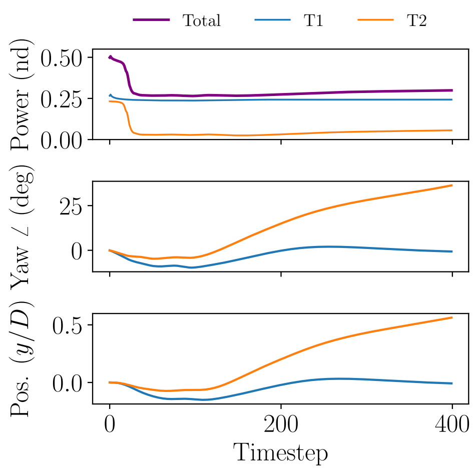
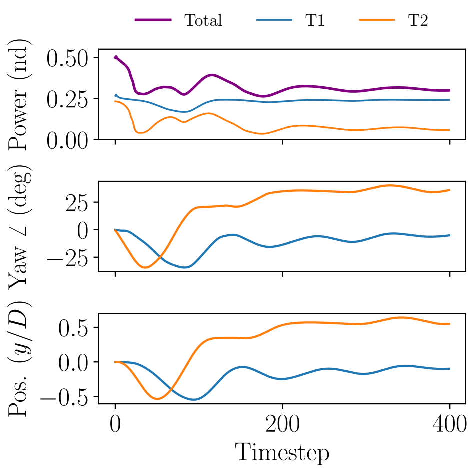


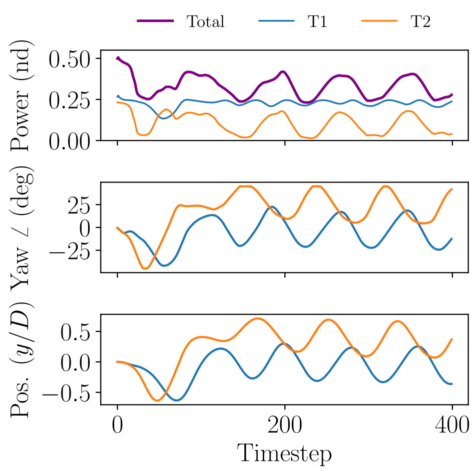
Figure 3-7: Reward increasing until a plateau is reached (optimal solutions exist in this plateau). Note that the magnitude of the reward is not directly comparable to the power gain calculations elsewhere in this chapter because for this figure, these values are directly from the low-resolution training runs.



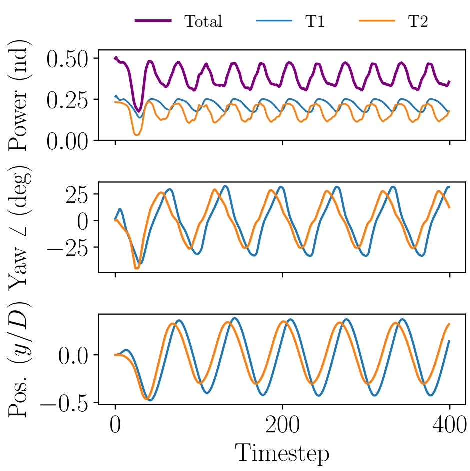
(1) Early training.



(2) Oscillations emerge.



(3) Oscillations strengthen.



(4) Converged periodic motion.

Figure 3-8: Yaw trajectories at the four training checkpoints marked in Figure 3-7.

3-3-2 Headline RL Results

Figure 3-9 shows the results from training ten different controllers for five different history lengths ranging from ten time steps up to 100. It is evident from Figure 3-9 that none of the history lengths offer a drastically different performance output. They all result in power outputs between 0.342 and 0.353, which are comparable to the optimal grid search sinusoidal cases, and significantly higher than the fully waked zero-yaw case (38-43% improvement) and the static optimum from van den Broek [4] (9-13% improvement). This is promising as it means that the RL controller independently and consistently learns *dynamic* optimal policies, strengthening our confidence in the hypothesis from the grid search that dynamic trajectories are optimal (for this mooring line configuration). Figure 3-10 compares all the RL controllers to the top-10-performing grid search results. The grid search spanned combinations of amplitude A, frequency f, and phase offset ψ, where in each test T1 and T2 shared their A and f but were allowed to differ in ψ. The RL agent operated under no such parametric assumptions, yet still converged on similar ranges of optimal A, f, and ψ — differing notably in its preference for solutions where the two turbines operate at different amplitudes of motion. The following subsections dive deeper into the intricacies of the relationships between frequency, amplitude, phase offset and power capture.

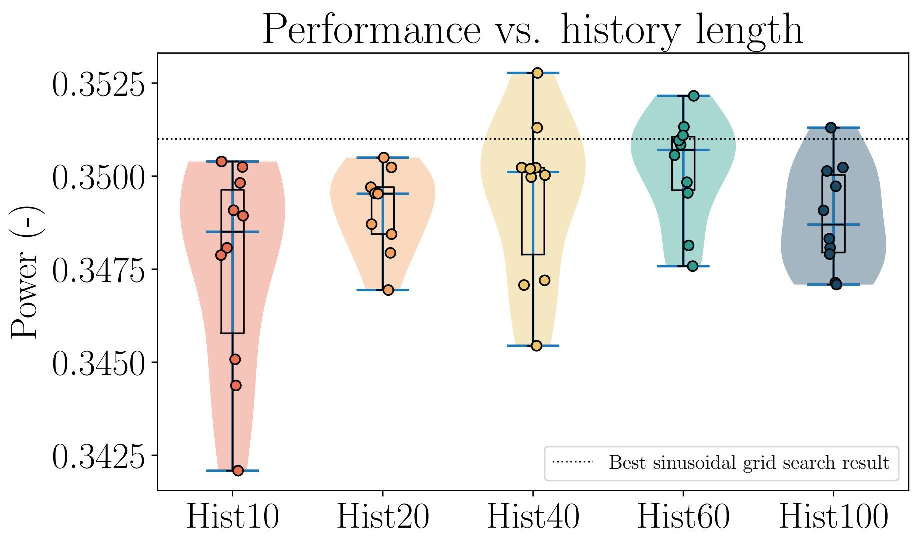


Figure 3-9: Violin box-plot of history length sensitivity analysis: the shaded region is widest where the most results are concentrated. This is a visual aid to the standard box plot. For context, the zero-yaw baseline array power is 0.247 and the static T1 yaw-only optimum from [4] is 0.313.

3-3-3 Optimal Amplitude

The optimal yaw and position periodic signal amplitudes lie in the range of 26 to 36 degrees and 0.28 to 0.44 (non-dimensionalised by rotor diameter), respectively. This result can be seen in Figures 3-11 and 3-12. From these figures, it is also evident that the history length influences the learned policy, with shorter history lengths appearing to favour a combination of smaller upstream turbine oscillation amplitude and larger downstream turbine amplitude.

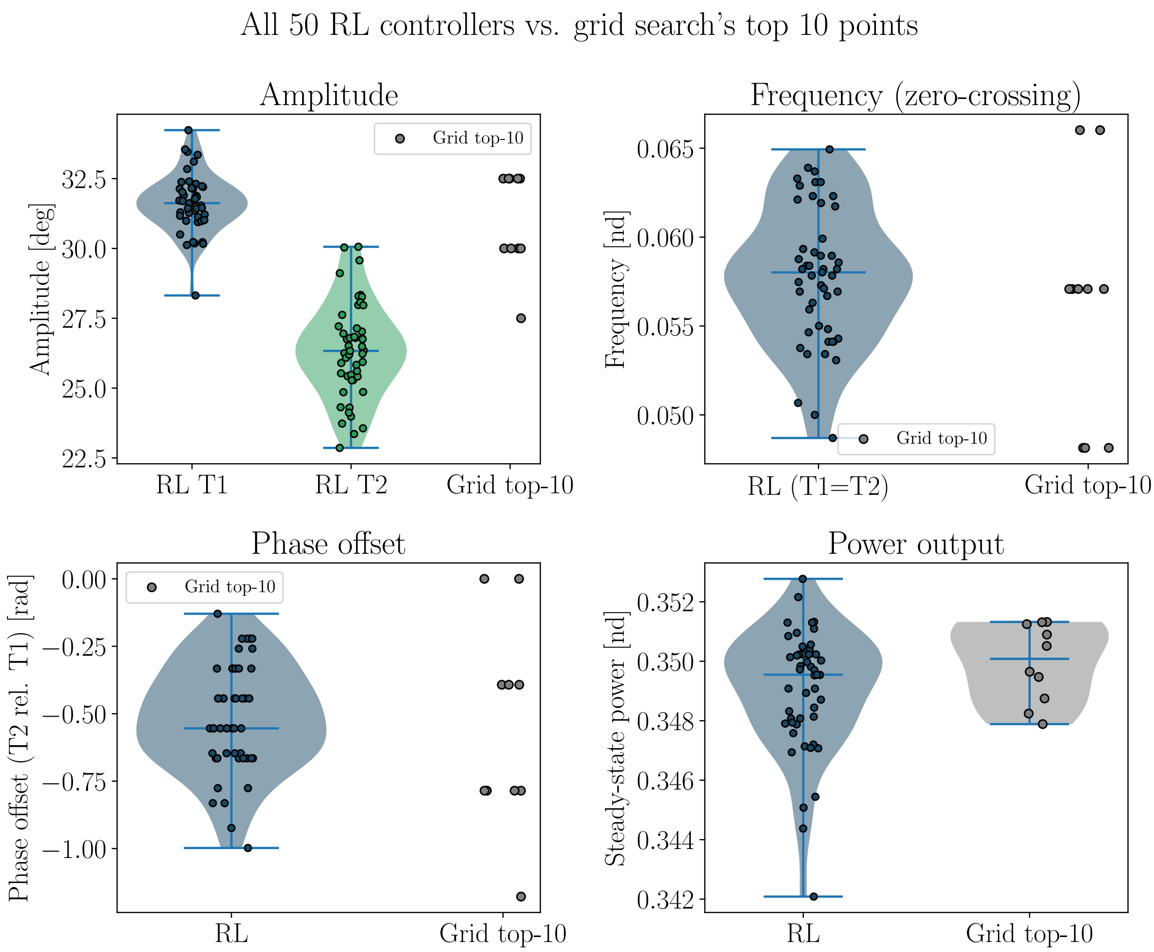
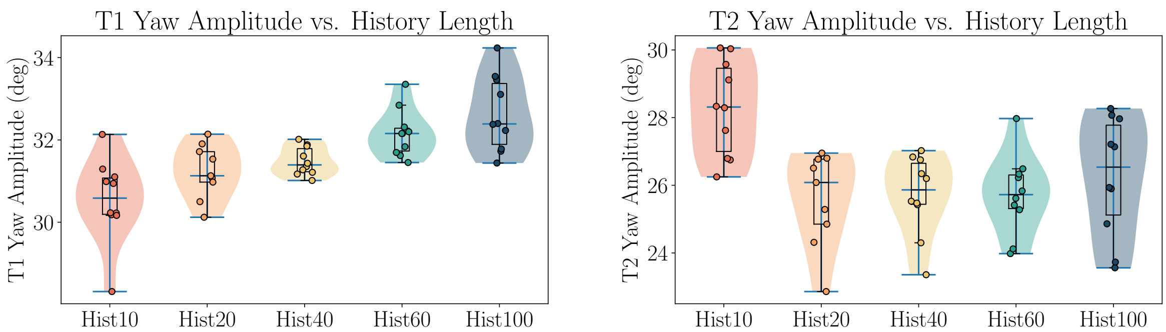


Figure 3-10: Highlighting the fact that the optimal RL policies lie in a similar frequency, amplitude and phase offset range to the grid search optima, but, RL optimal policies prefer slightly differing trajectories for T1 versus T2. The grid search constrains T1 and T2 to follow the same sinusoid.



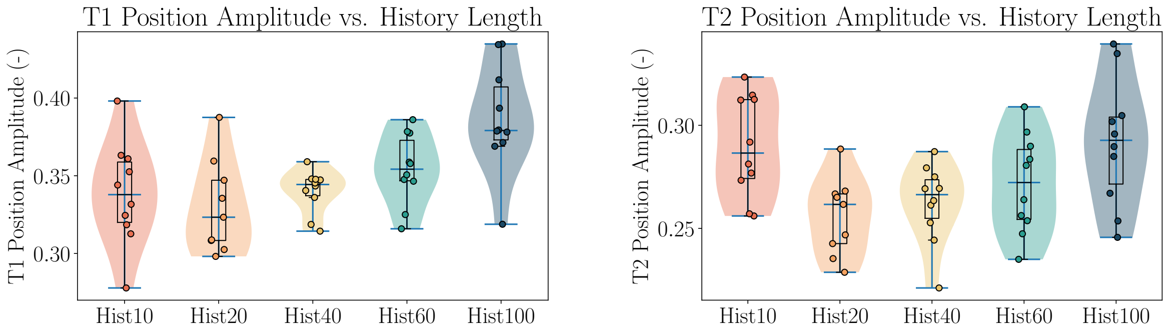
(a) Yaw angle T1.

(b) Yaw angle T2.

Figure 3-11: Optimal yaw angles across the evaluated history lengths.

3-3-4 Optimal Frequency

Continuing in the same vein, the optimal frequencies were computed via zero crossing. (The signal sampling rate was too course to enable good resolution with FFT methods.) In fact, the optimal frequencies lie in the range of 0.048 to 0.065 (Strouhal number), which is perfectly

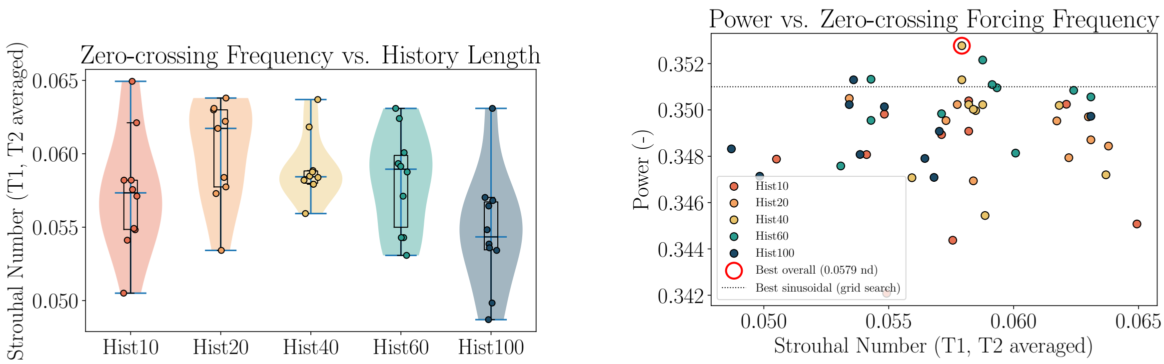


(a) Turbine repositioning T1.

(b) Turbine repositioning T2.

Figure 3-12: Optimal turbine repositioning amplitudes across the evaluated history lengths.

in alignment with the grid search result from Section 3-2. This is interesting because the best forcing frequencies are close to the platform natural frequency calculated back in Table 1-3, $St = 0.0597$. Due to the second order low pass filter damping motions above the platform natural frequency, it is impossible for the platform to oscillate much faster than this point. This result can be seen in Figure 3-13a. It is also interesting because this range differs from the typical Strouhal numbers expected for dynamic yaw, induction and helix wake mixing which are in the order of Strouhal number 0.2 [78, 24]. The coupling between platform natural frequency and turbine repositioning is a dominant factor here, whilst it is not for those other methods. To highlight the fact that all RL solutions result in similar power outputs, Figure 3-13b is also included; in this figure one can see that across this range of frequencies, all the resultant power outputs are between 0.342 and 0.353.



(a) Optimal forcing frequency (Strouhal number) violin box plots.

(b) Range of forcing frequencies achieved and the corresponding array power output.

Figure 3-13: Optimal forcing frequencies across history lengths and their relationship with power output.

3-3-5 Optimal Phase Offset

Regarding the phase offsets, all optimal results lie in the range of minus ten to fifty degrees, where a negative degree value corresponds to the front turbine lagging behind the rear. Only the yaw angle result has been provided for this, because the position signal phase is directly proportional (with some delay) to the yaw signal's. Essentially, what we see is that the rear turbine needs to move ahead of the front turbine by some tens of degrees in order to best escape the wake of the front turbine. As expected due to the delay between a yaw angle change and the associated turbine repositioning, the range of yaw angle phase offsets is larger than their corresponding repositioning phase offsets. This is displayed in Figures 3-14, 3-15a and 3-15b.

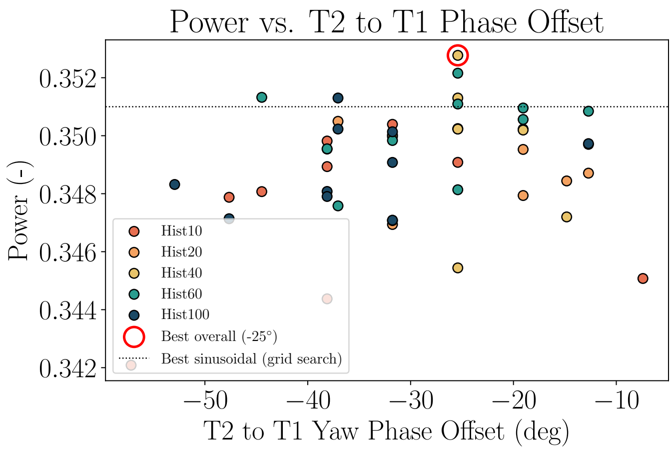
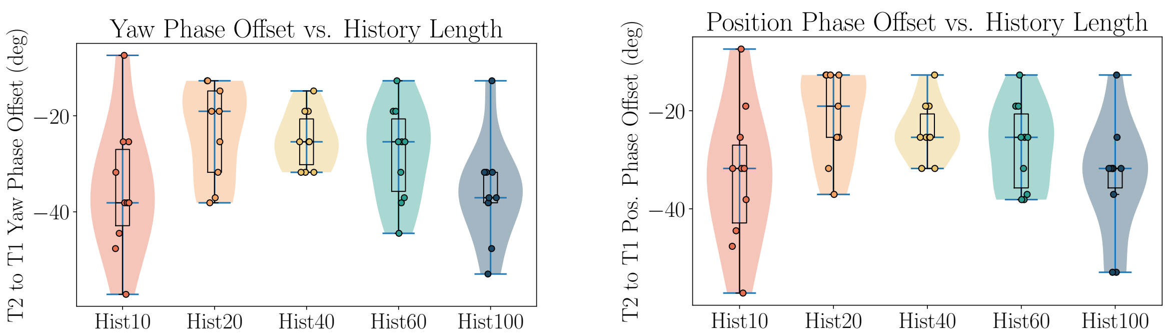


Figure 3-14: Relationship between yaw angle phase offset and power output.



(a) Optimal yaw angle phase offset.

(b) Optimal repositioning phase offset.

Figure 3-15: Optimal phase offset box plots obtained for yaw angle and turbine repositioning across the evaluated history lengths.

3-4 Training, Computation and Rollout Costs

3-4-1 RL Training in Low-resolution FVWM

A single wall-clock timer was placed around the entire training call (1,000,000 PPO timesteps, including checkpointing and periodic evaluation rollouts):

$$C_{\text{train}} = \frac{t_{\text{end}} - t_{\text{start}}}{3600} \quad [\text{h}] \tag{3-6}$$

This is the achieved cost of the full training time, not an isolated measurement of the simulator or the policy network alone. As introduced in Section 3-1-5, the RL controller was trained in the low-resolution FVWM configuration. For this configuration, training took between nine and 13 hrs per controller, with training time rising sharply for the history length of 100. The case with a history length of 60 had the lowest mean. These results are shown in Figure 3-16. They resolve to a step cost of $t_{\text{step}}^{\text{lo-res}} = C_{\text{train}}^{\text{lo-res}} / N_{\text{steps}}^{\text{lo-res}} = 0.03930$ s/step for the low-resolution FVWM configuration.

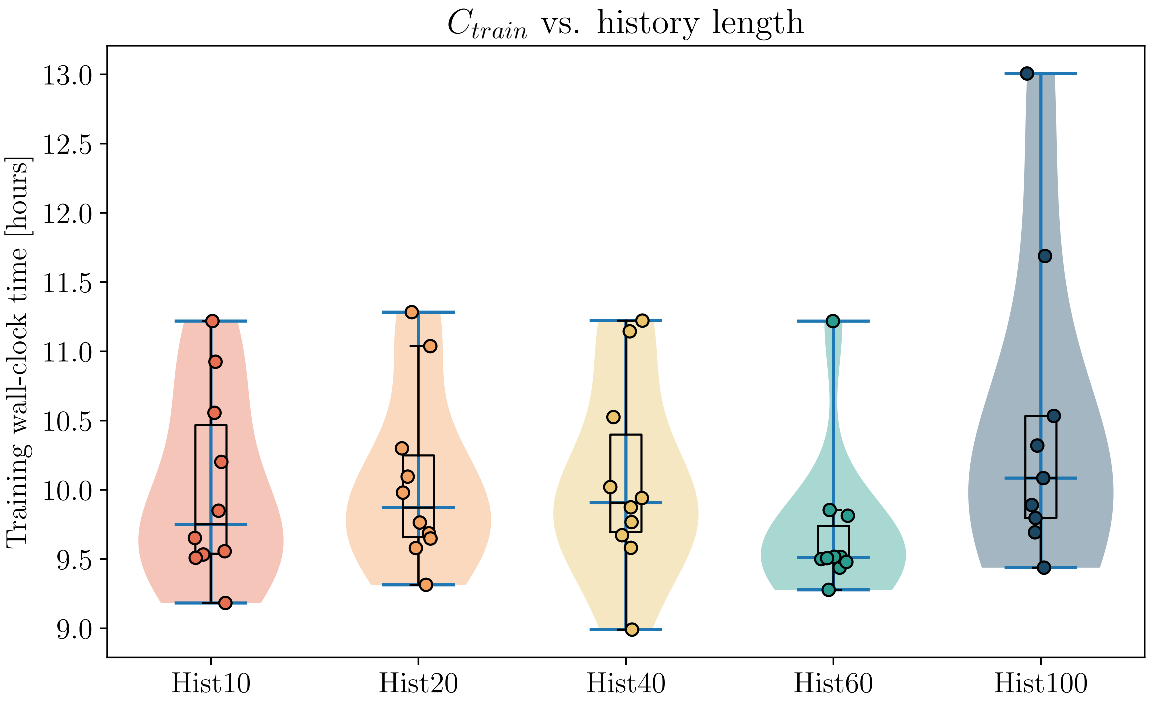


Figure 3-16: C_{train} across the models.

3-4-2 High-resolution FVWM Training Time Extrapolation

A full high-resolution FVWM configuration run was infeasible within the allocation constraints of the DelftBlue supercomputer [77], so instead, a truncated run of 2×10^4 steps was executed using the identical training pipeline. This showed that $t_{\text{step}}^{\text{hi-res}} = 0.82122$ s/step which is a $20.9\times$ slowdown on the low-resolution FVWM configuration.

The measured slowdown is compared against the theoretical $O(N^2)$ cost of the discretisation refinement. From the paper by van den Broek et al. [4], the number of states in the 3-D FVWM is:

$$n_p = n_e + 1 \quad \text{(3D)}, \qquad n_s = 2\, n_r\, n_p\, n_d + n_r\, n_e + n_t\, n_c, \tag{3-7}$$

where, n_e is the number of vortex elements per ring, n_r the number of vortex rings retained in the wake, $n_p = n_e + 1$ the number of points defining each ring in 3-D, n_d the spatial dimension of the simulation (3 in this work), n_t the number of turbines modelled, and n_c the number of control parameters per turbine; n_s is the resulting total size of the FVWM state vector.

This means that the low and high-resolution FVWM configurations have the following numbers of states displayed in Table 3-3.

The per-step cost is expected to scale as $O(N^2)$, with $N \approx n_r n_e$, because the induced-velocity equation from [4],

$$u_{\text{ind}}(x, q) = \sum_{b=0}^{n_r - 1} \sum_{j=1}^{n_e} u_i^{(b)}{}_j, \tag{3-8}$$

performs a summation at every one of the N wake points over every one of the N source elements. This means the ratio of the number of states between the high and low-resolution configurations is $n_{\text{high}}/n_{\text{low}} = 1280/240 = 5.33$. By this logic, the theoretical slowdown in using

Table 3-3: FVWM (3D) state-vector size, low- vs. high-resolution configuration.

Parameter	Low-resolution	High-resolution
n_e	8	16
n_r	30	80
$n_p = n_e + 1$	9	17
n_d	3	3
n_t	2	2
n_c	10	10
$2\,n_r n_p n_d$	1,620	8,160
$n_r n_e$	240	1,280
$n_t n_c$	20	20
n_s	**1,880**	**9,460**

the high-resolution configuration is $5.33^2 \approx 28.4\times$. This is slightly higher than the measured slowdown because the assumption that $N \approx n_r n_e$ does not hold for smaller numbers of elements and rings: then the other terms in Equation 3-7 have more of an effect.

Extrapolating the best-case measured high-resolution rate to a full 1-million-step run gives an estimated 228 hours (9.5 days) per run. This exceeds the 24 hr per job limit on the DelftBlue supercomputer [77], and is the reason the training was conducted at low-resolution. To reconcile the quoted power gains, the trained controllers were then run in the high-resolution configuration of the FVWM in order to quote comparable power gains with the grid search, static and fully-waked comparison cases.

3-4-3 Estimation of Training Time in Experimental and Real-World Scenarios

Having established that high-fidelity simulation is computationally prohibitive, this section investigates whether training could instead be faster on a scaled laboratory rig or on the full-scale platform itself. To keep the dynamics comparable across simulation, laboratory, and full-scale testing, the Strouhal number must be preserved:

$$\mathrm{St} = \frac{fD}{U} = \text{constant}. \tag{3-9}$$

We define scale ratios λ_L, λ_U, and λ_t for rotor diameter, inflow velocity, and time, respectively, so that λ_t converts one second of laboratory operation into its full-scale equivalent:

$$\lambda_L = \frac{D_{\text{lab}}}{D_{\text{full}}} \tag{3-10}$$

$$\lambda_U = \frac{U_{\text{lab}}}{U_{\text{full}}} \tag{3-11}$$

$$\lambda_t = \frac{\lambda_L}{\lambda_U} = \frac{f_{\text{full}}}{f_{\text{lab}}} \tag{3-12}$$

Table 3-4 shows these ratios for our full-scale and laboratory platforms. The forcing frequencies are shown using the optimal band found in training earlier in this chapter in Section 3-3-4, St $\in [0.048, 0.065]$, and Table 3-5 gives the resulting full-scale and laboratory forcing frequencies required to match St across this band, alongside the fixed natural frequency.

Table 3-4: Similarity scale factors, full scale to laboratory.

Quantity	Full scale	Laboratory	Ratio (lab/full)
Rotor diameter D	125 m	0.35 m	$\lambda_L = 2.80 \times 10^{-3}$ (1:357)
Wind speed U	8.2 m/s	5.0 m/s	$\lambda_U = 0.610$
Time	—	—	$\lambda_t = 4.59 \times 10^{-3}$

Table 3-5: Strouhal-matched frequency targets, St $\in [0.048, 0.065]$.

	Full scale	Strouhal	Laboratory
Forcing freq. f_f	0.0032–0.0043 Hz	0.048–0.065	0.69–0.93 Hz ($T = 1.458$–1.077 s)
Natural freq. f_n	0.0039 Hz	0.060	0.85 Hz ($T = 1.2$ s)
Freq. ratio f_f/f_n	0.80–1.1	0.80–1.1	0.80–1.1

With the laboratory time scale fixed, training cost can be built directly on physical duration rather than on the wall-clock simulator costs used in the previous section. Cost is now set by the number of forcing periods that must be observed, not by the arithmetic cost of advancing the state. Resolving each forcing period T_f into n_{ctrl} control steps gives a physical duration of T_f/n_{ctrl} per step, and the total training time for a budget of N_{steps} is

$$C_{\text{train}}^{\text{phys}} = \frac{N_{\text{steps}}}{n_{\text{ctrl}}} \cdot \frac{T_f}{\eta}, \qquad n_{\text{ctrl}} = T_f \cdot f_{\text{control}}, \tag{3-13}$$

where η is the acceleration factor of the laboratory (or simulator) time scale relative to full-scale physical time. T_f now varies over the St band, so $C_{\text{train}}^{\text{phys}}$ is likewise reported as a range, bounded by the shortest period (St = 0.065) and the longest period (St = 0.048). Simulation cost and physical cost respond in opposite directions to the control resolution n_{ctrl}, because they are bound by different things. Simulation cost is compute-bound: as established above, each high-resolution step costs a fixed $t_{\text{step}}^{\text{hi-res}} = 0.82122$ s of wall-clock time and each low-resolution step costs a fixed $t_{\text{step}}^{\text{lo-res}} = 0.03930$ s, set by the $O(N^2)$ cost of the vortex-induced-velocity summation over wake elements, and has nothing to do with how finely the forcing period is discretised. For a fixed step budget N_{steps}, refining n_{ctrl} means each step advances a smaller fraction $(1/n_{\text{ctrl}})$ of one forcing period, so the training run completes fewer total forcing periods, $N_{\text{steps}}/n_{\text{ctrl}}$, within the same wall-clock cost. The simulator becomes relatively more expensive per forcing period completed in this case. Laboratory and full-scale cost, by contrast, are physics-bound: Eq. (3-13) ties each step directly to a real-time interval T_f/n_{ctrl}, so refining n_{ctrl} shrinks that interval and reduces physical training time in proportion. Table 3-6 compares all four regimes for a fixed 1-million-step budget across a range of n_{ctrl}, with laboratory and full-scale costs given as ranges over St $\in [0.048, 0.065]$. The high-resolution simulator and the full-scale platform are never the cheapest option at any n_{ctrl} tested, so the practical choice is between the low-resolution simulator and the laboratory, and this choice flips between $n_{\text{ctrl}} = 25$ and $n_{\text{ctrl}} = 50$.

As a caveat, this assumes that the same amount of training steps is required, which may not be the case.

Table 3-6: Cheapest training regime by control resolution, 10^6-step budget.

n_{ctrl}	Sim (low-res.)	Sim (high-res.)	Laboratory	Full scale	Cheapest
10	10.9 h	228.1 h	29.9–40.5 h	271.4–367.6 d	Low-res. sim
25	10.9 h	228.1 h	12.0–16.2 h	108.6–147.0 d	Low-res. sim
50	10.9 h	228.1 h	6.0–8.1 h	54.3–73.5 d	Laboratory
100	10.9 h	228.1 h	3.0–4.1 h	27.1–36.8 d	Laboratory

Note: n^*_{ctrl} denotes the break-even control resolution at which the laboratory physical training cost of Eq. (3-13) falls to equal the (resolution-independent) wall-clock cost of the low-resolution simulator, 10.9 h: below n^*_{ctrl} the low-resolution simulator is the cheaper training regime, above it the laboratory. Across the $\mathrm{St} \in [0.048, 0.065]$ band this crossover lies at $n^*_{\mathrm{ctrl}} \approx 27.4$–37.1.

The training runs in this work operated at an effective control resolution fixed by the simulation timestep: with $\Delta t = 0.3$ and discovered forcing frequencies spanning $St \in [0.048, 0.065]$, the number of control steps per forcing period was $n_{\mathrm{ctrl}} = T_f/\Delta t = 1/(St \cdot \Delta t) \approx 51$–69. This lies at or above the crossover in Table 3-6, i.e. in the regime where, by this accounting, laboratory training would nominally have been the cheaper option — setting aside the practical obstacles of hardware-in-the-loop training (eg. sensor noise) that this cost model does not capture.

3-5 Case Study 1 Conclusion

A grid search over sinusoidal repositioning trajectories was first conducted to understand the solution landscape and provide a non-learned baseline. This resulted in multiple near-equivalent optima with an array power of 0.351. This was compared against two reference points computed in the same FVWM environment: the fully-waked case with a total non-dimensional array power output of 0.247, and, the static-yaw optimal solution first reported by van den Broek et al. [4] and rederived here for this altered FVWM configuration with a higher number of states to allow for proper learning of dynamic trajectories (0.313, a 27% improvement over the fully-waked baseline). The grid search optimum achieved a 42% improvement over the fully-waked baseline and 12% over the static optimum.

The history length, discount factor, and horizon of the RL controller were then selected based on the characteristic frequencies identified in this grid search, before training was carried out on the DelftBlue supercomputer, since training on a laptop with only 200,000 steps did not converge. A sensitivity analysis across five history lengths found no substantial difference in final performance due to history length. It must still be included, because without at minimum history length of 10 steps, convergence to periodic results is not achievable. All optimal RL controllers achieved array powers of between 0.342 and 0.353 (a 38–43% improvement on the fully waked scenario and a 9–13% improvement over the static yaw optimum). This is in the same range as the optimal grid search results, which is promising as it means that the RL controller independently and consistently learns that dynamic policies are optimal by a significant margin, for this mooring line configuration. This underscores the fact that further research into dynamic turbine repositioning is worthwhile.

The optimal repositioning trajectories learned by the RL controller were characterised across ten independently trained controllers for each of the five history lengths, and were found to consistently take the form of an approximately sinusoidal motion, with yaw amplitudes of 26 to 36 degrees, non-dimensional position amplitudes (scaled by 125 m rotor diameter) of 0.28 to 0.44, forcing frequencies of $St = 0.048$ to 0.065 (consistent with the grid search result), and inter-turbine phase offsets of ten to fifty degrees, with the front turbine lagging the rear.

Two cost barriers motivated the training approach taken in this chapter. Firstly, the high-resolution configuration of the FVWM has an $O(N^2)$ discretisation cost that make direct training impractical. This configuration would take 9 days to train one RL controller, which far exceeds the compute job limit of 24 hrs. Hence, training was conducted in low-resolution simulation, with high-resolution runs undertaken for final evaluation and to obtain power output metrics that could be compared to the grid search, static optimum and fully-wake cases run in the high-resolution simulator configuration. Secondly, a Strouhal-similarity comparison was undertaken to determine the likely speed-up or slowdown associated with switching to a laboratory or real-world RL training environment. This showed that at finer control resolutions ($n_{\text{ctrl}} \gtrsim 30$), physical laboratory testing actually becomes cheaper in wall-clock time than the low-resolution simulator, since simulator cost is fixed per step whilst laboratory cost scales down with the physical time each step represents. Full-scale and high-resolution training remained the most expensive options throughout, requiring 27–367 days and 9 days, respectively.

Overall, this demonstrates that dynamic turbine repositioning outperforms the static solutions, and of course, the fully-waked case. The RL solutions were computationally feasible on a supercomputer, the optimal signals were all periodic in nature, and observation of yaw, position and power history served as a suitable proxy for unobservable wake states. Therefore, RQ 2 through RQ 5, and Contributions 1, 2 and 3 have been addressed.

Chapter 4

Laboratory Setup

The experimental campaign investigates dynamic repositioning strategies using two scaled wind turbine models operating in tandem within the WindShape wind tunnel. The design specifications to enable such experiments are described in detail in this chapter. Individual design specifications are highlighted both where they are first introduced, and also in a summary for easy reference at the end of this chapter. The completed setup is displayed upfront in Figure 4-1. In this chapter, the following research question and contribution are addressed:

RQ 6 What are the design specifications for an affordable dynamic turbine repositioning laboratory rig and how does it perform?

Contribution 5 Specification, mechanical design, and experimental commissioning of a WindShaper-compatible laboratory rig for wake-steering repositioning control.

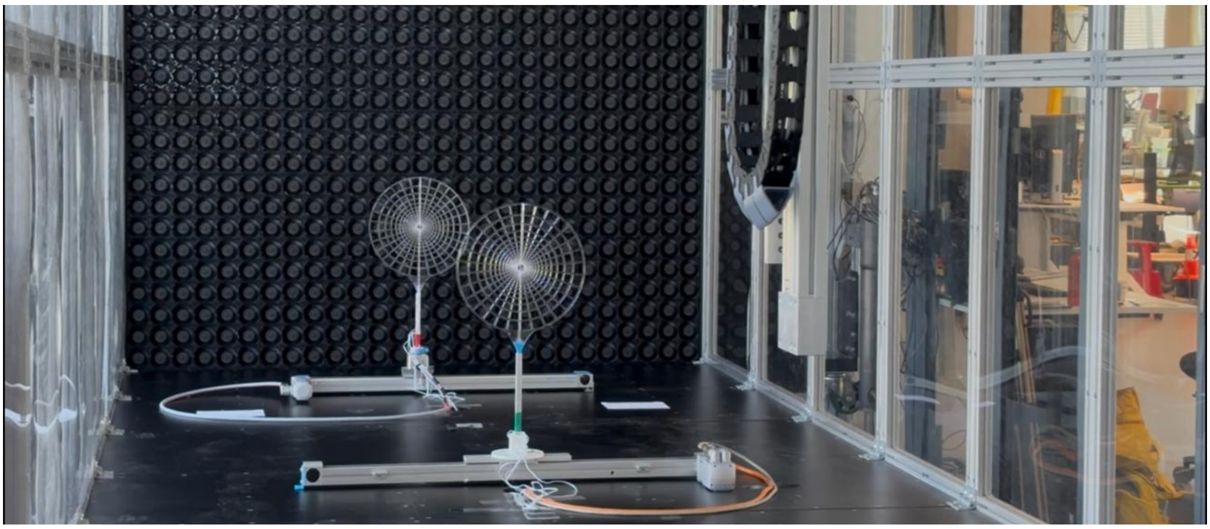


Figure 4-1: The laboratory setup.

4-1 General Experimental Design Details

The experiment aims to act as a proof of concept for dynamic wake steering using reinforcement learning. Accordingly, the setup consists of two turbines aligned in the downstream direction. Ideally, each model turbine would be mounted on an actuator-system providing both linear translation (to emulate repositioning motion) and yaw rotation. However, as the rotational actuators are expensive, we first wish to determine if the actuator is capable of producing the desired motions and if the wake escape phenomenon is visible with the porous discs. To enable more chances for the rear turbine to escape the forward turbine's wake, the dual yaw and translation capability is assigned to the forward turbine. This way it can direct its wake away from the rear turbine. This combined with the rear turbine's translation provides the maximum chance for it to escape the oncoming wake with the available equipment. It also means that the rear turbine acts more as a measurement device than as an extra agent for the RL controller to use for wake escape. Accordingly, the following design specifications apply:

Design Specification DS-1

The system shall consist of two turbines aligned in the downstream direction.

Design Specification DS-2 subject to Constraint C-1

The front turbine shall be mounted on a linear and a rotational actuator, enabling independent translation and yawing, whilst the rear turbine shall be mounted on a linear actuator enabling translation alone.

4-2 Wind Shaper

The custom-designed WindShape [79] wind tunnel at the TU Delft DCSC laboratory was used for these tests. Its specifications are as follows:

- 9 by 9 modules
- 1458 fans
- 16 m/s maximum speed (capable of commanding integer percentage increments of this maximum from 0 to 16, e.g. 50% of 16 results in a wind speed of 8m/s)
- 2.18 by 2.18 by 8.25m (height by width by length) tunnel

Based on advice from the laboratory technical staff, an experimental speed of 5m/s was chosen.

Design Specification DS-3

The wind tunnel shall be able to provide a wind speed of 5 m/s.

4-3 Reference 5 MW NREL Turbine

The experimental setup is based on a scale model of the 5 MW NREL Reference Turbine described in [8]. It has a rotor diameter $D_{\text{full}} = 125\,\text{m}$. These parameters will be used throughout this chapter, so are first highlighted here.

4-4 Dimensional Analysis — Frequency Scaling with Strouhal Number

As with the simulation, it is necessary to retain the same Strouhal number between the full-scale 5 MW NREL reference turbine, simulations and experiments in order to make fair comparisons between the optimal dynamic yaw and repositioning trajectories for the three. As the experimental setup needs to be built alongside the development of simulation results, a "best guess" had to be made to set a range for the forcing frequencies that would capture the likely optima from simulation, before they were obtained. Therefore, the upper and lower bounds from the 775m mooring line case in [10] were chosen as the upper and lower limits for the likely forcing frequencies that experiments would be run at. These are signals with: 150 s period at a stroke of 15 m, and 300 s period at a stroke of 125 m, both with a reference wind speed $U_{\text{ref}} = 8.2\,\text{m/s}$. This gives the reference Strouhal bounds (noting that the faster frequency corresponds to a shorter repositioning distance, and slower frequencies allow for larger repositioning distances):

$$St_{\text{forcing,min}} = \frac{(1/300)\cdot 125}{8.2} \approx 0.05 \tag{4-1}$$

$$St_{\text{forcing,min}} = \frac{(1/150)\cdot 15}{8.2} \approx 0.01 \tag{4-2}$$

These two values bound the non-dimensional forcing frequency range that the laboratory system must reproduce. Proceeding sub-sections go into more details of the scaling for the laboratory design.

4-4-1 Laboratory Turbine Repositioning Distance

At a fixed laboratory wind speed $U_{\text{lab}} = 5.0\,\text{m/s}$ which equates to a WindShaper percentage of 31%, and with the model rotor diameter D as the characteristic length, we obtain the actuation frequency band required for any candidate scale-model:

$$f_{\text{min}}(D) = \frac{St_{\text{ref,min}} \cdot U_{lab}}{D} \tag{4-3}$$

$$f_{\text{max}}(D) = \frac{St_{\text{ref,max}} \cdot U_{lab}}{D} \tag{4-4}$$

The associated actuator stroke range is scaled geometrically by a scaling factor SF, independently of the Strouhal matching, directly from the reference repositioning displacements:

$$d_{f_{\text{min}}} = 125\,\text{m} \cdot SF(D), \qquad d_{f_{\text{max}}} = 15\,\text{m} \cdot SF(D) \tag{4-5}$$

The final values for $d_{f_{\text{min}}}$ and $d_{f_{\text{max}}}$ are included in Section 4-8 after the scale-model diameter is set.

4-4-2 Actuator Kinematics

The repositioning (with the linear actuator) is modeled as sinusoidal motion of amplitude $A = S/2$ (half-stroke) and angular frequency $\omega = 2\pi f$:

$$x(t) = A\sin(\omega t), \qquad v(t) = A\omega\cos(\omega t), \qquad a(t) = -A\omega^2\sin(\omega t), \tag{4-6}$$

Accordingly, the peak velocity and accelerations are:

$$v_{peak} = A\omega, \qquad a_{peak} = A\omega^2. \tag{4-7}$$

These equations come into use in Section 4-4-4 where where the optimal rotor size that minimises the peak velocities and accelerations is determined.

In addition to the repositioning, the turbine with a rotational actuator can also vary its yaw angle. Accordingly we set:

Design Specification DS-4

The system shall be able to command yaw angles between $-30°$ to $30°$ at increments of $5°$, with no more than 1% of error.

4-4-3 Flow Constraints on the Geometric Scaling

The laboratory model also needs to respect the physical shape of the wind tunnel. The most critical dimension is the WindShaper's 2.18m width: the laboratory turbines must be small enough that their wake does not impinge upon the edge of the tunnel and cause rebound effects. A minimum distance of 2D from the wind tunnel edge was specified to mitigate against rebound effects. We assume that the full stroke of the wind turbine translational movement will always be within 6D.

Design Specification DS-5

The rotor shall maintain a minimum distance of $2D$ from the WindShaper side walls throughout its trajectory.

The second critical dimension is the hub height of the turbine. It needs to be tall enough that the turbine rotor sits above the shear boundary layer. To date, there has been no noticeable shear boundary layer detected in the WindShaper. However, there most likely is a small shear boundary layer. To ensure clearance from this region, the bottom of the rotor was set to be 0.4m above the tunnel floor.

Design Specification DS-6

The distance between the WindShaper floor and the bottom of the rotor shall be at least 0.4m.

In addition to wall and floor clearance, the downstream turbine must sit far enough behind the upstream turbine to avoid operating in the most turbulent core of its wake. A spacing of $5D$ was adopted between rotor centrelines in line with typical design standards for wind farms.

Design Specification DS-7

A spacing of $5D$ shall be maintained between the two rotors.

4-4-4 Parameter Sweep To Choose Optimal Design

Equation 4-3 was evaluated over a sweep of candidate model diameters $D \in [0.10, 0.50]$ m; a subset of these results are shown in Table 4-1. The actuator velocity computation has been excluded from the table because as a consequence of the scaling: $f(D) \propto 1/D$ whilst the geometrically scaled stroke $d_f(D) \propto D$, and their product - and hence peak actuator velocity - is *invariant with model scale* ($v_{f\min} = 1.6\,\text{m/s}$, $v_{f\max} = 0.38\,\text{m/s}$ for all D). The peak acceleration is included as it scales with $1/D$: smaller models demand higher actuator bandwidth. Therefore, it is desirable to maximise the rotor diameter in order to minimise the peak accelerations, making it easier to source a motor. The largest rotor diameter that also respects the edge distance constraint is 0.35m.

Table 4-1: Subset of results of the diameter sweep. Rows shaded red exceed the edge distance spacing constraint.

D [m]	Scale	$f_{\min}$ [Hz]	$f_{\min}$ [Hz]	$a_{\max}$ [m/s^2]	$a_{\max}$ [m/s^2]	Edge Dist. [m]	Status
0.10	1:1250	2.50	5.0	25.08	12.04	1.58	Pass
0.25	1:492	1.00	2.00	10.03	4.01	0.68	Pass
0.35*	2:703	0.71	1.43	7.16	3.44	0.08	Pass
0.38	1:328	0.67	1.33	6.69	3.21	−0.07	Fail
0.5	1:246	0.50	1.00	5.02	2.41	−0.82	Fail

*Adopted baseline diameter.

4-4-5 Resulting Scaling Factor

4-5 Scaling Factor

A geometric scaling factor SF relates the full-scale rotor diameter to the model rotor diameter D:

$$SF = \frac{D}{2 \cdot 62.5\,\text{m}} \tag{4-8}$$

For the adopted baseline model with $D = 0.35\,\text{m}$, the $SF \approx 0.003$.

Accordingly, and using the relationships described in the preceeding sections, the following design specifications can be finalised:

Design Specification DS-8

The system shall achieve a minimum frequency of linear actuation of 0.7 Hz.

Design Specification DS-9

The system shall achieve a maximum repositioning distance of 0.36 m, associated with the minimum frequency of linear actuation.

The parameter sweep results for $D = 0.35$m gives $f_{\max}(D) = 1.4$ Hz. Therefore, we set:

Design Specification DS-10 subject to Constraint C-3

The system shall achieve a maximum frequency of linear actuation of 1.4 Hz.

Design Specification DS-11

The system shall achieve a minimum repositioning distance of 0.04 m.

4-6 Rotor Design

4-6-1 Porous Disc Literature Review

It is difficult to recreate real-world conditions in a lab. Ideally a reduced-scale (small) rotating three-bladed turbine would be used to represent a real-world rotating three-bladed turbine. However, much of the research into wake interactions of turbine *arrays*, not individual turbines, successfully uses non-rotating porous discs. Figure 4-2 depicts the three aforementioned laboratory turbine designs.

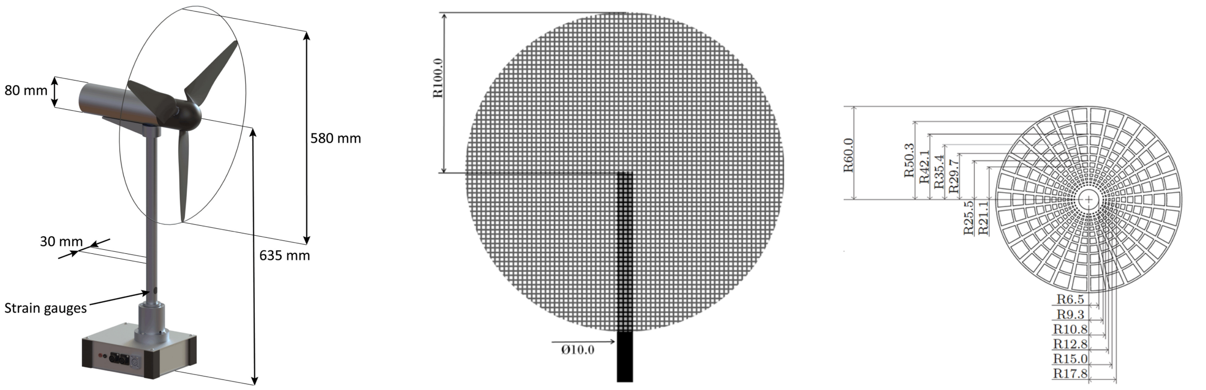


Figure 4-2: From L-R: a rotating 3-bladed MoWiTo turbine [5], a uniform porous disc [6] and a radially varying porous disc [7].

Non-rotating porous discs are designed based on the actuator disc model. Their adoption for wake steering experimental studies is largely due to the results of Aubrun et al.'s [80] experiments on rotating and non-rotating experimental setups which showed that non-rotating porous discs with uniform porosity are suitable for far-wake studies. These results were confirmed in round-robin inter-laboratory tests by Howland et al. [6] for both uniform and radially varying designs. Importantly, they show that the progression to far-wake characteristics (for which these porous discs are valid) occurs faster for the radially varying design than the uniform. This suggests that it is preferable to use the radially varying design, although this is not explicitly stated. Other studies use only the radially varying design. Howland et al. [47] and Ranjbar et al.[81] showed that with appropriate thrust coefficient tuning of discs with radially varying porosity, far-wake (farm or array level) results can be accurately recreated: i.e. the wake deflection, mean wake characteristics and momentum distributions were accurate, but representative internal wake dynamics and tip vortices were not. De Vos et al. [25] and Gutknecht et al. [24] used discs with varying porosity in order to obtain the helical perturbations in the wake. Finally, Camp et al. [7] provide a standardised specification for a porous disc design and again underline the point that they are valid for use in far-wake phenomenon studies with turbulent inflow, but that in the near wake (2-4D) they are *not* valid. In other words, the array-scale energy transport is comparable between porous discs and their rotating analogues, and so too are the wake recovery mechanisms. Based on this, it is reasonable to move forward with a radially varying porous disc design for experiments on wake steering using dynamic turbine repositioning.

4-6-2 Porous Disc Design

According to the literature study in to porous discs in Section 4-6-1, the rotor was designed as a radially varying porous disc, to preserve representative far-wake effects under dynamic yawing. It is a bigger scale version of that described by Camp and Cal in [7]. The disc was scaled in such a way as to preserve the disc solidity and concentric ring spacing ratio. The resultant design is shown in Figure 4-3. It consists of 19 concentric rings, defined by the outer radii measured from the centre of the rotor. Starting from the outermost ring and progressing inwards, the outer radii are: 175, 145, 121, 101.5, 85, 71.5, 60.0, 50.3, 42.1, 35.4, 29.7, 25.5, 21.1, 17.8, 15.0, 12.8, 10.8, 9.3, 6.5 mm.

This was designed in the computer aided drawing tool Autodesk Fusion, and then laser cut from PMMA perspex.

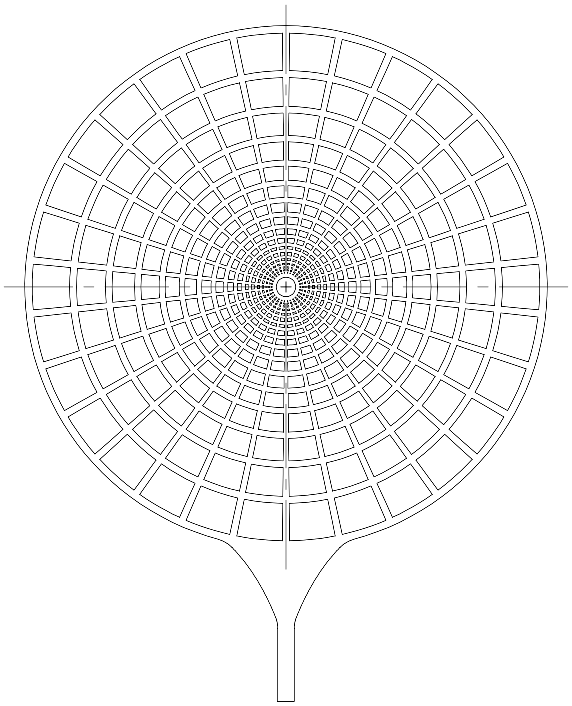


Figure 4-3: Rotor design drawing.

4-7 Load and Torque Requirements

4-7-1 Actuator Torque

The approximate required actuator torque was chosen using the following calculation:

$$\tau = \omega \left(\frac{\pi}{180}\right) \frac{1}{0.2} \left(\frac{1}{2}\right) (1) \left(\frac{D}{2}\right)^2 = 0.10 \text{ N m}, \tag{4-9}$$

where D is the rotor diameter and ω is the desired angular velocity in deg/s. The calculation assumes a combined rotor and tower mass of 1 kg and acceleration from rest to the desired angular velocity in 0.2 s. This is a deliberate overestimate to ensure the purchased actuator is sufficiently capable.

Design Specification DS-12

The motors should be capable of providing a torque of 0.1 Nm.

4-7-2 Structural Integrity of the Rotor and Tower

The structural integrity of the laboratory rotor and turbine tower were chosen to withstand the oncoming wind thrust force. Rough calculations based on the allowable stress of aluminium were conducted, but since the thinnest aluminium square tube available in the workshop (1.5 mm wall thickness) far exceeds requirements, it was selected and no fine tuning was done on these calculations. The rotor material calculations are provided here, because there was more uncertainty about the strength of the available PMMA material when many holes are made in it.

Rotor Material Selection

The rotor was modelled as a circular PMMA plate rigidly attached to the aluminium tower at the bottom. The tower acts as a cantilever support. PMMA was selected because it warps less than metal during laser cutting, only metal or PMMA were available in the faculty workshop, and laser cutting was the only available machining method. It has a tensile strength of approximately 76 MPa. Approximately 70% of the rotor surface consists of cut-outs, giving an effective projected area of

$$A_{\text{eff}} = 0.3\pi R^2, \tag{4-10}$$

where $R = 0.175$ m for the 350 mm diameter rotor.

The aerodynamic load is estimated using

$$F_{\text{wind}} = \frac{1}{2}\rho C_D A_{\text{eff}} V^2 = 0.53 \text{ N}, \tag{4-11}$$

which acts as the principal structural loading on the rotor. Using classical thin-plate bending theory for a clamped circular plate under uniformly distributed loading, the required plate thickness is found to be approximately

$$t_{\text{req}} \approx 0.1 \text{ mm}. \tag{4-12}$$

Since this is substantially smaller than commercially available PMMA sheet thicknesses, strength is not the governing design criterion. The thinnest available 6 mm PMMA sheet was therefore selected to provide sufficient stiffness, robustness during machining, and resistance to vibration while maintaining a low overall rotor mass due to the porous rotor geometry.

4-8 Summary of Design Specifications

Table 4-2 summarises all design specifications derived above. These specifications were provided to the laboratory technicians Will van Geest and Ralph Willekes who constructed the hardware.

Table 4-2: Design specifications summary.

ID	Specification
DS-1	The system shall consist of two turbines aligned in the downstream direction
DS-2	The front turbine shall be mounted on a linear and a rotational actuator, enabling independent translation and yawing, whilst the rear turbine shall be mounted on a linear actuator enabling translation alone, subject to Constraint C-1
DS-3	The wind tunnel shall be able to provide a wind speed of 5 m/s.
DS-4	The system shall be able to command yaw angles between $\pm 30°$ in $5°$ steps, $\leq 1°$ error
DS-5	The rotor shall maintain a minimum distance of 2D from the WindShaper side walls throughout its trajectory.
DS-6	The rotor shall not sit within the shear boundary layer.
DS-7	A spacing of $5D$ shall be maintained between the two rotors.
DS-8	The system shall achieve a minimum frequency of linear actuation of 0.7 Hz
DS-9	The system shall achieve a maximum repositioning distance of 0.36 m, associated with the minimum frequency of linear actuation.
DS-10	The system shall achieve a maximum frequency of linear actuation of 1 Hz, subject to Constraint C-3
DS-11	The system shall achieve a minimum repositioning distance of 0.04 m.
DS-12	The motors should be capable of providing a torque of 0.1 Nm
DS-13	The setup shall be able to sustain approximately 1 N of force from the wind tunnel at 5 m/s.

4-9 Experimental Plan

With the setup described in the previous sections it was decided that the following tests could be conducted:

1. A grid search in the vicinity of the optimal combinations of forcing frequency, amplitude and phase offset between the two turbines from simulations, to determine where the optimal dynamic trajectories in the laboratory environment lie, and to quantify the command tracking accuracy of the system.

2. If the system tracking accuracy is good, an offline validation of the reinforcement learning controller trained on simulation data can be carried out.

The results of these tests are described in Chapter 5.

4-10 Experimental Setup Conclusion

The design of the laboratory setup was described in this chapter, covering the WindShaper wind tunnel, the reference NREL 5 MW turbine used to derive the scaling, the Strouhal-number-based frequency scaling and resulting actuator kinematics, the porous disc rotor design, and the load and torque requirements placed on the structure and actuators. A summary of the resulting design specifications was collated in Section 4-9 for reference.

It is recognised that restricting T2's motion to translational motion driven by an assumed yaw angle rather than a physically actuated one (Section 4-1) is not physically correct. Ideally, each model turbine would be mounted on an actuator-system providing both linear translation (to emulate repositioning motion) and yaw rotation. However, as the rotational actuators are expensive, it is necessary to first determine if the actuator is capable of producing the desired motions, and if the wake escape phenomenon is visible with the porous discs. The consequence of this choice is a limited ability to test RL control policies in the lab in its current state, but this is deliberate choice. The purpose of the laboratory setup in this thesis is to test whether the components are suitable for testing and training dynamic repositioning control schemes.

With the laboratory rig specified, designed, and commissioned, Contribution 1.5 is complete, and the setup described here forms the basis for the experimental campaign presented in Chapter 5.

Chapter 5

Case Study 2: Laboratory Experiments

This chapter presents the laboratory campaign designed to test, on physical hardware, whether the power gains predicted for dynamic turbine repositioning in Chapter 3 can be observed empirically. The test plan and calibration procedure for the laboratory rig specified in Chapter 4 are first described (Sections 6-1 and 6-2), before the fidelity with which the actuators tracked their commanded repositioning signals is quantified (Section 6-3). The resulting strain gauge measurements are then analysed to assess whether an intelligible increase in array power is observable (Section 6-4). This chapter addresses the following research questions, and the following contributions are delivered in this chapter:

RQ 4 How much power can dynamic repositioning recover, compared to a static baseline?

RQ 6 What are the design specifications for an affordable dynamic turbine repositioning laboratory rig and how does it perform?

Contribution 5 Specification, mechanical design, and experimental commissioning of a WindShaper-compatible laboratory rig for wake-steering repositioning control.

Contribution 6 Empirical characterization of the tracking fidelity and repositioning limitations of the laboratory actuators, including quantified commanded-vs-achieved error in frequency, amplitude, and phase.

5-1 Test Plan

Based on the optimal results from the grid search and RL training in Section 3, a matrix of test cases was developed. Each test point in this matrix was run ten times. It was noted that the likely optimal frequencies would not exactly align with the simulation results, mostly due to the fact that whilst porous discs preserve the wake phenomenon under yawing, the exact frequencies of operation are often shifted [82].

Table 5-1: Full forced-motion test matrix (24 points, each repeated 10 times). Cell entries are test_id, one per ψ level.

		ψ (°)			
f (Hz)	A (°)	−180	−90	0	90
0.6	20	1	2	3	4
0.6	30	5	6	7	8
0.8	20	9	10	11	12
0.8	30	13	14	15	16
1.0	20	17	18	19	20
1.0	30	21	22	23	24

5-2 Strain Gauge Calibration

As a proxy for power, fore-aft strain was measured to determine the thrust force experienced due wind blown onto the turbines. Two separate baselines were run to calibrate the strain gauges: (1) the rear turbine fully waked and the front turbine aligned for maximum wind capture, and (2) force spring balance calibration to determine the thrust force associated with each strain gauge measurement.

Calibration for both of these scenarios was repeated ten times each to ensure repeatability. Figure 5-1 shows part of one test case, including the fully waked rear turbine with free front turbine at the beginning of the time series, and then two push and pull forces of 1N on both turbines. This is just to help the reader visualise the process.

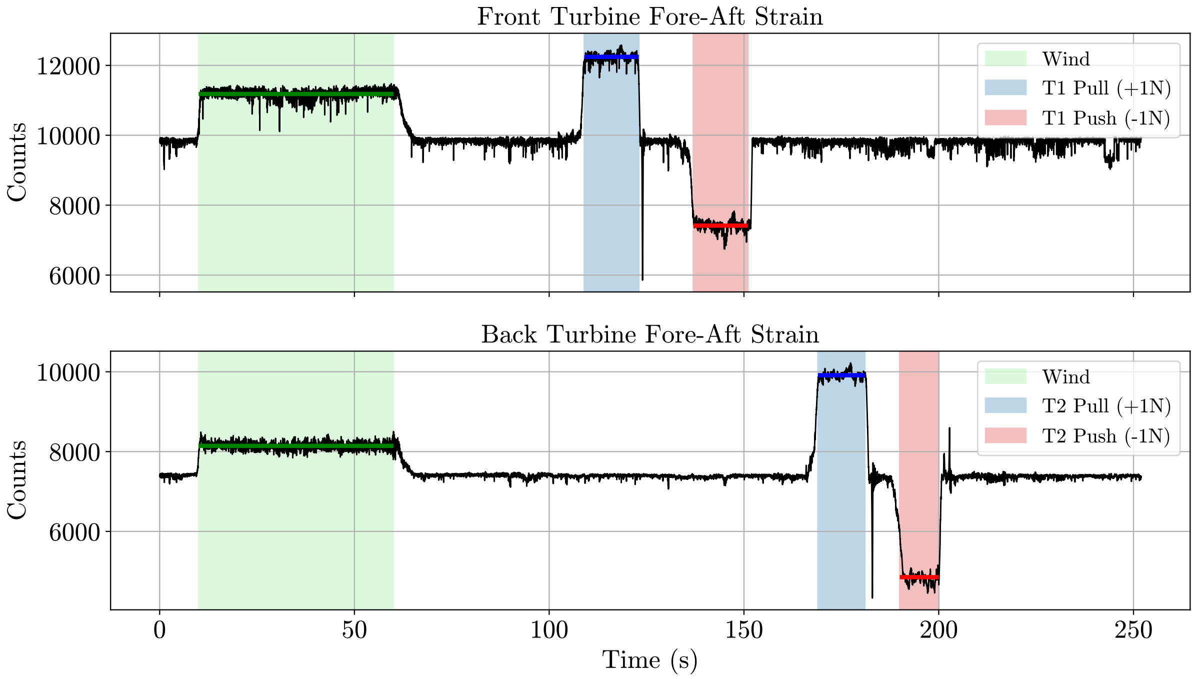


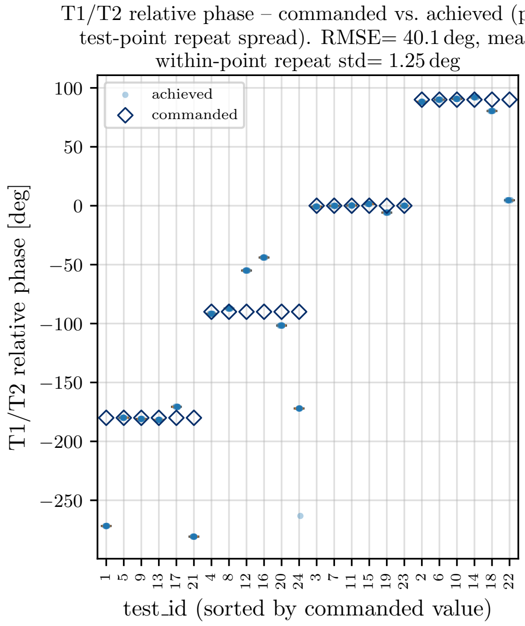
Figure 5-1: Strain gauge calibration data snippet. Wind =5m/s, pull=movement away from wind source, push=movement towards wind source.

5-3 Commanded Versus Achieved Signals

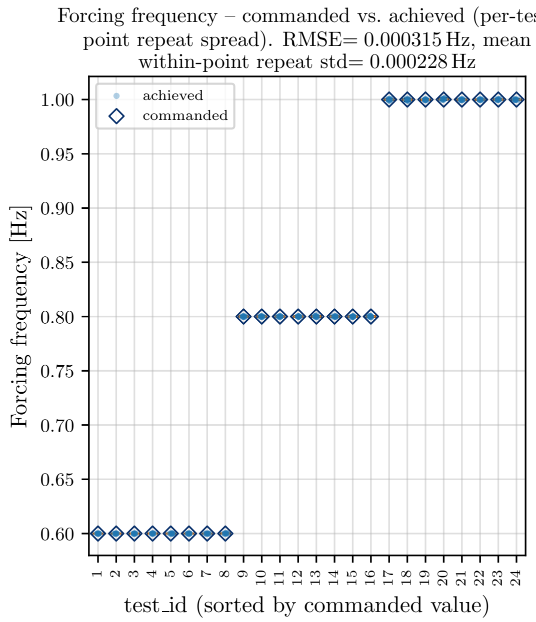
The accuracy of the linear and rotational actuators used for translating and yawing the turbines is summarised in this section.

Phase Tracking: They did not perform well in phase tracking, and their performance degraded over the course of the test campaign. The phase-tracking error is summarised in Figure 5-2a. It is the result of the compounding of errors in amplitude and phase tracking, and also appears to be an in-built error for relative phase tracking inside the actuator programmable logic controllers.

Frequency Tracking: The frequency tracking was, however, very good. Figure 5-2b shows that there is a very small error associated with frequency tracking.



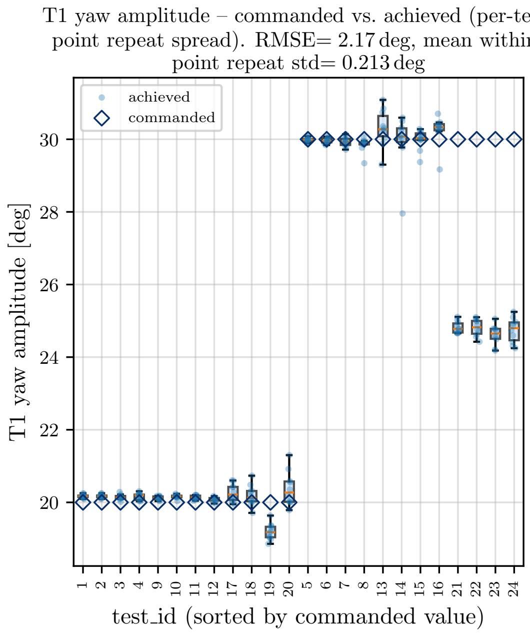
(a) Error in relative phase tracking between T1 and T2.



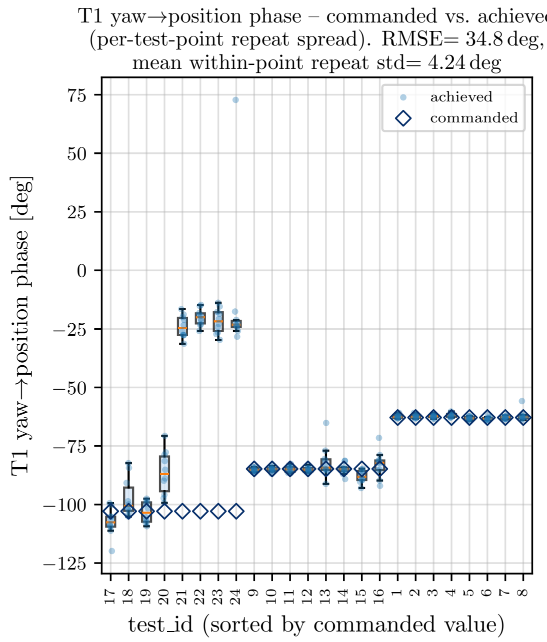
(b) Good frequency tracking achieved by the linear and rotational actuators.

Figure 5-2: Commanded versus achieved signal tracking for the forced-motion actuators.

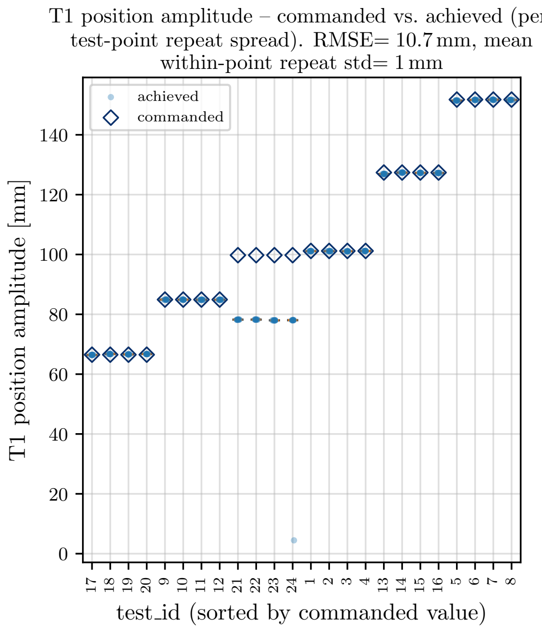
Amplitude Tracking: The amplitude tracking was mostly accurate, but at higher frequencies combined with larger amplitudes (such as the 1 Hz and 30 degrees yaw angle cases) performance significantly worsened. This can be seen in the noticeably worse performance in test cases 21 to 24 in Figure 5-3a. This same issue is the cause of the instances of poor performance in phase tracking for Turbine 1's position commands (this is the phase offset between a yaw command and the associated position change). This can be seen in Figure 5-3b. Again, this same issue drives the drop-off in performance for position tracking in Turbine 1 as well (Figure 5-3c). T2 suffers from a different accuracy problem that also causes some instances of poor amplitude tracking (see Figure 5-3d).



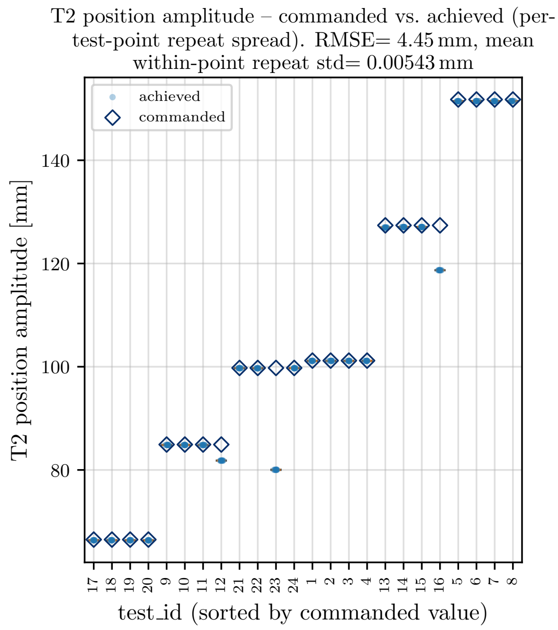
(a) T1 yaw amplitude tracking.



(b) T1 yaw-to-position phase offset tracking.



(c) T1 position amplitude tracking.



(d) T2 position amplitude tracking.

Figure 5-3: Amplitude tracking analysis: for (a-c), the decline in performance is for high-frequency high-amplitude test cases 21-24.

5-4 Strain Results: Is There An Intelligible Increase in Array Power?

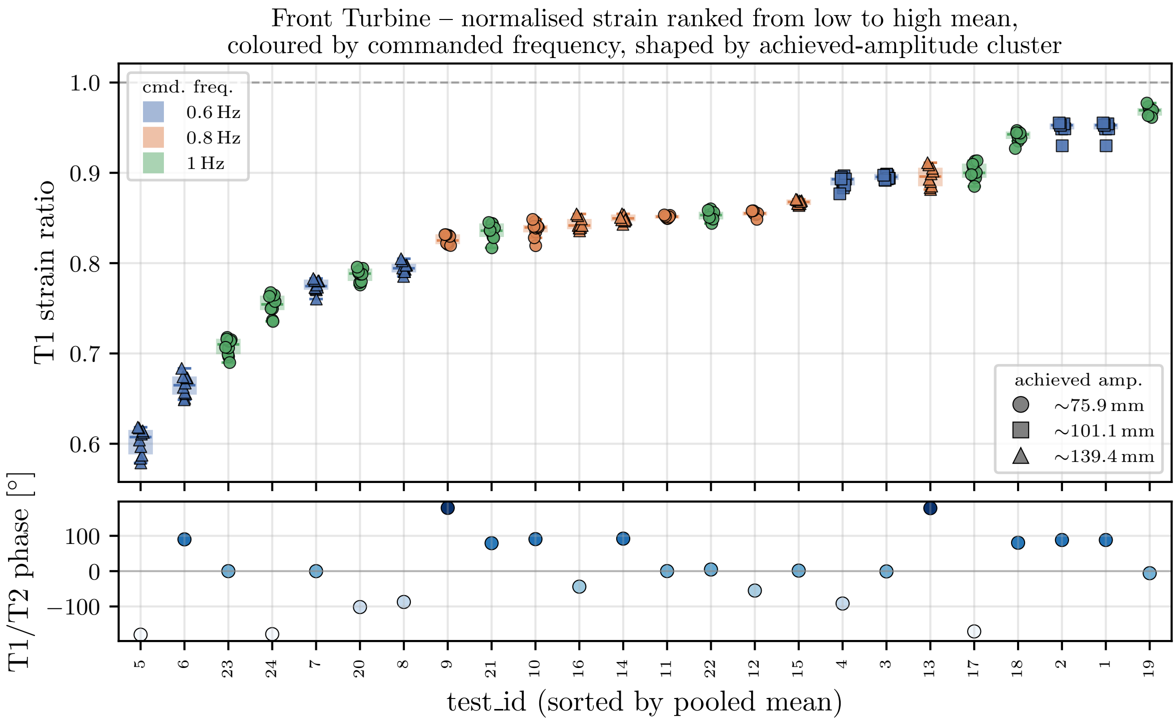
The intent of dynamic turbine repositioning is that the front turbine sacrifices its power for an overall array power increase. The tests show that the losses in thrust at the front turbine under dynamic yawing are so great, that the gain in the rear turbine's thrust cannot make up the difference. In this section, this quandary is investigated.

In order to compare like with like, the strain and thrust values in the aforementioned figures were normalised according to the basline fully-waked T2 case described in Section 5-2. In this way the actual combined change in array thrust force from the wind can be seen. Figures 5-4a and 5-4b illustrate that there is a significant decrease in aft-strain on the front turbine, and also an increase in aft-strain on the rear turbine. One can see that the gains at T2 are not sufficient to make up for the losses at T1. The values in the figures are sorted based on decreasing strain losses for the front turbine, moving in a rightwards direction in Figure 5-4a. Both plots have the same x-axis ordering, to make it easy for the viewer to match which loss in the front turbine corresponds to which gain in the rear turbine strains. These increases for T2, although small, are statistically significant. In Table 5-2, the statistical significance of the strain-decreases at T1 and strain-increases at T2 is shown for each test point, noting that each test point was repeated 10 times to enable this statistical analysis. Figure 5-5 also shows in Newtons that there is a net array thrust loss due to the excessive thrust losses experienced by T1. A discussion of possible root causes ensues in the following paragraphs.

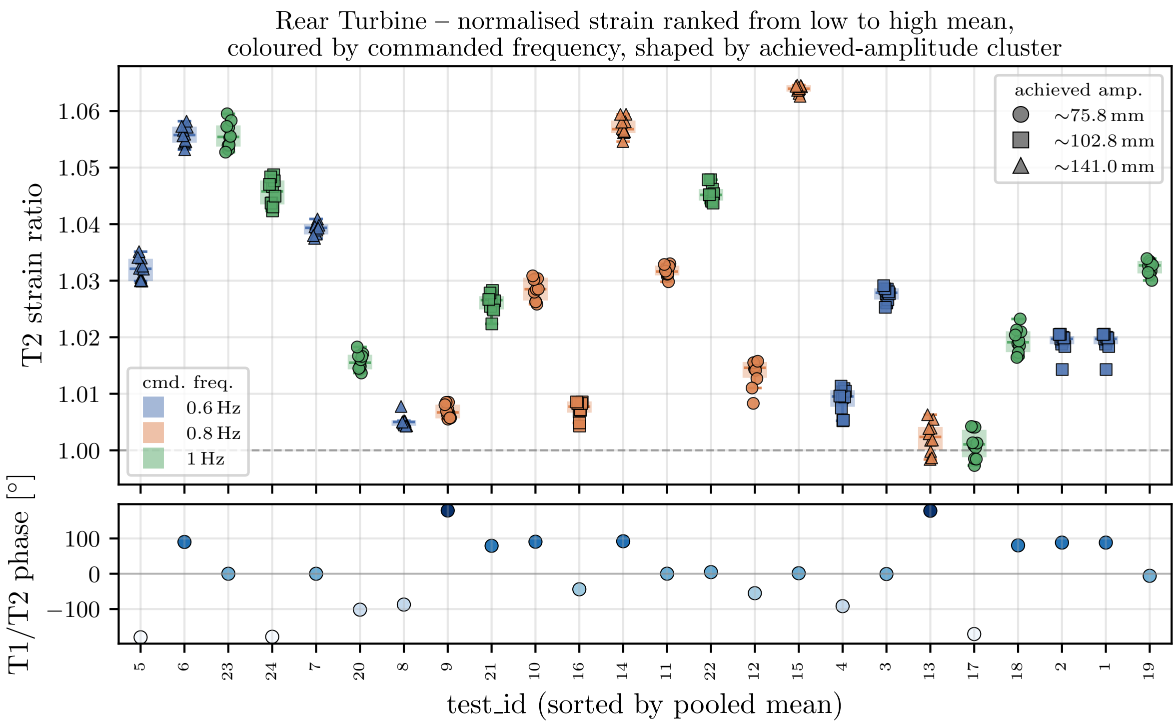
There is an *intra-turbine* trend, but no clear *inter-turbine* trend. The worst performing cases for T1 do not correspond to the best cases for T2, nor is there any other such shared *inter-turbine* trend between the two turbines. For T1, at frequencies of 0.6 and 1 Hz, the higher amplitude yaw cases perform worse than the lower amplitude cases. This *intra-turbine* trend is reversed for the 0.8 Hz cases. For T2, the best performance is obtained at high amplitude regardless of the frequency (see test cases 6, 14, 15, 22, 23 and 24). This means that wake the wake deficit is reduced at T2 for higher amplitude yawing of T1. This makes sense as T1 spends less time directing its wake at T2 in these cases, and it agrees with the simulation results that favour yaw angle amplitudes of about 30 degrees in Chapter 3.

The aerodynamic effects of dynamic yawing with porous discs were recently published by Purohit et al. in 2026 [83], wherein they showed that dynamic yawing does improve wake recovery, especially so at larger yaw amplitudes of 30 degrees. They also found that yaw amplitude has a stronger effect on wake recovery than frequency. Purohit et al. did not investigate the effect of dynamic yaw motions on turbine power capture; they focussed solely on the aerodynamic mechanisms and measuring wake recovery. In the context of the work by Purohit et al., it is likely that the higher amplitude T1 motions are indeed increasing wake recovery and so causing the increased T2 strains in these cases.

The source of the large strain decrease at T1 itself is a separate problem. There are likely two mechanisms at play. Firstly, for small-scale rotors, the thrust of the turbine is known to decrease under yawing [84]. Secondly, although capable of producing realistic wake characteristics, porous discs are still an imperfect substitute for real bladed turbines and likely affect the turbine thrust beyond the aforementioned small turbine effects. Finally, it is certain that this reduction in thrust is not due to vibration or the motion itself. Although there was visible vibration at the extrema of the stoke during experiments, analysis of the strain signals for both the side-side and fore-aft gauges of T1 do not show that vibration is a dominant factor. The dominant factor influencing the strain is the DC gain component; all other factors such as the forced motion, vibration and high frequency noise contribute less than 1% to the total fore-aft strain signal power and less than 5% to the side-side strain signal power. If side-side motion or vibration was causing the drop in thrust, the motion and vibration lines in Figures 5-6 and 5-7 would track higher percentage values.



(a) Front turbine aft-strain normalised by its maximum wind capture baseline.



(b) Rear turbine aft-strain normalised by its fully waked wind capture baseline.

Figure 5-4: Turbine aft-strain normalised by wind capture baseline.

Table 5-2: Fore-aft strain ratio (where the wind-on baseline case = 1.0) per test point means and confidence intervals vs. baseline. Each test point was repeated 10 times and the confidence interval (CI) is quoted as either the 99 or 99.9% CI. Legend: $^{***}99.9\%$ CI, $^{**}99\%$ CI, ns = not significant. Note: Test 1 is omitted as its raw data files were corrupted between graph analysis and creation of this table.

Test	f (Hz)	A (°)	X (mm)	T1 strain ratio [CI]	T2 strain ratio [CI]
2	0.6	20	101.1	0.950 [0.946, 0.955]***	1.019 [1.018, 1.020]***
3	0.6	20	101.1	0.895 [0.894, 0.897]***	1.028 [1.027, 1.028]***
4	0.6	20	101.2	0.890 [0.886, 0.894]***	1.009 [1.008, 1.010]***
5	0.6	30	151.5	0.602 [0.593, 0.612]***	1.032 [1.031, 1.033]***
6	0.6	30	151.7	0.664 [0.657, 0.672]***	1.056 [1.055, 1.057]***
7	0.6	30	151.7	0.775 [0.771, 0.779]***	1.039 [1.039, 1.040]***
8	0.6	30	151.7	0.795 [0.791, 0.799]***	1.005 [1.004, 1.006]***
9	0.8	20	84.9	0.826 [0.823, 0.829]***	1.007 [1.006, 1.008]***
10	0.8	20	84.9	0.838 [0.832, 0.843]***	1.029 [1.027, 1.030]***
11	0.8	20	84.9	0.851 [0.851, 0.852]***	1.032 [1.031, 1.032]***
12	0.8	20	84.9	0.855 [0.853, 0.856]***	1.014 [1.012, 1.015]***
13	0.8	30	126.9	0.896 [0.889, 0.903]***	1.002 [1.001, 1.004]**
14	0.8	30	127.4	0.850 [0.847, 0.852]***	1.057 [1.056, 1.058]***
15	0.8	30	127.3	0.868 [0.866, 0.869]***	1.064 [1.063, 1.064]***
16	0.8	30	127.2	0.844 [0.840, 0.847]***	1.007 [1.006, 1.008]***
17	1.0	20	66.4	0.902 [0.896, 0.907]***	1.001 [1.000, 1.003]^{ns}
18	1.0	20	66.7	0.941 [0.937, 0.944]***	1.019 [1.018, 1.021]***
19	1.0	20	66.6	0.968 [0.966, 0.971]***	1.032 [1.032, 1.033]***
20	1.0	20	66.7	0.787 [0.783, 0.792]***	1.016 [1.015, 1.017]***
21	1.0	30	78.2	0.835 [0.830, 0.841]***	1.026 [1.025, 1.027]***
22	1.0	30	78.2	0.853 [0.850, 0.856]***	1.045 [1.045, 1.046]***
23	1.0	30	78.0	0.707 [0.702, 0.713]***	1.056 [1.054, 1.057]***
24	1.0	30	78.0	0.753 [0.746, 0.761]***	1.046 [1.044, 1.047]***

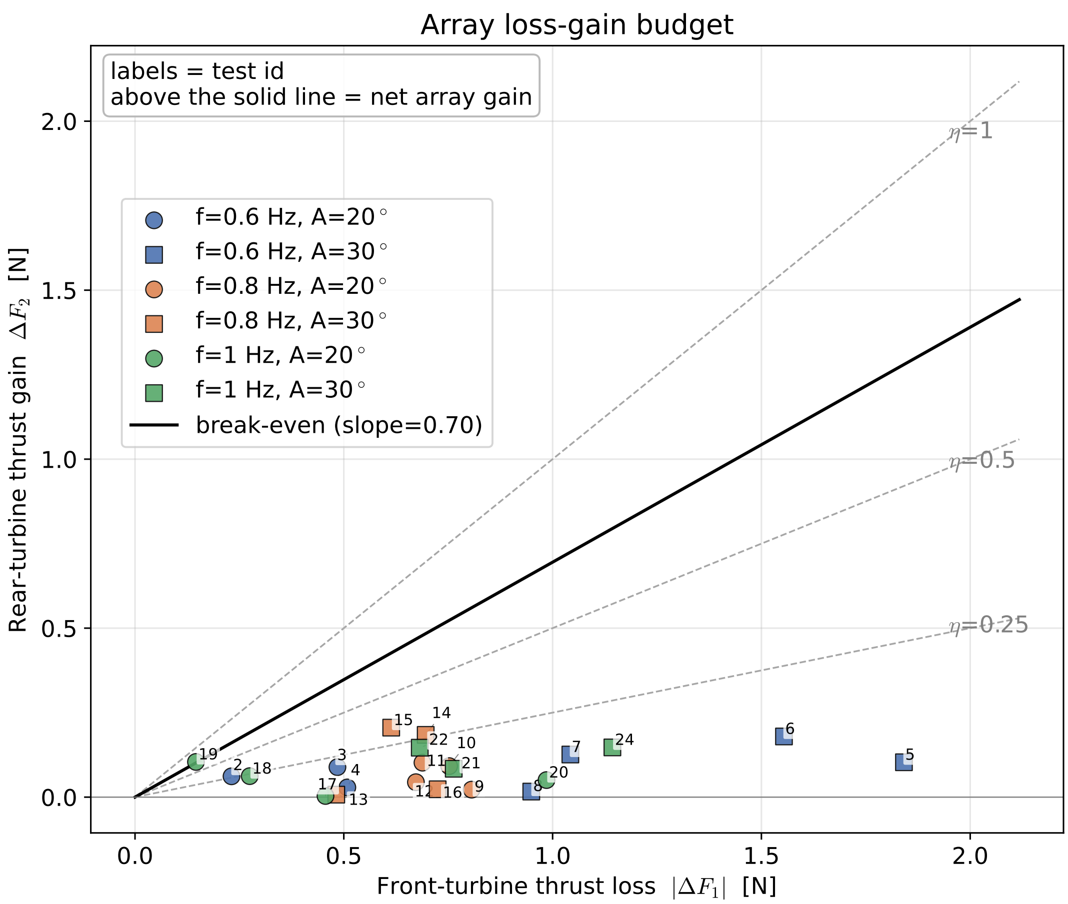


Figure 5-5: Net array thrust losses are due to T1's exacerbated losses under yawing. This figure shows all cases except Test 19 do not 'break even' in terms of the losses at T1 being sufficiently compensated by gains at T2.

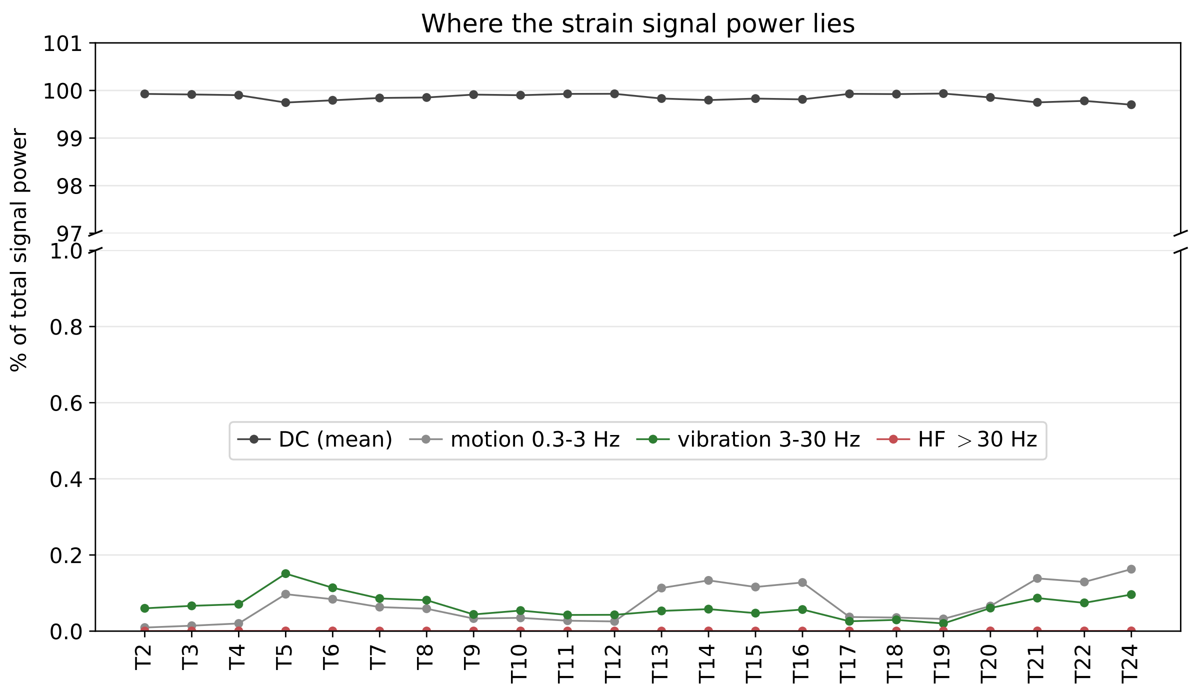


Figure 5-6: Fore-aft T1 strain signal energy across turbines.

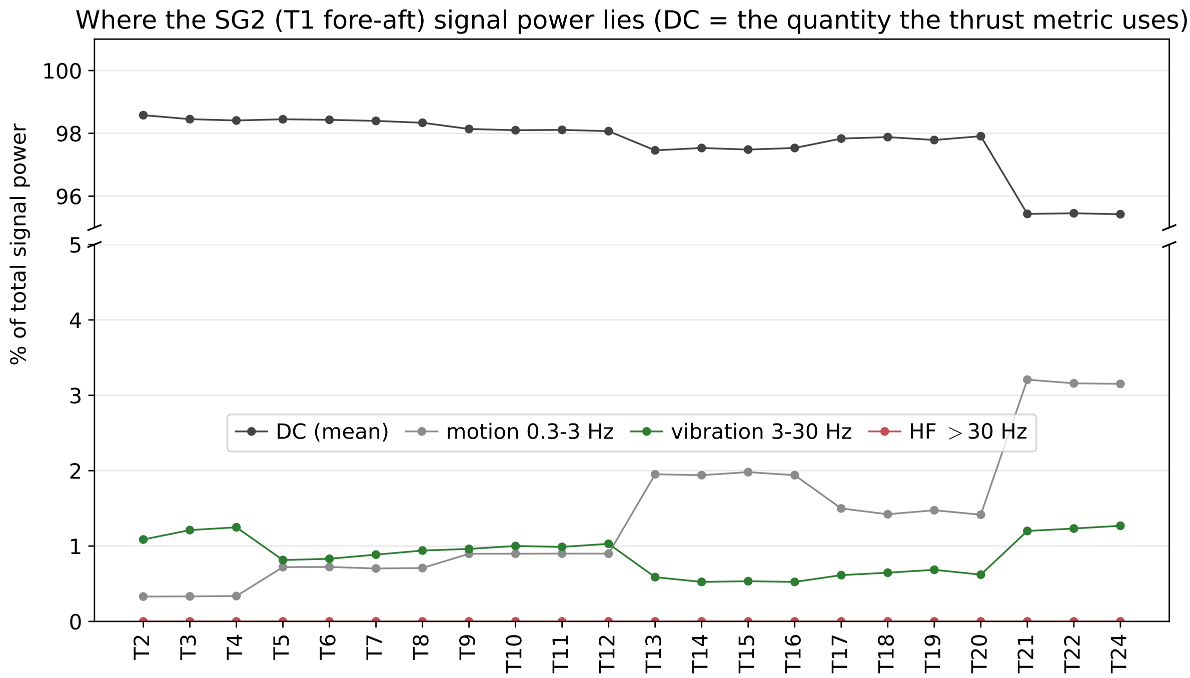


Figure 5-7: Side-side T1 strain signal energy across turbines.

Regardless of the cause of this result, it does mean that the laboratory experiments are unable to support or reject a claim on the efficacy of turbine repositioning. Therefore, it was decided it would not be fruitful to proceed with RL training on this setup. In the future the following four things should be changed to enable RL controller testing: improved accuracy of the actuators at high-amplitude high-frequency motions, using a bladed turbine instead of a porous disc, experimental quantification of the thrust losses due to dynamic repositioning, and implementing a yawing capability for the rear turbine. Without detailed experimental quantification of the thrust losses due to dynamic repositioning, implementing a yawing capability for the rear turbine would not be a fruitful pursuit as the same thrust drops due to yawing will be experienced there too.

5-5 Case Study 2 Conclusion

This chapter tested whether the power gains predicted for dynamic turbine repositioning in simulation are observable in a physical laboratory setting. The actuators tracked frequency well, amplitude adequately, and phase poorly — particularly at high-frequency, high-amplitude conditions. The strain results show a statistically significant thrust decrease at T1 and a small but significant increase at T2, with the T2 gains consistently insufficient to offset the T1 losses, producing a net array thrust loss in all but one test case. Spectral analysis confirms that vibration and side-side motion are not the cause of these losses: over 95% of the strain signal powers resides in the DC component. The losses are more likely attributable to the known sensitivity of porous discs and small-scale rotors to yaw misalignment, though the precise mechanism cannot be determined from these experiments alone. The laboratory experiments are therefore unable to support or reject a claim on the efficacy of dynamic turbine repositioning, and four changes are recommended before further testing is conducted: improved actuator robustness at high-amplitude, high-frequency conditions; replacement of the porous discs with bladed rotors; experimental quantification of the resulting thrust losses; and implementation of a yawing capability for the rear turbine — though the latter should only be pursued once the thrust loss problem is understood and addressed.

Chapter 6

Conclusion

The objective of this thesis was to design, implement, and evaluate a proof-of-concept reinforcement-learning controller for dynamic repositioning of floating wind turbines via wake steering in a physics-based dynamic wake model, and then use these results to design a purpose-built laboratory setup for future testing of dynamic repositioning control techniques. This objective was pursued in three parts: the Free Vortex Wake Model (FVWM) was extended to represent yaw-induced turbine repositioning motion (Chapter 1); a POMDP-based reinforcement learning controller was formulated, trained and evaluated in this extended model (Chapters 2 and 3); and, a scaled laboratory rig was designed, built, and used to test whether the resulting repositioning trajectories could be physically realised and whether they produced a measurable power benefit (Chapters 4 and 5). All parts of this objective were successfully met. This conclusion is structured according to the research questions and contributions that were used to dissect the research objective into manageable work packages. The limitations of, and recommended future work for, each research question is woven into each respective section directly.

6-1 Research Questions

RQ 1 How can we formulate the POMDP of the wind farm for wake steering?

RQ 2 Is RL-based repositioning control feasible for online and/or offline deployment?

RQ 3 What wake-steering policies does the RL agent learn, and how do they compare against baseline strategies?

RQ 4 How much power can dynamic repositioning recover, compared to a static baseline?

RQ 5 What effect do configuration parameters have on RL controller performance, especially the length of aggregated historical state measurements as a proxy for the unobservable states?

RQ 6 What are the design specifications for a dynamic turbine repositioning laboratory rig and how does it perform?

6-1-1 RQ 1: POMDP Formulation

The wind farm wake-steering problem was formulated as a partially observable Markov decision process in Chapter 2, with state, action, observation, transition, reward, and observation functions defined explicitly for the turbine repositioning task. The FVWM's vortex advection breaks down beyond a second row of turbines. This restricted this thesis to a two-turbine array rather than a larger farm, but is sufficient to analyse the wake effect and validate a proof-of-concept turbine repositioning control scheme. PPO was used to optimise the policy due to its ease of implementation framework and suitability for continuous action, stochastic policies. The transition dynamics of this POMDP were supplied by the extended FVWM developed in Chapter 1 (Contribution 1.4), which added support for yaw-induced repositioning motion that the original model did not provide. The discount factor and full horizon were set so as to include at least a full period of the dynamic trajectory to be learnt, and the time required for the wake to travel from the upstream to beyond the downstream turbine. This, as well as the specific configuration parameters for runs in the FVWM, was discussed in Section 3.

Directed graph methods, as used by the FVWM creator himself in [62], could be used in the future if more testing of RL controllers is to be done with the FVWM. However, it is likely more effective to carry out testing in a laboratory environment due to the faster compute times afforded by the faster scale-model optimal frequencies. It would also be beneficial for future works to extend the training regime to a variety of representative time-varying wind inflow conditions in order to predict the year-long increases to farm power with dynamic repositioning techniques. This thesis focussed only on mitigating the worst case: the fully-waked condition. Additionally, inclusion of load and fatigue constraints inside the objective function would provide the necessary information for wind turbine and farm developers to determine the economic viability of turbine repositioning. Ahead of real-world implementation of RL controllers on wind farms, it would also be necessary to limit control actions to a safe set: control barrier functions are the best candidate here, because the wind farm environment is too complex and contains too many states to allow for traditional Lyapunov stability analyses. Orbital and Poincaré-based stability analyses may also prove useful for this complex, non-linear system.

6-1-2 RQ 2: RL Controller Feasibility

Training the RL controller was found to be computationally demanding: an initial attempt to train on a laptop for 200,000 timesteps did not reach convergence, and training was instead carried out for one million timesteps on the DelftBlue supercomputer [77], which was sufficient to reach a clear reward plateau. Each controller took 9–13 hours to train using the low-resolution FVWM configuration described in Table 3-1. Higher resolution FVWM configurations were infeasible; they were estimated to require more than nine days to train a single controller, exceeding the DelftBlue supercomputer job limits (24 hours per job). Training in the laboratory is also feasible: the scale model setup with its faster operating frequencies, enables a reduction in per-controller training time to as little as three hours. The RL controller training feasibility was driven by the fact that only the yaw angle, turbine position and power were observed. This significantly reduced the number of states that the RL controller needed to store in memory. This same goal of minimising the number of states for computation motivated the decision to not use the FVWM's discrete adjoint gradient capability in this thesis, instead choosing a model-free RL approach. The success of the controller under such conditions is an important result, as it shows that with minimal state information similar to the likely measured states in a real wind farm, RL is still able to find optimal dynamic trajectories. In the future, it would be worth conducting a trade-off study to determine if the inclusion of extra state information results in large enough power increases to warrant the extra computational complexity.

6-1-3 RQ 3: Learned Control Policies

The RL agent consistently learned an approximately sinusoidal repositioning policy for both turbines, operating at a shared forcing frequency, converging from near-steady-state or small-amplitude behaviour early in training towards fully periodic motion as training progressed (see Chapter 3). A larger amplitude was preferred for the upstream turbine and a slightly smaller amplitude for the downstream. This indicates that the RL policy was able to exploit a specific wake effect that the grid search, constraining both turbines to the same signal amplitude, could not. Of course, a grid search could be expanded to include more signal combinations, but this would quickly become more computationally expensive than the RL training. The optimal frequencies lay between Strouhal numbers of 0.048 to 0.065, which are near the floating platform's natural frequency Strouhal number of 0.058. This is interesting as it differs from the typical Strouhal numbers expected for dynamic yaw, induction and helix wake mixing which are in the order of Strouhal number 0.2 [78, 24]. It means that the coupling between platform natural frequency and turbine repositioning is a dominant factor. This is expected as the two are interlinked: the turbine repositioning is directly encoded via second order low pass filtering of the yaw signal using the platform natural frequency as the cut-off frequency. The learned policies all achieved array non-dimensional power outputs between 0.342 and 0.353 across all tested configurations, exceeding the fully-waked baseline case and the static optimum from literature in every instance, and in agreement with the grid search optima. The phase offsets all show a preference for a lagging upstream turbine (lagging by 10 to 50 degrees).

6-1-4 RQ 4: Power Gains from Dynamic Repositioning

In simulation, the trained RL controller consistently outperformed the baseline target case of a fully-waked downstream turbine by 38–43%, and also outperformed the static yaw-optimal solution from van den Broek et al. [4] by 9–13%. The static yaw case for a fixed-bottom turbine was chosen as the comparison case, because the static yaw case for a floating turbine is self-limiting. (For floating turbines, yawing displaces the platform laterally in the direction opposite to the wake's aerodynamic redirection: a static yaw sufficient to steer the wake away from the downstream turbine simultaneously repositions the upstream turbine such that the wake drifts back toward it, unless the downstream turbine also repositions in turn. Dynamic floating turbine repositioning avoids this problem through continuous coordinated wake escape in an oscillatory manner.) These results were confirmed via a dedicated grid search over sinusoidal trajectories with frequency, amplitude and phase offset properties in the same range, which resulted in an almost identical array power gain. This means that for this mooring line configuration, dynamic repositioning trajectories do outperform static, and it would be well worth further developing this concept to determine its suitability for future deployments on real wind farms.

Due to computational constraints, the statistical characterisation of optimal signal properties in Chapter 3 was based on a limited number of ten repeat trainings on five different history lengths in the low-resolution FVWM configuration. With more repeat trainings on a wider range of mooring line slackness configurations, a higher level of statistical significance can be attained. In using a high-resolution variant of the FVWM for training, there is the potential for different more accurate wake characteristics to be exploited. Care was taken to ensure power gains were not over-quoted due to training in the low-resolution FVWM; all low-resolution-trained controllers were run via foward-pass again in the high-resolution FVWM to determine their power production capabilities in this environment. It was these high-resolution values that were compared to the other high-resolution FVWM configuration comparison cases: fully-waked, static optimum and grid search. This was necessary because the low-resolution FVWM overpredicts the power gains.

6-1-5 RQ 5: Effect of History Length as a Surrogate for Unobservable States

Controller performance was found to be largely insensitive to history length within the tested range of 10 to 100 timesteps, with all history lengths producing array power outputs within a narrow band (0.342 to 0.353), each substantially exceeding every comparison baseline. That said, not including history length severely limited the ability to achieve periodic results, so its inclusion is necessary.

6-1-6 RQ 6: Laboratory Setup Specifications

A system consisting of two scaled versions of the 5 MW reference turbine was designed and built for testing dynamic repositioning control techniques. Porous discs were used for the turbine rotors because they have been shown in literature to retain the necessary wake characteristics under yawing, although they do cause a shift in the resultant optimal frequencies and exacerbate a drop in thrust under yawing beyond what real three-bladed rotating turbine rotors would experience.

The upstream turbine was mounted on a linear and a rotational actuator such that it could yaw and rotate, and the downstream turbine was constructed with available materials to translate according to a hypothetical yaw angle. Tests were conducted on this setup to determine its efficacy for testing dynamic repositioning controllers, in support of a future purchase application for an additional rotational actuator to enable both turbines to yaw. Testing showed good frequency command tracking, and good amplitude and phase offset tracking at low amplitudes and frequencies. At the combination of high amplitude and high frequency, phase tracking and amplitude tracking deteriorated. Due to the tracking error associated with the current laboratory setup, RL was not implemented in the laboratory. Modest but clear increases in the downstream turbine's thrust were noticeable with this setup, and the sacrificial decrease in the upstream turbine's thrust was clearly visible, and too large to be offset by the gains downstream. The use of the porous disc and the small rotor size causes a significant drop in expected thrust at the upstream turbine.

Implementation of the following recommendations would render the setup suitable for training RL controllers. Firstly, it is necessary to improve the bandwidth and tuning of the laboratory position actuators to reduce phase-tracking error, particularly at higher forcing frequencies. Secondly, if bladed rotating turbines are available, they should be used for dynamic turbine repositioning tests instead of porous discs, but if not, porous discs can still be used to prove the concept although they exacerbate the thrust-decreases under yawing. In other words, the concept can be proven with porous discs in terms of measuring either an increase or a decrease in thrust, but to make quantative claims about the thrust, a three bladed rotor would be required. Finally, it is recommended to procure and integrate a dedicated yaw actuator for T2.

The key driver for future investment in the laboratory setup is thus: laboratory training with a small, scaled setup would reduce training time to a tenth of that required for the high-resolution FVWM simulation, and likely even further. It is also, in some feasible configurations, faster than low-resolution FVWM simulation. In pursuit of a realistic and time-efficient RL controller training environment, and noting that high-resolution configurations of the FVWM would likely take more than nine days to train a single controller, a laboratory setup with two yawing and translating turbines would be an excellent choice.

6-2 Contributions Delivered

Simulation Contributions

Contribution 1 A POMDP formulation and RL controller for dynamic wake-steering repositioning, trained on the extended FVWM.

Contribution 2 Characterisation of the optimal repositioning signals discovered by the RL controller, benchmarked against literature strategies (Starink et al. [10] and van den Broek et al. [4]) and standard grid search techniques.

Contribution 3 Quantification of the training time requirements for the RL controller in the FVWM.

Contribution 4 An extension to van den Broek et al.'s Free Vortex Wake Model (FVWM) [4, 9] to simulate yaw-induced turbine repositioning motion.

Laboratory Contributions

Contribution 5 Specification, mechanical design and build, and experimental commissioning of a WindShaper-compatible laboratory rig for wake-steering repositioning control.

Contribution 6 Empirical characterisation of the tracking fidelity and repositioning limitations of the laboratory actuators, including quantified commanded-vs-achieved error in frequency, amplitude, and phase.

6-2-1 Contribution 1: POMDP and RL Formulation

A POMDP formulation of the dynamic wake-steering repositioning problem was developed, with state, action, observation, transition, and reward functions defined explicitly for the two-turbine repositioning task (Chapter 2), and a corresponding model-free RL controller was implemented and trained on the extended FVWM (Chapter 3). This formulation is directly reusable for other dynamic wake models or farm configurations, requiring only the substitution of the transition dynamics.

6-2-2 Contribution 2: Optimal Signal Characterisation

The repositioning signals discovered by the RL controller were characterised in terms of amplitude, frequency (Strouhal number), and inter-turbine phase offset, and benchmarked against the static yaw optimum of van den Broek et al. [4] and a dedicated grid search over sinusoidal trajectories informed by the periodic case of Starink et al. [10]. This provides a quantified reference for the signal properties that future dynamic repositioning controllers should target. Only a single mooring-line slackness configuration from Starink et al. was reproduced here; their wider parameter suite is not directly comparable without recomputation within the FVWM.

6-2-3 Contribution 3: Computational Complexity and Training Time

The computational cost of training the RL controller was quantified across FVWM resolutions and hardware configurations, establishing that low-resolution training is feasible on HPC resources (9–13 hours per controller) while high-resolution training is not (exceeding nine days), and that laboratory-based training offers a substantially faster alternative (the minimum being three hours). This cost analysis directly motivates using the laboratory setup as a training environment.

6-2-4 Contribution 4: FVWM Extension

The Free Vortex Wake Model of van den Broek et al. [4] was extended to represent yaw-induced turbine repositioning motion (Chapter 1), which the original model did not support. This extension supplies time-varying wake dynamics suitable for any repositioning control study, not just the RL approach taken here.

6-2-5 Contribution 5: Laboratory Setup Specification

A WindShaper-compatible laboratory rig for wake-steering repositioning control was specified and designed (Chapter 4), comprising two scaled porous-disc turbines, with the upstream turbine actuated in both yaw and translation and the downstream turbine translated according to a prescribed yaw trajectory. The testing results support a future procurement case for a second yaw actuator.

6-2-6 Contribution 6: Laboratory Commissioning and Empirical Characterisation

The rig was commissioned and its actuator tracking fidelity empirically characterised (Chapter 5), including quantified commanded-versus-achieved error in frequency, amplitude, and phase across the test matrix. This characterisation identifies the specific actuation limitations (phase and amplitude tracking at combined high frequency and amplitude, particularly for T1) that must be addressed before the rig can deliver a conclusive empirical test of dynamic wake steering, and serves as a concrete improvement roadmap.

6-3 Closing Statement

This thesis demonstrates that a reinforcement learning controller, trained on an extended, time-varying wake model, can learn dynamic turbine repositioning strategies that substantially outperform the fully-waked target problem by 38–43%, as well as also surpassing static yaw-based wake steering optima uncovered by the FVWM creators, van den Broek et al. [4] by 9–13%. These results were supported by a grid search of sinusoidal signals based on the work by Starink et al. [10, 43]. Physical validation of this result was constrained by identifiable, and in principle correctable, limitations in the laboratory actuators' tracking fidelity, lack of rear turbine yaw motion, and thrust loss due to yawing porous discs, rather than by any evidence against the underlying wake steering concept itself. The laboratory characterisation work presented here provides a concrete, quantified foundation from which future iterations of this experimental setup can be improved, and on which a more conclusive empirical test of dynamic wake steering can ultimately be built.

RL controllers trained on dynamic wake models or laboratory setups are therefore a promising alternative to traditional LUT-based wind farm control. When the controller is built as a neural-network of relationships between observable states, actions and rewards (as is the case in this thesis), the real-time deployment can be just as fast as for LUT-control. The work in this thesis indicates that it is worthwhile to continue developing the dynamic turbine repositioning technique for mitigating the wake effect. The next phase of research should focus on validating the power gains in the context of yearly wind conditions, turbine and mooring line fatigue and load considerations, and safety constraints in the controller.

Appendix A

Appendix A: Surveys of Real Life Wind Farm Wake Steering Test Campaigns, and Existing RL Control of Wind Farms

A-1 Survey of the Most Recent Real Life Wind Farm Wake Steering Test Campaigns

Table A-1: Comparative summary of wake steering field tests in commercial wind farms. LUT = Look-Up Table; FLORIS = FLOw Redirection and Induction in Steady State; SCADA = Supervisory Control and Data Acquisition; D = rotor diameter.

Category	Howland et al. 2019	Fleming et al. 2019, 2020	Doekemeijer et al. 2021	Simley et al. 2021	Howland et al. 2022	Xie et al. 2024
METHOD	Yaw misalignment control using LUT computed with FLORIS and site-specific wind data.	Gradient ascent (Adam optimisation) on historical SCADA data.	Open-loop FLORIS-calibrated LUT; yaw misalignment optimised for 2 upstream turbines; 3-turbine arrays or turbine pairs by wind direction.	Same as Fleming et al. 2020, but only for positive wake offsets.	Physics-based, data-assisted flow control model to predict the power-maximising control strategy, paired with an energy-maximising control protocol.	Yaw offset control based on LUT built on 2 years of SCADA data informed wake model.
SITE	5 turbines (T2 and T4 steer wake to benefit T3; T1 and T5 are reference turbines). Flat and complex terrain.	Utility-scale farm, Alberta, Canada; 6 turbines at 3.5D spacing; stable, nocturnal low-turbulence conditions.	Commercial onshore farm, Italy (TUDelft/CL-Windcon); 3-turbine arrays; 5.2D-6.5D spacing.	Commercial onshore farm of 7 turbines; only 2 turbines studied at 3.7D spacing.	Commercial onshore farm of 100 turbines; only 3 wake-affected turbines studied.	2 turbines in a commercial onshore farm of 24 turbines in mountainous terrain.
ASSUMPTIONS/ LIMITATIONS	Wind direction stationary within 1-hour toggle window; FLORIS valid; load envelope pre-validated for 20° yaw offset.	Stable nocturnal atmosphere; fixed yaw offset schedule per wind direction sector; gradient ascent converges to global optimum on historical data.	Inflow stationary during measurement windows; FLORIS surrogate calibrated to site; open-loop adequate.	Results inflated due to winter high-wind test period; fixed yaw direction sector; gradient ascent converges to global optimum on historical data.	3D wake effects unmodelled; not all wind directions studied; yaw angles ±25°. Buoyancy suppresses/contributes to turbulence.	Historical years valid as baseline; wake model calibrated from SCADA captures site-specific behaviour; yaw offsets only applied below rated wind speed.
SIGNIFICANCE	First multi-turbine commercial farm campaign; FLORIS field validation; load constraint characterisation.	First statistically significant multi-turbine field demonstration; gradient ascent directly on SCADA without wake model.	First European onshore field experiment (CL-Windcon).	Highlights wind speed dependence of feasibility and energy improvements.	First full-scale test and validation of a physics-based, data-assisted flow control model.	Combination of wind direction calibration, model tuning, LUT generation, and external controller deployment.
KEY RESULTS	6.4-6.6% overall wake loss reduction; roughly half of the predicted static optimal values.	7-13% power increase at site-average wind speeds; 28-47% at low wind speeds. Farm power variability reduced by 72%.	Up to 35% power gain for 2-turbine interactions; up to 16% for 3-turbine interactions.	Estimated 5.6% net reduction in wake losses for the two-turbine combination.	Energy production increased by $3.0 \pm 0.7\%$ for 6-8 m/s winds and $1.2 \pm 0.4\%$ across all wind speeds.	Yaw execution accuracy was poor, but power gains of 7-10% were observed.

A-2 Survey of Existing RL Control for Wind Farms

Table A-2 provides the results of a comprehensive survey of this area of research: what RL types are frequently used, whether model-based or model-free is most popular, typical problem formulations, simulation environments and computational costs. Of the 44 papers surveyed, only six are true model-free implementations. The others are almost all hybrid implementations; they use partial models or knowledge-assisted model-free techniques.

Table A-2: Reinforcement Learning approaches for wind turbine and farm control (including some especially useful general machine learning approaches that do not use RL).

Authors (Year)	RL Type	State s	Action a	Value Fn	Reward r	Environment	Computing Cost
Marden et al. (2013) [85]	Model-free game-theoretic bandit	-	AIF	-	P_f	Static Park	L
Saenz-Aguirre et al. (2019) [86]	MultiLayer Perceptron with BackPropagation (MLP-BP) Artificial Neural Network (ANN) RL	$\{\gamma, U, P\}$	γ_d	Q	$\{P, L\}$	-	L
Verstraeten et al. (2019) [87]	Model-based (Gaussian Process Policy Iteration)	$\{\gamma, P\}$	-	Bayes	P	FLORIS	M
Bui et al. (2020) [88]	Multi-Agent Deep RL	$\{\beta, U, TSR, AIF, P\}$	$\{\beta, TSR\}$	DQN	P_f	-	M
Saenz-Aguirre et al. (2020) [89]	Q-learning + ANN + PSO Pareto optimal front	$\{\gamma, U, \omega_r, L\}$	γ_d	ANN-Q	$\{P, L\}$	FAST	M
Stanfel et al. (2020) [90]	Distributed Reinforcement Learning	$\{U, \phi, P, \gamma\}$	γ_d	TD	P_f	FLORIS	L
Verstraeten et al. (2020) [91]	Multi-agent bandit (MATS Multi-Agent Thompson Sampling)	-	γ_d	Bayes	$\{P, P_f\}$	FLORIS	L
Xu et al. (2020) [92]	Model-free decentralized Q learning	-	AIF	Tab-Q	P_f	No wake model	L
Zhao et al. (2020) [93]	KA-DDPG	$\{U, P, \gamma\}$	γ_c	DDPG	P_f	WFSim	M
Dong et al. (2021) [75]	DDPG with reward regularisation	$\{U, \gamma, P\}$	γ_c	DDPG	$\{P, P_f\}$	SOWFA	H

Continued on next page

Authors (Year)	RL Type	State s	Action a	Value Fn	Reward r	Environment	Computing Cost
Korb et al. (2021) [76]	Model-based	$\{U, \text{helix}, \tau_g\}$	helix	Critic NN	$\{P_f, \tau_g\}$	LES	H
Stanfel et al. (2021) [94]	GARLIC framework distributed MARL	$\{U, \phi, \gamma, W\}$	γ_d	Tab-Q	P_f	FLORIS	L
Verstraeten et al. (2021) [95]	Model-based and Multi-armed Bandit	$\{\gamma, P\}$	$\Delta\gamma$	Bayes	$R - \lambda L$	FLORIS	L
Dong & Zhao (2022a) [61]	Composite Experience Replay Deep Deterministic Policy Gradient	$\{U, P, \gamma, W\}$	γ_c	DDPG	P_f	WFSim	M
Dong & Zhao (2022b) [96]	DDPG with grouping	$\{U, P, \gamma\}$	$\{\gamma_c, AIF\}$	Critic NN	P_f	WFSim	M
Dong & Zhao (2022d) [97]	Preview-based Robust DRL with H-infinity game formulation	$\{P, U, E, R, C_T\}$	$\{\gamma_c, C_T\}$	Critic NN	E minim.	WFSim	M
Fernandez-Gauna et al. (2022) [98]	Actor-Critic RL	$\{U, \omega_r, \tau_g, \beta, P\}$	$\{\beta, \tau_g\}$	AC	η_P	OpenFAST	M
He et al. (2022) [99]	Actor Bagging DDPG	$\{U, P, \gamma\}$	γ_c	DDPG	P_f	WFSim	M
Li et al. (2022) [100]	NOT RL,Bilateral Convolutional Neural Network	-	-	-	-	LES	L
Monroc et al. (2022) [101]	MARL	$\{U, \phi, \gamma, P\}$	γ_d	Tab-Q	P_f	FAST.Farm	L
Neustroev et al. (2022) [102]	DRL	$\{U, \phi, \gamma\}$	γ_c	DNN	P_f	FLORIS	L-M
Padullaparthi et al. (2022) [103]	2-agent DRL, FALCON	$\{P_f, L, \gamma, \beta\}$	$\{\gamma, \beta\}$	FALCON	$\max P_f, \min L$	FLORIS+ccBlade	M-H

Continued on next page

Authors (Year)	RL Type	State s	Action a	Value Fn	Reward r	Environment	Computing Cost
Deng et al. (2023) [104]	Multi-agent policy gradient (MADDPG Multi-Agent Deep Deterministic Policy Gradient)	$\{U, \phi, \gamma, AIF, W\}$	γ_c	CAC	P_f	Custom wake	M
Kadoche et al. (2023) [74]	MARLYC	$\{U, \phi, \gamma, P\}$	γ_c	CAC	$\{P, P_f\}$	FLORIS	L-M
Monroc et al. (2023) [68]	MARL	$\{U, \phi, \gamma, P\}$	γ_c	Linear AC	P_f	FAST.Farm	L-M
Peng & Feng (2023) [105]	RL + function approximation	$\{\beta, U, \omega_r\}$	β_c	Q_θ	P	SCADA	L
Tavakol Aghaei et al. (2023) [106]	Model-free Bayesian RL	$\{\omega_r, U, TSR, P\}$	load$_c$	Bayes	AEP	-	L
Xie et al. (2023) [107]	RL-MPC hybrid; DNN for dynamics approximation + MPC for real-time control policy)	$\{\omega_r, \beta, \tau_g, U\}$	$\{\tau_g, \beta\}$	DNN	P	FAST	M
Espinoza et al. (2024a) [108]	Double-DQN	$\{\omega_g, \beta, U, P\}$	β_d	DDQN	ΔP minim.	OpenFAST	M
He et al. (2024) [60]	NOT RL - physics-guided deep learning	-	-	-	AEP	FLORIS	L
Li et al. (2024) [109]	Hybrid: deep learning (RBFN) + value-based RL (DQN) + Model Predictive Control (MPC); single turbine	$\{\omega_r, \beta, U\}$	$\{\omega_r, \beta\}_d$	DQN	$P + L$	-	M
Liew et al. (2024) [110]	Model-free RL	$\{U, \phi, P\}$	γ_c	PG	P_f	HAWC2Farm	M

Continued on next page

Authors (Year)	RL Type	State s	Action a	Value Fn	Reward r	Environment	Computing Cost
Mazare (2024) [111]	Adaptive model-free optimal RL NN	$\{\omega_g, \omega_r, \beta\}$	β_c	AC	P	FAST	L
Nabeel et al. (2024) [112]	Fuzzy DDPG	$\{\omega_g, \beta, U, L\}$	β_c	DDPG	$\{P, L\}$	OpenFAST	M-H
Wang et al. (2024) [113]	Model-free RL	$\{U, \gamma, P\}$	γ_d	Q	P_f	Gauss-Curl + FLORIS	L
Mei et al. (2025) [26]	Model-based DRL	$\{U, x_f, P, L\}$	γ_c	Critic NN	$\max P_f, \min L$	FLORIS Wake, with MoorPy, FAST and Custom DRL physics	M
Mole et al. (2025) [114]	Closed-loop RL directly coupled to LES	$\{S_{LES}, W, P\}$	γ_c	AC	P_f	LES	H
Monroc et al. (2025) [69]	MARL	$\{U, \phi, \gamma, \beta, \tau_g, L\}$	$\{\gamma, \beta, \tau_g\}$	MARL	$\{P_f, L\}$	FLORIS+FAST.Farm	M-H
Yuan & He (2025) [115]	Reinforcement learning-enhanced neural network sliding mode control	$\{\omega_r, \beta, U, L\}$	β_c	DDPG	ΔP + load bal.	FAST	M
Kadoche et al. (2025) [71]	attention-based DRL	$\{U, \phi, \gamma\}$	γ_c	Critic NN	P_f	FLORIS	L
Espinoza et al. (2026b) [116]	Double-DQN	$\{\omega_g, \beta, U, P\}$	β_d	DDQN	$P + L$	OpenFAST	M

Symbol Definitions for Table A-2:

U	Wind speed
ϕ	Wind direction
γ	Yaw angle
β	Blade pitch angle
ω_r	Rotor speed
ω_g	Generator speed
τ_g	Generator torque
P	Turbine power
P_f	Farm power
AIF	Axial induction factor
C_T	Thrust coefficient
L	Structural/fatigue load metrics
x_f	Floating turbine position
W	Wake state representation
R	Reference signal
E	Tracking error
TSR	Tip speed ratio
S_{LES}	LES-resolved atmospheric flow field

Bibliography

[1] Hüseyin Can Önel. Numerical simulations of wind turbine wake interactions using actuator line and les models. Master's thesis, Middle East Technical University, 2015. Accessed online via ResearchGate, Mar. 12, 2026.

[2] Alexandra Ministeru. Wave effect mitigation on floating offshore wind turbines through feedforward data-driven predictive control. Master's thesis, Delft University of Technology, 2024.

[3] Yue Niu, Parth Prashant Lathi, and Ryozo Nagamune. Floating Offshore Wind Farm Control via Turbine Repositioning with Aerodynamic Force*. In *2023 American Control Conference (ACC)*, pages 2542–2547, May 2023.

[4] Maarten J. Van Den Broek, Delphine De Tavernier, Benjamin Sanderse, and Jan-Willem Van Wingerden. Adjoint optimisation for wind farm flow control with a free-vortex wake model. *Renewable Energy*, 201:752–765, December 2022.

[5] Aemilius A. W. Van Vondelen, Marion Coquelet, Sachin T. Navalkar, and Jan-Willem Van Wingerden. Synchronized Helix wake mixing control. *Wind Energy Science*, 10(10):2411–2433, October 2025.

[6] S. Aubrun, M. Bastankhah, R.B. Cal, B. Conan, R.J. Hearst, D. Hoek, M. Hölling, M. Huang, C Hur, B. Karlsen, I. Neunaber, M. Obligado, J. Peinke, M. Percin, L. Saetran, P Schito, B. Schliffke, D. Sims-Williams, O. Uzol, M.K. Vinnes, and A. Zasso. Round-robin tests of porous disc models. *Journal of Physics: Conference Series*, 1256(1):012004, July 2019.

[7] Elizabeth H. Camp and Raúl Bayoán Cal. Mean kinetic energy transport and event classification in a model wind turbine array versus an array of porous disks: Energy budget and octant analysis. *Physical Review Fluids*, 1(4):044404, August 2016.

[8] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical Report NREL/TP-500-38060, 947422, February 2009.

[9] Maarten J. van den Broek. Free-vortex wake. http://github.com/TUDelft-DataDrivenControl/vortexwake, 2022.

[10] Leendert Starink, Zhaochen Xie, Marcus Becker, and Jan-Willem van Wingerden. Optimal (Dynamic) Turbine Repositioning Strategies for a Floating Wind Farm Depending on Mooring Line Stiffness. *Journal of Physics: Conference Series*, 3131(1):Article ID 012039.

[11] United Nations Framework Convention on Climate Change. Paris agreement. https://unfccc.int/files/essential_background/convention/application/pdf/english_paris_agreement.pdf, 2015. Accessed: 2026-03-12.

[12] NedZero. How to deal with wake effects and wake theft, n.d.

[13] Yue Niu, Arpit Dwivedi, Joel Sathiaraj, Parth P. Lathi, and Ryozo Nagamune. Floating Offshore Wind Farm Control via Turbine Repositioning: Unlocking the Potential Unique to Floating Offshore Wind. *IEEE Control Systems*, 44(5):106–129, October 2024.

[14] Mahdi Abkar, Navid Zehtabiyan-Rezaie, and Alexandros Iosifidis. Reinforcement learning for wind-farm flow control: Current state and future actions. *Theoretical and Applied Mechanics Letters*, 13(6):100475, November 2023.

[15] Ali C. Kheirabadi and Ryozo 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, September 2019.

[16] World Wind Energy Association. Global statistics: Wwea annual report 2025 – global wind power in 2025: Record growth and emerging challenges. https://wwindea.org/GlobalStatistics, 2026. Published April 1, 2026; preliminary statistics for 2025. Accessed 2026-07-24.

[17] International Renewable Energy Agency. Wind energy. https://www.irena.org/Energy-Transition/Technology/Wind-energy, 2026. Accessed 2026-07-24.

[18] NedZero. How to deal with wake effects and 'wake theft'. https://nedzero.nl/en/news/how-to-deal-with-wake-effects-and-wake-theft, 2026. News article; quotes Prof. Rebecca Barthelmie (Cornell University) and Prof. Carlos Simão Ferreira (TU Delft). Accessed 2026-07-24.

[19] Paul Fleming, Pieter M.O. Gebraad, Sang Lee, Jan-Willem Van Wingerden, Kathryn Johnson, Matt Churchfield, John Michalakes, Philippe Spalart, and Patrick Moriarty. Simulation comparison of wake mitigation control strategies for a two-turbine case: Simulation comparison of wake mitigation control strategies for a two-turbine case. *Wind Energy*, 18(12):2135–2143, December 2015.

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

[21] Zamiyad Dar, Koushik Kar, Onkar Sahni, and Joe H Chow. Optimization of Windfarm Power by Controlling the Yaw Angle using Dynamic Programming.

[22] P Fleming, M Churchfield, A Scholbrock, A Clifton, S Schreck, K Johnson, A Wright, P Gebraad, J Annoni, B Naughton, J Berg, T Herges, J White, T Mikkelsen, M Sjöholm, and N Angelou. Detailed field test of yaw-based wake steering. *Journal of Physics: Conference Series*, 753:052003, September 2016.

[23] M. Steinbuch, W.W. De Boer, O.H. Bosgra, S.A.W.M. Peters, and J. Ploeg. Optimal control of wind power plants. *Journal of Wind Engineering and Industrial Aerodynamics*, 27(1-3):237–246, January 1988.

[24] Jonas Gutknecht, Anne Van Den Homberg, Jelle Linke, Jeroen Van Der Marel, Jesper Van Der Meulen, Raul Hendriks, Axelle Viré, and Jan-Willem Van Wingerden. Synergizing Helix Active Wake Mixing with Dynamic Yawing: An Exploration Study using Porous Discs in a Wind Tunnel. *Journal of Physics: Conference Series*, 3016(1):012014, May 2025.

[25] Brian De Vos, Bjorn M. Harder, Thomas A. Huisman, Jonas Gutknecht, and Jan-Willem Van Wingerden. Mimicking the Helix with a porous disc for wind tunnel testing. *Journal of Physics: Conference Series*, 2767(9):092063, June 2024.

[26] Mingyang Mei, Peng Kou, Yilin Xu, Zhihao Zhang, Runze Tian, and Deliang Liang. Improving Floating Offshore Wind Farm Flow Control With Scalable Model-Based Deep Reinforcement Learning. *IEEE Transactions on Automation Science and Engineering*, 22:18255–18268, 2025.

[27] Bart M. Doekemeijer, Daan Van Der Hoek, and Jan-Willem Van Wingerden. Closed-loop model-based wind farm control using FLORIS under time-varying inflow conditions. *Renewable Energy*, 156:719–730, August 2020.

[28] Paul Fleming, Jennifer King, Christopher J. Bay, Eric Simley, Rafael Mudafort, Nicholas Hamilton, Alayna Farrell, and Luis Martinez-Tossas. Overview of FLORIS updates. *Journal of Physics: Conference Series*, 1618(2):022028, September 2020.

[29] S.F. Rodrigues, R. Teixeira Pinto, M. Soleimanzadeh, Peter A.N. Bosman, and P. Bauer. Wake losses optimization of offshore wind farms with moveable floating wind turbines. *Energy Conversion and Management*, 89:933–941, January 2015.

[30] Chenlu Han and Ryozo Nagamune. Position control of an offshore wind turbine with a semi-submersible floating platform using the aerodynamic force. In *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, May 2016.

[31] Ali C. Kheirabadi and Ryozo Nagamune. Real-time relocation of floating offshore wind turbine platforms for wind farm efficiency maximization: An assessment of feasibility and steady-state potential. *Ocean Engineering*, 208:107445, July 2020.

[32] Gabrielle Froese, Shan Yu Ku, Ali C. Kheirabadi, and Ryozo Nagamune. Optimal layout design of floating offshore wind farms. *Renewable Energy*, 190:94–102, May 2022.

[33] Chaoneng Huang, Li Wang, Qian Huang, Dongran Song, Jian Yang, Mi Dong, Young Hoo Joo, and Neven Duić. Bi-level multi-objective optimization framework for wake escape in floating offshore wind farm. *Applied Energy*, 377:124712, January 2025.

[34] Shangshang Wei, Zhihan Li, Xin Wang, Dachuan Feng, and Xianhua Gao. Multi-object reposition control of floating wind farms considering time-varying change of wind. *Ocean Engineering*, 327:120974, May 2025.

[35] Ali C. Kheirabadi and Ryozo Nagamune. Modeling and Power Optimization of Floating Offshore Wind Farms with Yaw and Induction-based Turbine Repositioning. In *2019 American Control Conference (ACC)*, pages 5458–5463, July 2019.

[36] Yuksel R. Alkarem, Kimberly Huguenard, Amrit S. Verma, Diederik Van Binsbergen, Erin Bachynski-Polić, and Amir R. Nejad. Passive Mooring-based Turbine Repositioning Technique for Wake Steering in Floating Offshore Wind Farms. *Journal of Physics: Conference Series*, 2767(9):092056, June 2024.

[37] Mandar Phadnis and Lucy Pao. Individual Pitch Control for Lateral Motion Control of Floating Offshore Wind Turbines *. In *2025 American Control Conference (ACC)*, pages 1515–1520. 2025 American Control Conference (ACC), July 2025.

[38] Chenlu Han and Ryozo Nagamune. Platform position control of floating wind turbines using aerodynamic force. *Renewable Energy*, 151:896–907, May 2020.

[39] Ali C. Kheirabadi and Ryozo Nagamune. Real-time Relocation of Floating Offshore Wind Turbines for Power Maximization Using Distributed Economic Model Predictive Control. In *2021 American Control Conference (ACC)*, pages 3077–3081, May 2021.

[40] Yuan Gao, Aravindh Padmanabhan, Ouyang Chen, Ali C. Kheirabadi, and Ryozo Nagamune. A Baseline Repositioning Controller for a Floating Offshore Wind Farm. In *2022 American Control Conference (ACC)*, pages 4224–4229, June 2022.

[41] Timothé Jard and Reda Snaiki. Real-Time Repositioning of Floating Wind Turbines Using Model Predictive Control for Position and Power Regulation. *Wind*, 3(2):131–150, March 2023.

[42] Timothé Jard and Reda Snaiki. Real-time dynamic layout optimization for floating offshore wind farm control. *Ocean Engineering*, 316:119971, January 2025.

[43] Leendert Starink, Sebastiaan Mulders, Joseph Saverin, and Jan-Willem van Wingerden. Data-driven phase synchronization for dynamic repositioning of floating wind turbines. *To appear in the proceedings of the 2026 Conference on Control Technology and Applications*, 2026. Delft Center for Systems and Control, TU Delft; Hermann-Föttinger Institute, TU Berlin.

[44] Marcus Becker, Bastian Ritter, Bart Doekemeijer, Daan Van Der Hoek, Ulrich Konigorski, Dries Allaerts, and Jan-Willem Van Wingerden. The revised FLORIDyn model: Implementation of heterogeneous flow and the Gaussian wake. *Wind Energy Science*, 7(6):2163–2179, November 2022.

[45] M Becker, D Allaerts, and J W Van Wingerden. FLORIDyn - A dynamic and flexible framework for real-time wind farm control. *Journal of Physics: Conference Series*, 2265(3):032103, May 2022.

[46] David Marten. *QBlade: A Modern Tool for the Aeroelastic Simulation of Wind Turbines*. PhD thesis, Technische Universität Berlin, 2019.

[47] Michael F. Howland, Juliaan Bossuyt, Luis A. Martínez-Tossas, Johan Meyers, and Charles Meneveau. Wake structure in actuator disk models of wind turbines in yaw under uniform inflow conditions. *Journal of Renewable and Sustainable Energy*, 8(4):043301, July 2016.

[48] S. Boersma, P.M.O. Gebraad, M. Vali, B.M. Doekemeijer, and J.W. Van Wingerden. A control-oriented dynamic wind farm flow model: "WFSim". *Journal of Physics: Conference Series*, 753:032005, September 2016.

[49] Matthew J. Churchfield, Sang Lee, Patrick J. Moriarty, Luis A. Martínez, Stefano Leonardi, Ganesh Vijayakumar, and James G. Brasseur. A large-eddy simulation of wind-plant aerodynamics. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, number AIAA 2012-0537, Nashville, Tennessee, 2012.

[50] Björn Maronga, Micha Gryschka, Rieke Heinze, Fabian Hoffmann, Farah Kanani-Sühring, Martin Keck, Klaus Ketelsen, Marcus O. Letzel, Matthias Sühring, and Siegfried Raasch. The parallelized large-eddy simulation model (PALM) version 4.0 for atmospheric and oceanic flows: Model formulation, recent developments, and future perspectives. *Geoscientific Model Development*, 8(8):2515–2551, 2015.

[51] Jess A. Michelsen. Block structured multigrid solution of 2D and 3D elliptic PDE's. Technical Report AFM 94-06, Technical University of Denmark, 1994.

[52] Niels N. Sørensen. *General Purpose Flow Solver Applied to Flow over Hills*. PhD thesis, Risø National Laboratory, Roskilde, Denmark, 1995. Risø-R-827-(EN).

[53] John M. Prospathopoulos, Evangelos S. Politis, Konstantinos G. Rados, and Panagiotis K. Chaviaropoulos. Evaluation of the effects of turbulence model enhancements on wind turbine wake predictions. *Wind Energy*, 14(2):285–300, 2011.

[54] Niels Otto Jensen. A note on wind generator interaction. Technical Report Risø-M-2411, Risø National Laboratory, Roskilde, Denmark, 1983.

[55] Mads M. Pedersen, Alexander Meyer Forsting, Paul van der Laan, Riccardo Riva, Leonardo A. Alcayaga Romàn, Javier Criado Risco, Mikkel Friis-Møller, Julian Quick, Jens Peter Schøler Christiansen, Rafael Valotta Rodrigues, Bjarke Tobias Olsen, and Pierre-Elouan Réthoré. PyWake 2.5.0: An open-source wind farm simulation tool, February 2023.

[56] Marcus Becker, Maxime Lejeune, Philippe Chatelain, Dries Allaerts, Rafael Mudafort, and Jan-Willem van Wingerden. A dynamic open-source model to investigate wake dynamics in response to wind farm flow control strategies. *Wind Energy Science*, 10(6):1055–1075, 2025.

[57] Ryan King. WindSE: Wind systems engineering [SWR-17-03], 2017. OSTI ID: code-5441.

[58] Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction.

[59] Shengbo Eben Li. *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer Nature Singapore, Singapore, 2023.

[60] Shukai He, Hangyu Wang, Jie Yan, Cheng Tao, Yongqian Liu, and Shuang Han. A Low-Computational Physics-Guided Deep Learning Model for Wind Farm Flow Control Under Time-Varying Wind Conditions, 2024.

[61] Hongyang Dong and Xiaowei Zhao. Composite Experience Replay-Based Deep Reinforcement Learning With Application in Wind Farm Control. *IEEE Transactions on Control Systems Technology*, 30(3):1281–1295, May 2022.

[62] Maarten J. van den Broek, Marcus Becker, Benjamin Sanderse, and Jan-Willem van Wingerden. Dynamic wind farm flow control using free-vortex wake models. *Wind Energy Science*, 9(3):721–740, 2024.

[63] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017.

[64] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2015.

[65] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[66] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[67] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

[68] Claire Bizon Monroc, Ana Bušić, Donatien Dubuc, and Jiamin Zhu. Actor Critic Agents for Wind Farm Control. In *2023 American Control Conference (ACC)*, pages 177–183, May 2023.

[69] Claire Bizon Monroc, Ana Bušić, Donatien Dubuc, and Jiamin Zhu. WFCRL: A Multi-Agent Reinforcement Learning Benchmark for Wind Farm Control, January 2025.

[70] C B Monroc, A Bušić, D Dubuc, and J Zhu. Towards fine tuning wake steering policies in the field: An imitation-based approach. *Journal of Physics: Conference Series*, 2767(3):032017, June 2024.

[71] Elie Kadoche. *Deep Reinforcement Learning for Wind Farm Flow Control*. PhD thesis, Institut Polytechnique de Paris, 2025.

[72] Elie Kadoche, Pascal Bianchi, Florence Carton, Philippe Ciblat, and Damien Ernst. Helping mitigate climate change through efficient reinforcement learning-based wind farm flow control.

[73] Elie Kadoche, Pascal Bianchi, Florence Carton, Philippe Ciblat, and Damien Ernst. How to craft a deep reinforcement learning policy for wind farm flow control, August 2025.

[74] Elie Kadoche, Sébastien Gourvénec, Maxime Pallud, and Tanguy Levent. MARLYC: Multi-Agent Reinforcement Learning Yaw Control. *Renewable Energy*, 217:119129, November 2023.

[75] Hongyang Dong, Jincheng Zhang, and Xiaowei Zhao. Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations. *Applied Energy*, 292:116928, June 2021.

[76] Henry Korb, Henrik Asmuth, Merten Stender, and Stefan Ivanell. Exploring the application of reinforcement learning to wind farm control. *Journal of Physics: Conference Series*, 1934(1):012022, May 2021.

[77] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

[78] Daan van der Hoek, Baptiste Van den Abbeele, Carlos Simão Ferreira, and Jan-Willem van Wingerden. Maximizing wind farm power output with the helix approach: Experimental validation and wake analysis using tomographic particle image velocimetry. *Wind Energy*, 27(5):463–482, 2024.

[79] WindShape SA. Windshaper. https://www.windshape.com/products/windshaper. Accessed: 2026-07-11.

[80] S. Aubrun, S. Loyer, P.E. Hancock, and P. Hayden. Wind turbine wake properties: Comparison between a non-rotating simplified wind turbine model and a rotating model. *Journal of Wind Engineering and Industrial Aerodynamics*, 120:1–8, September 2013.

[81] Mohammad Hassan Ranjbar, Hadi Zanganeh Kia, Seyyed Abolfazl Nasrazadani, Kobra Gharali, and Jatin Nathwani. Experimental and numerical investigations of actuator disks for wind turbines. *Energy Science & Engineering*, 8(7):2371–2386, 2020.

[82] F. V. Mühle, S. Tamaro, F. Klinger, F. Campagnolo, and C. L. Bottasso. Experimental and numerical investigation on the potential of wake mixing by dynamic yaw for wind farm power optimization. *Journal of Physics: Conference Series*, 2767(9):092068, 2024.

[83] Shantanu Purohit, Haoyuan Sun, Andrea Sciacchitano, and Wei Yu. Experimental investigation of dynamically yawed porous discs for wind turbine yaw control. *Journal of Physics: Conference Series*, 3224(3):032067, 2026. The Science of Making Torque from Wind (TORQUE 2026).

[84] Jan Bartl, Franz Mühle, Jannik Schottler, Lars Sætran, Joachim Peinke, Muyiwa Adaramola, and Michael Hölling. Wind tunnel experiments on wind turbine wakes in yaw: effects of inflow turbulence and shear. *Wind Energy Science*, 3(1):329–343, 2018.

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

[86] Aitor Saenz-Aguirre, Ekaitz Zulueta, Unai Fernandez-Gamiz, Javier Lozano, and Jose Manuel Lopez-Guede. Artificial Neural Network Based Reinforcement Learning for Wind Turbine Yaw Control. *Energies*, 12(3):436, January 2019.

[87] Timothy Verstraeten, Pieter JK Libin, and Ann Nowé. Fleet Control using Coregionalized Gaussian Process Policy Iteration, November 2019.

[88] Van-Hai Bui, Thai-Thanh Nguyen, and Hak-Man Kim. Distributed Operation of Wind Farm for Maximizing Output Power: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE Access*, 8:173136–173146, 2020.

[89] Aitor Saenz-Aguirre, Ekaitz Zulueta, Unai Fernandez-Gamiz, Alain Ulazia, and Daniel Teso-Fz-Betono. Performance enhancement of the artificial neural network–based reinforcement learning for wind turbine yaw control. *Wind Energy*, 23(3):676–690, March 2020.

[90] Paul Stanfel, Kathryn Johnson, Christopher J. Bay, and Jennifer King. A Distributed Reinforcement Learning Yaw Control Approach for Wind Farm Energy Capture Maximization. In *2020 American Control Conference (ACC)*, pages 4065–4070, July 2020.

[91] Timothy Verstraeten, Eugenio Bargiacchi, Pieter J. K. Libin, Jan Helsen, Diederik M. Roijers, and Ann Nowé. Multi-Agent Thompson Sampling for Bandit Applications with Sparse Neighbourhood Structures. *Scientific Reports*, 10(1):6728, April 2020.

[92] Zhiwei Xu, Hua Geng, Bing Chu, Menghao Qian, and Ni Tan. Model-Free Optimization Scheme for Efficiency Improvement of Wind Farm Using Decentralized Reinforcement Learning. *IFAC-PapersOnLine*, 53(2):12103–12108, 2020.

[93] Huan Zhao, Junhua Zhao, Jing Qiu, Gaoqi Liang, and Zhao Yang Dong. Cooperative Wind Farm Control With Deep Reinforcement Learning and Knowledge-Assisted Learning. *IEEE Transactions on Industrial Informatics*, 16(11):6912–6921, November 2020.

[94] P. Stanfel, K. Johnson, C. J. Bay, and J. King. Proof-of-concept of a reinforcement learning framework for wind farm energy capture maximization in time-varying wind. *Journal of Renewable and Sustainable Energy*, 13(4):043305, July 2021.

[95] Timothy Verstraeten. *A Multi-Agent Reinforcement Learning Approach to Wind Farm Control*. PhD thesis, Vrije Universiteit Brussel, 2021.

[96] Hongyang Dong and Xiaowei Zhao. Intelligent Wind Farm Control via Grouping-Based Reinforcement Learning. In *2022 European Control Conference (ECC)*, pages 993–998, July 2022.

[97] Hongyang Dong and Xiaowei Zhao. Wind-Farm Power Tracking Via Preview-Based Robust Reinforcement Learning. *IEEE Transactions on Industrial Informatics*, 18(3):1706–1715, March 2022.

[98] Borja Fernandez-Gauna, Manuel Graña, Juan-Luis Osa-Amilibia, and Xabier Larrucea. Actor-critic continuous state reinforcement learning for wind-turbine control robust optimization. *Information Sciences*, 591:365–380, April 2022.

[99] Binghao He, Huan Zhao, Gaoqi Liang, Junhua Zhao, Jing Qiu, and Zhao Yang Dong. Ensemble-based Deep Reinforcement Learning for robust cooperative wind farm control. *International Journal of Electrical Power & Energy Systems*, 143:108406, December 2022.

[100] Rui Li, Jincheng Zhang, and Xiaowei Zhao. Dynamic wind farm wake modeling based on a Bilateral Convolutional Neural Network and high-fidelity LES data. *Energy*, 258:124845, November 2022.

[101] Claire Bizon Monroc, Eva Bouba, Ana Bušić, Donatien Dubuc, and Jiamin Zhu. Delay-Aware Decentralized Q-learning for Wind Farm Control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 807–813, December 2022.

[102] Grigory Neustroev, Sytze P Andringa, Remco A Verzijlbergh, and Mathijs M De Weerdt. Deep Reinforcement Learning for Active Wake Control. 2022.

[103] Venkata Ramakrishna Padullaparthi, Srinarayana Nagarathinam, Arunchandar Vasan, Vishnu Menon, and Depak Sudarsanam. FALCON- FArm Level CONtrol for wind turbines using multi-agent deep reinforcement learning. *Renewable Energy*, 181:445–456, January 2022.

[104] Zhiwen Deng, Chang Xu, Xingxing Han, Zhe Cheng, and Feifei Xue. Decentralized yaw optimization for maximizing wind farm production based on deep reinforcement learning. *Energy Conversion and Management*, 286:117031, June 2023.

[105] Shenglin Peng and Qianmei Feng. Data-driven optimal control of wind turbines using reinforcement learning with function approximation. *Computers & Industrial Engineering*, 176:108934, February 2023.

[106] Vahid Tavakol Aghaei, Arda Ağababaoğlu, Biram Bawo, Peiman Naseradinmousavi, Sinan Yıldırım, Serhat Yeşilyurt, and Ahmet Onat. Energy optimization of wind turbines via a neural control policy based on reinforcement learning Markov chain Monte Carlo algorithm. *Applied Energy*, 341:121108, July 2023.

[107] Jingjie Xie, Hongyang Dong, and Xiaowei Zhao. Data-driven torque and pitch control of wind turbines via reinforcement learning. *Renewable Energy*, 215:118893, October 2023.

[108] Victor Espinoza, Carolina Ormaza, Yolanda Vidal, and Christian Tutivén. Optimizing Collective Pitch Control in Wind Turbines Using Deep Reinforcement Learning and OpenFAST Simulation. *IFAC-PapersOnLine*, 59(9):151–156, 2025.

[109] Tenghui Li, Jin Yang, and Anastasia Ioannou. Data-driven control of wind turbine under online power strategy via deep learning and reinforcement learning. *Renewable Energy*, 234:121265, November 2024.

[110] Jaime Liew, Tuhfe Göçmen, Wai Hou Lio, and Gunner Chr. Larsen. Model-free closed-loop wind farm control using reinforcement learning with recursive least squares. *Wind Energy*, 27(11):1173–1187, 2024.

[111] Mahmood Mazare. Adaptive optimal secure wind power generation control for variable speed wind turbine systems via reinforcement learning. *Applied Energy*, 353:122034, January 2024.

[112] Abdelhamid Nabeel, Ahmed Lasheen, Abdel Latif Elshafei, and Essam Aboul Zahab. Fuzzy-based collective pitch control for wind turbine via deep reinforcement learning. *ISA Transactions*, 148:307–325, May 2024.

[113] Longyan Wang, Qiang Dong, Yanxia Fu, Bowen Zhang, Meng Chen, Junhang Xie, Jian Xu, and Zhaohui Luo. Effectiveness of cooperative yaw control based on reinforcement learning for in-line multiple wind turbines. *Control Engineering Practice*, 153:106124, December 2024.

[114] Andrew Mole, Max Weissenbacher, Georgios Rigas, and Sylvain Laizet. Reinforcement Learning Increases Wind Farm Power Production by Enabling Closed-Loop Collaborative Control, June 2025.

[115] Jiawang Yuan and Shan He. Reinforcement learning-enhanced neural network sliding mode control for unified pitch regulation of large-scale wind turbines. *Ocean Engineering*, 339:122020, November 2025.

[116] Víctor Espinoza, Carolina Ormaza, Christian Tutivén, and Yolanda Vidal. Reinforcement learning-based pitch control for wind turbines using double deep Q-networks. *ISA Transactions*, 169:428–435, February 2026.

AI Declaration

I acknowledge the use of Claude (https://claude.ai/) to support the following activities: assisting with debugging of code, error-checking equations, and improving figure, plot and report graphics aesthetics.

Glossary

List of Acronyms

3mE	Mechanical, Maritime and Materials Engineering
AMS	American Mathematical Society
DCSC	Delft Center for Systems and Control
TU Delft	Delft University of Technology
DRL	Deep Reinforcement Learning
FFT	Fast Fourier Transform
FOWT	Floating Offshore Wind Turbine
FVW	Free Vortex Wake
FVWM	Free Vortex Wake Model
GAE	Generalised Advantage Estimation
HAWT	Horizontal Axis Wind Turbine
HPC	High Performance Computing
IPC	Individual Pitch Control
LQI	Linear Quadratic Integral
LQR	Linear Quadratic Regulator
LUT	Look-Up Table
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
MPC	Model Predictive Control
NREL	National Renewable Energy Laboratory
PMMA	Polymethyl Methacrylate (colloquially known as perspex)
POMDP	Partially Observable Markov Decision Process

PPO Proximal Policy Optimisation

RL Reinforcement Learning

TRPO Trust Region Policy Optimisation