

Data-driven closed-loop control of a WindShaper wind fan array

Shane van Rhijn

Master of Science Thesis

Data-driven closed-loop control of a WindShaper wind fan array

MASTER OF SCIENCE THESIS

For the degree of Master of Science at Delft University of Technology

Shane van Rhijn

August 2026



DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL

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

DATA-DRIVEN CLOSED-LOOP CONTROL OF A WINDSHAPER WIND FAN ARRAY

by

SHANE VAN RHIJN

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Dated: August 2026

Supervisor(s):

J. W. van Wingerden

T. J. Crumpton

Reader(s):

Abstract

Wind tunnel testing is a fundamental part in wind energy control validation. Despite that, facilities around the globe used for wind energy control experiments offer limited capability in generating spatially heterogeneous and non-stationary inflow conditions in a controlled and repeatable manner. The WindShaper facility in the WindAI Lab at the Delft Center for Systems and Control (DCSC) provides independent control of a 729 fan-array and therefore offers the potential for spatial and temporal flow control. This thesis investigates to what extent the WindShaper facility is able to generate prescribed heterogeneous and non-stationary reference profiles.

The high-dimensional actuator space was first reduced to three spatial input coefficients, from which a static input-output model was identified. The static map captured the general input-output behaviour well but was not accurate enough for open-loop tracking. Fixed-input experiments showed that both stationary and non-stationary flow remained approximately temporally stable over a duration of 1800 s, suggesting that within-run drift is unlikely to be a dominant source of the observed tracking errors. Fixed-model feedback experiments showed that the identified static map contained sufficient sensitivity information for accurate reference tracking, while periodic tracking experiments showed the relevance of temporal dynamics.

To account for sensitivity to model uncertainty, the static input-output model was updated online through several adaptive least-squares approaches and combined with a regularized inverse control law. The recursive least-squares approach showed the strongest tracking improvement. Under nominal conditions, the recursive least-squares algorithm saw a tracking improvement of 33.2 % relative to the open-loop controller and of 57.5 % under enforced model mismatch conditions. Under enforced model mismatch conditions, ten repeated recursive least-squares experiments were found to be statistically consistent with the repeatability limit estimated from the data. Under stricter and more relevant repeatability conditions repeatability coverage was found to depend on operating conditions and no universal repeatability was observed.

The results show that the WindShaper facility is able to generate repeatable prescribed mean-velocity profiles under the investigated operating conditions and that online model adaptation significantly improved robustness to model uncertainty. The achievable run-to-run repeatability depends on the operating conditions and the specific criterion used to assess it.

Acknowledgements

I would like to thank Prof J.W. van Wingerden for his supervision of this thesis. I would also like to thank T.J. Crumpton for all the help he has offered me along the way.

I would also like to thank R.J.M Willekes and W.J.M van Geest at the DCSC Lab for helping me setup my experiments and allowing me to use the lab.

AI Statement: During this research I made use of generative AI tools. I used it to help me understand certain concepts, assist me in writing code and rewrite certain passages. I remain fully responsible for the work.

Table of Contents

1	Introduction	1
2	Experimental System	4
2-1	Experimental Facility	4
2-2	Control-Oriented Facility Characterization	6
2-2-1	Sensor Assessment	9
2-3	Control Problem Formulation	10
2-4	Static Map	13
3	Within-run Temporal Stability	17
3-1	Stationary Input Sequences	19
3-2	Periodic Input Sequences	21
4	Control Architectures	25
4-1	Fixed-Model Feedback Control	25
4-2	Online Model Adaptation and Inverse Control	27
5	Feedback Control	29
5-1	Online Feedback Optimization	29
5-2	Dynamic Model	31
5-3	Final Remarks	34
6	Online Model Estimation and Inverse Control	36
6-1	Adaptive Least-Squares	37
6-1-1	Cumulative least-squares	38
6-1-2	Batch-wise least-squares	39
6-1-3	Moving-window least-squares	39
6-1-4	Recursive Least Squares	40

6-1-5	Expected performance and tuning trade-off	41
6-2	Simulations	42
6-2-1	Can online least-squares damage performance when the model is already good?	42
6-2-2	Can online least-squares improve tracking when the model is wrong? . . .	43
6-2-3	What does the dataset size trade-off look like?	45
6-2-4	What is the effect of the regularisation terms?	46
6-2-5	Takeaways for experimental validation	48
6-3	Experimental Validation Setup	49
7	Experimental Validation	52
7-1	Nominal Configuration	52
7-2	Model Mismatch	55
7-3	Closed-Loop Recursive Least-Squares Repeatability Assessment	58
8	Conclusion	64
	Appendices	68
A	Alternative Flow Measurement Techniques	69
B	Facility Characterization	73
B-1	Spatial Non-Uniformities	75
C	Static-Model Identification Inputs	79
D	Within Run Temporal Stability	80
D-1	Temperature Evolution	80
D-2	Stationary Input Sequences	82
D-3	Periodic Input Sequences	84
E	Least-Squares Simulation Details	87
E-1	Simulation Settings	87

List of Figures

2-1	Schematic overview of the WindShaper Facility	5
2-2	Experimental sensor configuration used in the WindAI Lab	6
2-3	Flow characterisation of the experimental facility under different operating conditions	8
2-4	Measured velocity response at the three sensor locations to repeated uniform WindShaper step commands. Each input is applied twice, providing an initial indication of short-term within-run consistency and revealing persistent differences between the measured sensor outputs	10
2-5	WindShaper GUI and normalized coordinate system	12
2-6	Comparison of measured and static model-predicted mean velocities at the three sensor locations for the training and validation datasets.	16
3-1	Drift characteristics for the stationary profiles	19
3-2	Linear drift models fitted to the 0.75 s averaged velocity measurements for the $\alpha_0 = 35$ uniform input experiment after discarding the first 60 s	20
3-3	The estimated temporal drift rates of the period wise fitted mean level A , amplitude R and unwrapped phase ϕ for all periodic input experiments.	23
3-4	Period-wise drift rates of the fitted mean level A , amplitude R and unwrapped phase ϕ for sensor 2 under the $T_s = 30$ s periodic input. Solid lines show the fitted linear temporal trends after removal of the first 60 s.	24
4-1	General optimization-based feedback control architecture for the WindShaper facility. The controller uses the measured flow output and the identified static map G to update the parametrized input online, while control inputs are restricted to the operational constraints of the admissible input set.	26
4-2	General model predictive feedback control architecture for the WindShaper facility. A fixed dynamic model is used to predict the system response and determine the control action over a receding horizon. The prediction error $\mathbf{y}_k - \hat{\mathbf{y}}_k$ is used to update the output-bias estimate $\hat{\mathbf{b}}_k$, allowing part of the plant-model mismatch to be compensated without adapting the identified dynamic sensitivities.	27
4-3	Online model adaptation and inverse-control architecture for the WindShaper facility. Measured input-output data are used to update the estimated static map $\hat{\mathbf{G}}_k$ and offset $\hat{\mathbf{b}}_k$ online. The updated model is subsequently used to determine the control input required for tracking the prescribed mean-velocity reference.	28

5-1	Measured mean-velocity tracking of a homogeneous stationary reference using on-line feedback optimization with the fixed static input-output map.	30
5-2	Measured mean-velocity tracking of a heterogeneous stationary reference using online feedback optimization with the fixed static input-output map.	31
5-3	Measured tracking of a periodic reference with $T_s = 30$ s using online feedback optimization based on the fixed static input-output map.	32
5-4	Measured tracking of a sinusoidal reference with $T_s = 30$ s using the FIR-based model predictive controller with online output-bias adaptation.	34
6-1	Open-loop inverse control tracking using the fixed identified static model.	37
6-2	overall RMSE improvement I_{RMSE} of selected adaptive least-squares approaches relative to open-loop inverse control for the four simulation cases.	44
6-3	Overall RMSE versus dataset size for batch-wise and moving-window least-squares in the gain-and-bias-mismatch and slow-drift cases.	45
6-4	Median regressor conditioning plotted against dataset size	46
6-5	Reference tracking trajectories for the gain and bias mismatch simulations.	48
6-6	Sensor-rake configurations used for experimental validation.	50
7-1	Sensor-specific RMSE comparison for the open-loop inverse controller and adaptive least-squares approaches under the nominal configuration.	53
7-2	Ensemble-mean tracking trajectory of the RLS controller across five runs for the <i>NC</i> case.	55
7-3	RMSE comparison of the different control approaches under enforced model mismatch	56
7-4	Ensemble-mean tracking trajectory of the RLS controller across five runs for the <i>EMM</i> configuration.	58
7-5	Ten repeated runs for the RLS algorithm in the <i>EMM</i> configuration.	60
A-1	Example of a single-wire hot-wire anemometer used for pointwise velocity measurements	70
A-2	Standard (planar) 2C-2D particle image velocimetry	71
A-3	Oxford probe design	71
B-1	Comparison between stepwise and continuous planar measurements.	74
B-2	Spatial mean-velocity deviation for a uniform 40% WindShaper command for the bottom half of the downstream planes 2–5.	75
B-3	Spatial mean-velocity deviation for a uniform 40% WindShaper command for the top half of the downstream planes 2–5.	76
B-4	Spatial mean-velocity deviation for a uniform 60% WindShaper command for the bottom half of the downstream planes 2–5.	77
B-5	Spatial mean-velocity deviation for a uniform 60% WindShaper command for the top half of the downstream planes 2–5.	78
D-1	Measured ambient-temperature evolution during the ten stationary and periodic experiments used for the within-run temporal-stability assessment.	81
D-2	Residual autocorrelation after removal of the fitted linear drift for the stationary fixed-input experiments.	83

D-3	Residual autocorrelation of the fitted mean-level A drift models for the periodic input experiments.	84
D-4	Residual autocorrelation of the fitted amplitude R drift models for the periodic input experiments.	85
D-5	Residual autocorrelation of the fitted unwrapped-phase ϕ drift models for the periodic input experiments.	86

List of Tables

2-1	Mean velocity measurements for the sensor assessment	11
2-2	Performance metrics of the static affine model.	15
3-1	Overview of the ten experimental runs for the drift assessment	18
3-2	Mean fitted sinusoidal parameters for each excitation period and sensor	22
3-3	Largest absolute fit changes among all experimental non-stationary runs	23
6-1	Overview of the four different simulation cases	43
6-2	Evaluation settings and performance metrics used in the simulations	43
6-3	Simulation results for the nominal plant case, where the identified model parameters are equal to the true model parameters. For RLS, the conditioning is evaluated using the exponentially weighted regressor $\Phi_k \mathbf{W}_k^{1/2}$. Rank-deficient evaluations correspond to regressors with rank smaller than four.	44
6-4	Worst-case smallest-singular-value and rank diagnostics across the nominal, bias-mismatch, gain-and-bias-mismatch, and slow-drift simulations. The minimum median is the lowest case-wise median value of $\sigma_{\min}(\Phi)$, whereas the overall minimum is the smallest value observed at any evaluated model-update instant. For RLS, Φ denotes the exponentially weighted regressor $\Phi_k \mathbf{W}_k^{1/2}$	48
6-5	Algorithm-specific settings used in the experimental validation.	51
7-1	Tracking performance of the inverse controller and adaptive least-squares approaches for the nominal case. The first 10 s of each run are excluded to remove the initial transient.	53
7-2	Regressor and estimated-gain conditioning metrics for the nominal case	54
7-3	Tracking performance of the inverse controller and adaptive least-squares approaches for the <i>EMM</i> configuration. The first 10 s of each run are excluded to remove the initial transient.	56
7-4	Regressor and estimated-gain conditioning metrics for the <i>EMM</i> configuration.	57
7-5	ASTM repeatability results for the RLS algorithm.	61
7-6	Repeatability coverage of the RLS algorithm relative to the application-specific tolerance.	61

7-7	Repeatability coverage over four equal segments for the RLS algorithm.	62
A-1	Minimum inflow velocity associated with the recommended Reynolds-number range for different Oxford-probe diameters.	72
A-2	Comparison of the flow-measurement techniques considered for the WindShaper experiments in terms of temporal and spatial resolution, real-time capability, practical implementation and calibration requirements.	72
D-1	Drift characteristics for the stationary profiles	82
E-1	Numerical settings and implementation choices used for the adaptive least-squares simulations presented in Chapter 6.	88

Chapter 1

Introduction

If wind energy is to play a larger role in a low-carbon electricity system, further improvements in its performance, reliability, and cost-effectiveness are required. In this context, control plays an important role: for individual wind turbines common control objectives are load mitigation and maximizing power output, whereas for wind farms emphasis is often put on maximizing power output [1].

Bianchini et al. [2] mention that the actual power production of a wind farm is significantly lower than the nominal cumulative nameplate capacity of all the installed turbines in the farm. Although this difference cannot solely be accounted for by aerodynamic interactions between turbines, wake effects are an important source of wind farm power losses. In a wind farm, turbines are usually placed in the far wake of upstream turbines, which falls in the range of three to ten rotor diameters [3]. This leads to the wind downstream of a turbine having a lower incoming flow velocity, resulting in lower power production for downstream turbines compared to when they would operate in isolation [4, 5, 1, 6]. In addition to the occurring velocity deficit, this region is also characterized by an increase in turbulence intensity with respect to free-stream wind, which can lead to higher loads on downstream turbines [1]. Control strategies have therefore been implemented to influence these wakes and minimize these loads. Examples include Individual Pitch Control (IPC) [7, 1] for mitigating periodic loads, and Dynamic Induction Control (DIC) [8] and the Helix approach [9, 10] for actively manipulating wake development and recovery.

Validation of these different control strategies is generally done in what is known as a four step process [6]. These steps are: high-fidelity simulation, wind tunnel testing on scaled models, testing on research turbines and full scale field experiments. High-fidelity simulations are well suited for assessing controller behaviour under idealized operating conditions and full-scale testing provides us information on how turbines will behave under true operating conditions. Wind tunnel testing is generally done as an intermediate step and tests control concepts on scaled-down turbines under controlled inflow conditions [6]. Besides the controller performance, such experiments allow for the analysis of physical flow-turbine interactions.

True flow that is found in the Atmospheric Boundary Layer (ABL) can exhibit heterogeneity (spatially varying), non-stationary (time-varying) and stochastic fluctuations. Many conven-

tional wind tunnel facilities that are used for such strategies are able to produce selected characteristics of the ABL [11, 1]. However, these facilities are typically limited in their ability to produce controlled, heterogeneous, non-stationary and stochastic flow in a repeatable manner. As a result, validation of control strategies in wind tunnels under realistic complex inflow conditions, as seen in the ABL, remains challenging. For validation of control strategies under complex inflow conditions, it is therefore desirable to control both spatial and temporal variations in the inflow. Aside from these control capabilities, repeatability is an important requirement for controlled wind-tunnel experiments, since repeated tests should expose the investigated experiments to comparable inflow conditions. Previous active grid studies have therefore explicitly assessed the reproducibility of prescribed inflow by comparing repeated time histories and their spectral characteristics [1]. A facility with these capabilities is able to prescribe both temporal and spatial variations across a predefined test-section over the duration of an experiment and do so repeatedly over multiple experimental runs. In this research, however, repeatability is not treated as a control objective but more so as an experimental validation requirement.

The WindAI Lab at the Delft Center for Systems and Control contains an experimental setup with a WindShaper fan array. This WindShaper fan array offers a flexible way of generating flow since each of its 729 individual fans can be individually controlled. Therefore, the WindShaper potentially provides spatial and temporal control over the flow.

A challenge associated with the WindShaper is that due to its high input dimensionality, it is difficult to describe the system through a physical model [12]. Furthermore, since this research only uses three sensors, the system is highly over-actuated, meaning that many actuator combinations may lead to the same desired reference profile. The high-dimensionality and non-uniqueness of the problem may be accounted for through regularization or by reducing the actuator space through parametrization. Therefore, in this thesis, the actuator space is reduced to a low dimensional parametrized coefficient-space, from which a static input-output map to the mean measured velocities is identified. The static map approach has been introduced by Zavala et al. [12] and offers a simple and interpretable modelling approach. However, it is uncertain whether a static model identified under selected experimental conditions is sufficiently accurate for a broader operating range.

If the available static model does not represent the system with high enough accuracy for accurate reference tracking, this model mismatch must be compensated for. One way of doing so is by keeping a fixed model and applying feedback using the measured tracking error to tune the control input. Another approach is to use past measurements to adapt the input-output model online, after which the updated model is used to find the appropriate control input. These approaches motivated the control architectures that are to be implemented in this thesis.

Therefore, this thesis will investigate whether the WindShaper in the WindAI Lab at the Delft Center for Systems and Control can be controlled in such a way that it can generate repeatable, non-stationary and heterogeneous prescribed flow patterns which are controlled through the mean velocity under uncertainty in its input-output behaviour. This introduces the main research question.

RQ1 : *To what extent can the WindShaper facility in the WindAI Lab track prescribed heterogeneous and non-stationary mean velocity profiles under uncertainty in its input-output behaviour?*

Answering this main research question requires solving some challenges. First, the high dimensional input space requires a comprehensible representation of the relationship between the parametrized inputs and mean velocity outputs. Second, the temporal behaviour of the WindShaper must be investigated under fixed stationary and non-stationary input sequences. Finally, observed model inaccuracy needs to be compensated for through an appropriate control architecture. This leads to the following sub-questions.

SQ1 : *How can the relationship between WindShaper actuation and measured mean velocity be described in a suitable way for control?*

SQ2 : *How does the measured WindShaper response behave over time under fixed stationary and non-stationary input sequences?*

SQ3 : *What capabilities and limitations does feedback control provide for tracking prescribed mean velocity reference profiles under model uncertainty?*

SQ4 : *To what extent can online adaptation of the input-output model improve tracking accuracy under model uncertainty?*

The remainder of this report is structured as follows. Chapter 2 discusses the experimental facility and how the system will be modelled. Chapter 3 investigates the behaviour of the WindShaper by assessing within run temporal stability under fixed input sequences. Chapter 4 introduces the different control architectures that are applied in this research. Chapter 5 implements fixed-model feedback control approaches to the WindShaper. Chapter 6 implements online model adaptation in a simulation environment through a least-squares approach. Chapter 7 presents the results of the experimental validation of the adaptive least-squares approaches implemented to the WindShaper. Finally, the conclusion is presented in Chapter 8.

Experimental System

In this chapter, the experimental facility will be introduced and a static input-output model will be identified. This chapter is aimed at understanding the facility and investigating the static model. The findings presented in this chapter in combination with the drift assessment in Chapter 3 motivate the control architectures in Chapter 4.

2-1 Experimental Facility

A better understanding of the experimental facility is required to be able to formulate a control problem. This section will describe the experimental facility and all of the equipment needed in order to achieve the thesis objective of reproducing prescribed flow profiles. A general visualisation of the WindAI Lab is given in Figure 2-1 and its different parts of the WindAI Lab are described as follows:

- **WindShaper:** The WindShaper is the actuator of the experimental setup. It consists of 729 individual fans that each consist of two fan layers, which can be controlled as a whole or individually. Fan commands can be described globally or spatially, allowing for the production of different mean-flow variations across the fan array.
- **Wind Box:** The wind box forms the open-return experimental environment between actuation and measurements. Airflow generated by the WindShaper moves through the box before it reaches the measurement locations. Therefore the spatial and temporal behaviour of the flow can be seen by the controller at these locations. The fact that the box is open-return means that ambient conditions are not isolated and can therefore act as disturbances on the input-output model.
- **Robot Arm:** The robot arm determines the position of the measurements system within the wind box. It can be moved in the stream-wise, lateral and vertical directions. During experiments, the position is kept fixed, such that measured outputs correspond to specific spatial locations.

- **TriSonica:** Three TriSonica LI-560 sonic anemometers are used to measure the velocity of the incoming flow. Each have a maximum sampling frequency of 100 Hz. In the $0\text{--}10\text{ m s}^{-1}$ range the TriSonica LI-560 has an accuracy of $\pm 1\text{ m s}^{-1}$. The TriSonicas need to be placed at a 1 m distance from each other to ensure they do not interfere with each others measurements [13]. Some of the other sensors that have been considered for this research are described in Appendix A.
- **Sensor Rake:** The sensor rake has been designed and manufactured specifically for this purpose and determines the relative location of the sensors. It therefore defines the three measured outputs of the control problem. The sensor locations are denoted as T, R, L corresponding to top, right and left respectively when viewed downstream. The rake is denoted by number 1 in Figure 2-1. For a more detailed view of the rake and sensors see Figure 2-2.
- **WindShaper Computer:** The WindShaper Computer receives the real-time TriSonica measurements, averages them over a specified duration to obtain mean measurements, evaluates a control algorithm and sends the resulting commands to the WindShaper.

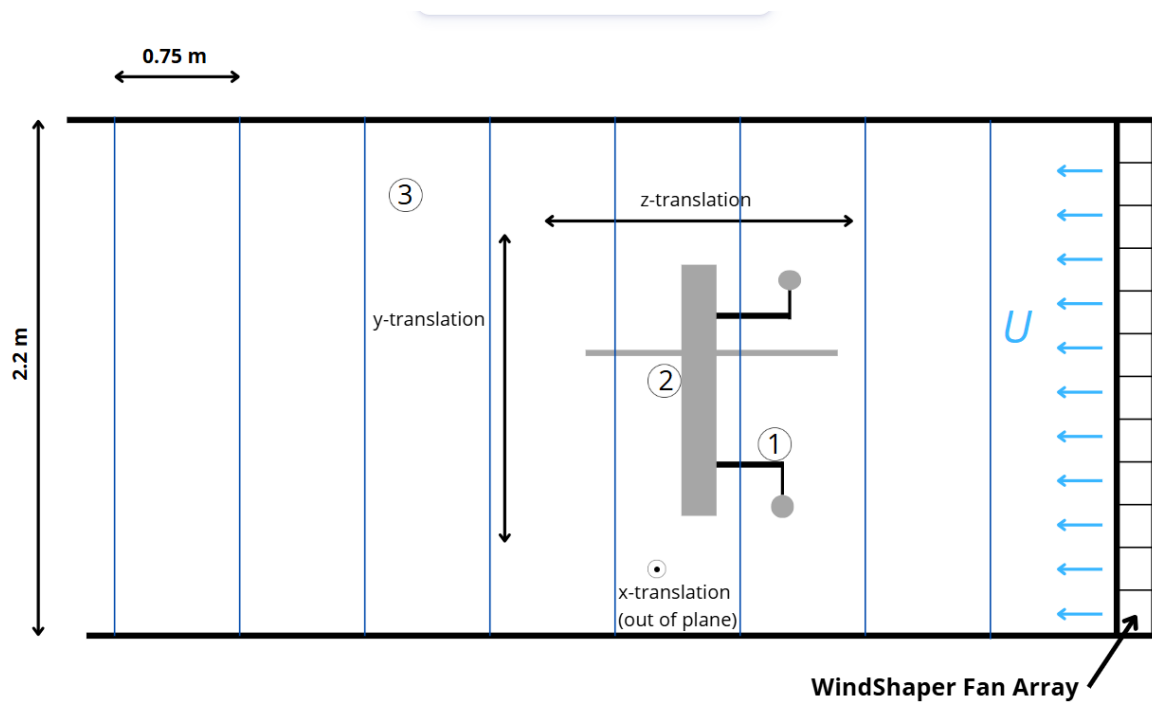
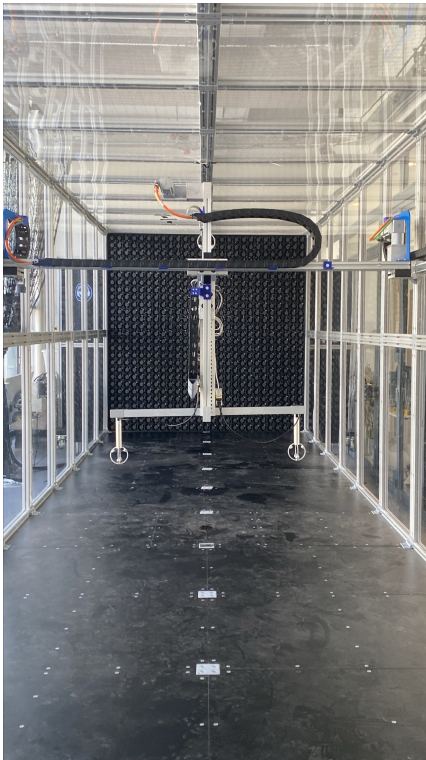


Figure 2-1: Schematic side view of the WindAI Lab experimental facility. The modular wind box consists of 0.75 m sections and has a height of 2.2 m. The robot arm can translate the sensor rake in the stream-wise, lateral and vertical directions; lateral motion is perpendicular to the illustrated plane. Flow generated by the WindShaper enters the wind box from the right. (1) is the sensor rake, (2) the robot arm and (3) the modular wind box.

These components together could form a closed-loop control architecture. The control algorithm generates a command, which is sent to the WindShaper through the WindShaper

computer. The resulting flow generated by the WindShaper then makes its way through the wind box until it is sensed by the three TriSonicas, which correspond to the controlled outputs. The measured output signals are sent back to the WindShaper computer which averages them to obtain mean velocity measurements which are fed back into the control algorithm.

The WindShaper can be commanded through its Graphical User Interface (GUI) or through a Python interface. The Python interface has been used for all the experiments done in this research. Although the WindShaper consists of 729 individually controllable fans, for this research this actuator space has been reduced to a lower-dimensional decision-space, as described in Section 2-3.



(a) Sensor Rake used to mount the three TriSonicas at fixed relative positions.



(b) Close-up of a LI-560 TriSonica mounted on the rake for real-time velocity measurements in the Wind Box.

Figure 2-2: Experimental sensor configuration used in the WindAI Lab.

2-2 Control-Oriented Facility Characterization

The purpose of this facility characterization is to obtain a better understanding of the flow quality and to establish a suitable downstream sensing plane for control of the flow. Owen and Owen [14] mention that poor wind flow quality can have significant effects on experimental testing. Flow irregularities can be found in the form of spatial non-uniformities, swirl, low-frequency unsteadiness and turbulence. These irregularities are not caused by a total circuit but more by the primary components of a wind tunnel such as the fan nacelle or wind tunnel

contractions. These irregularities can also be registered in the form of sensor noise caused by sensor vibrations for example.

This characterization can therefore be used to assess how the mean flow and turbulence intensity (TI) develop throughout the box, investigate spatial non-uniformities and select the most suitable plane for future closed-loop repeatability experiments. This is not to be mistaken with an aerodynamic characterization of the facility.

Flow characterization, in this context, consists of assessing the mean flow, local velocity fluctuations and spatial mean flow variations. The local velocity variations are defined with the TI and is defined as

$$TI = 100 \frac{\sigma_U}{\bar{U}} [\%] \quad (2-1)$$

where σ_U is the standard deviation of the velocity measurements and $\bar{U}$ the mean velocity of the measurements. Since coarse planar measurements were taken, however, and the measurements are subject to actuator limitations, this characterization merely acts as an indication of the quality of the flow field.

Measurements were done by taking a single TriSonica and traversing it through four stream-wise planes using the robot arm. The distance between the planes is 693 mm and from the WindShaper to plane 2 is approximately 1.5 m. At each plane, top and bottom measurements were taken separately due to the robot arms movement limitations. These measurements were repeated for 20%, 40%, 60% and 80% of the WindShapers maximum capacity. Preliminary testing showed that stopping the robot arm at individual measurement points introduced sensor vibrations and increased disturbance on the measurement signal. It was therefore chosen to characterize the flow using continuous measurements. A comparison of the continuous and step-wise measurements is given in Appendix B.

Since the characterization was done using coarse measurements, each plane is subdivided into 20×20 mm bins. For each bin, the mean velocity and local TI is calculated using the samples belonging to that bin. Then the mean planar velocity and mean planar TI is calculated from the average of all bins in each plane. The maximum TI is obtained from the bin with the highest calculated TI for each plane. Bins that included robot stops or turnarounds were discarded since they contained robot-transition effects rather than only flow effects. Because the sensor traverses through each bin continuously, the calculated local TI may contain both temporal fluctuations and residual spatial variations within each bin. Therefore it is merely used as an indication of downstream flow fluctuations rather than a precise local turbulence characterization.

Wind box blockage effects were initially quantified by measuring WindShaper flow at a predefined location using an Oxford Probe without the box. No measurable difference was observed at the investigated location between the configurations with and without the wind box.

The mean velocity, mean TI and maximum TI of the different planes at the different operating speeds are given in Figure 2-3. For all WindShaper commands, the mean velocity experiences a small downstream trend for increasing downstream distances. For example, at 60% WindShaper capacity the mean velocity varies between 8.236 m s⁻¹ at plane 2 and 8.006 m s⁻¹ at plane 5. The observed differences are smaller than the manufacturers stated absolute accuracy of ±1 m s⁻¹ and should therefore be interpreted cautiously in terms of absolute

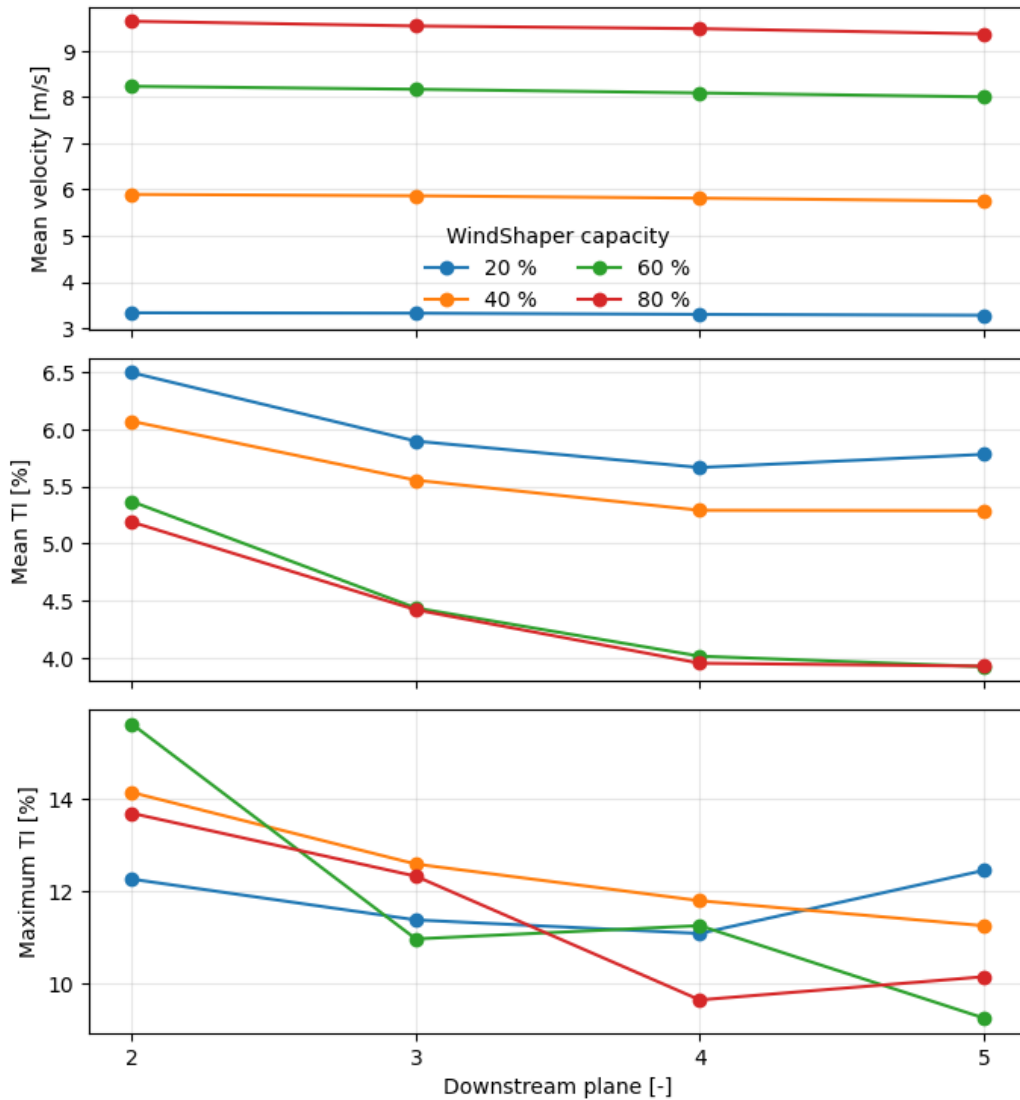


Figure 2-3: Flow characterisation at four downstream planes and four uniform WindShaper operating conditions. Mean velocity and mean turbulence intensity (TI) are spatial averages over the measured 20×20 mm bins. Maximum TI denotes the largest local bin value within the complete measured plane after removal of bins associated with robot pauses or turnarounds.

magnitude. More importantly, for the present control objective, the mean flow magnitude remains approximately preserved for the different downstream planes.

The mean TI shows a more distinct downstream trend. For all operating conditions the highest mean TI is observed at the plane closest to the WindShaper. For further downstream distances, the mean TI becomes increasingly less. Additionally, the differences in mean TI between the first and last plane increase in magnitude under increasing WindShaper commands. For a 20 % WindShaper command the mean TI decreases from 6.493 % at plane 2 to 5.779 % at plane 5, whereas at a 80 % WindShaper command the mean TI decreases from 5.187 % at plane 2 to 3.926 % at plane 5. Even though the maximum TI differs substantially from

the spatial mean, the same overall decreasing trend is visible for larger downstream distances. However, for a 20% and 80% WindShaper capacity, the maximum TI shows a slight increase between planes 4 and 5. Therefore, the mean TI provides the clearest indication of decreasing local flow fluctuations for further downstream planes.

The spatial variation of the mean velocity measurements was also analysed to investigate the spatial non-uniformity of the mean flow. For each bin, the local mean velocity was compared to the planar spatially averaged mean velocity. The corresponding deviation maps for WindShaper input commands of 40% and 60 % are given in Figures B-2-B-5 in Appendix B. These maps show that spatial non-uniformities remain persistent for further downstream planes for both cases. Therefore, in terms of the most spatially uniform flow, there does not seem to be one plane that stands out in particular. That being said, for flow control, these spatial variations are less restrictive since the location of the sensor rake is fixed. Therefore persistent location dependent differences can simply be represented in the identified input-output model.

Overall, the characterization does not identify a preferable region for the sensing plane with the objective of flow control through the mean velocity. The spatially averaged mean velocity is approximately preserved throughout the investigated region and no plane provides consistent spatial uniformity. The main difference for the downstream planes is the reduction of the mean TI at further downstream distances which settles by plane 4. Since the velocity measurements will be averaged in future experiments, the high frequency velocity fluctuations will be attenuated. This means the mean TI is still a good indicator of flow fluctuations but becomes a less important metric for assessing suitable locations for mean velocity flow control. However, since the region between planes 4 and 5 has lower velocity fluctuations, this region is selected for the future control experiments.

2-2-1 Sensor Assessment

Before modelling the input-output relation or developing any controllers, an initial experiment was conducted to analyse the short-term input-output behaviour of the WindShaper. A sequence of uniform step functions with commands ranging between 10% and 60% of the maximum capacity was applied to the WindShaper and the velocities measured at the three sensor locations. Figure 2-4 shows that applying consecutive inputs results in approximately the same measurement. This provides an initial indication of short-term within run consistency under repeated input commands, which will be further evaluated in Chapter 3.

This experiment also reveals consistent offsets between the measured outputs. These differences are already observed when no input is applied to the WindShaper, indicating that there are sensor specific offsets, local background flow or a combination of both. However, when non-zero inputs are applied, the order of these offsets does not remain consistent. For lower velocities sensor 2 consistently measures the highest velocity whereas at higher velocity sensor 3 measures slightly higher than sensor 2. This shows that the differences between measured outputs vary with operating condition and therefore cannot be modelled by constant sensor-specific offsets.

To distinguish sensor dependencies from location dependencies, three additional experiments were done. Again, uniform step inputs are applied to the WindShaper with the measurements initialised by applying no input for the first 20 s. For each consecutive experiment the sensors

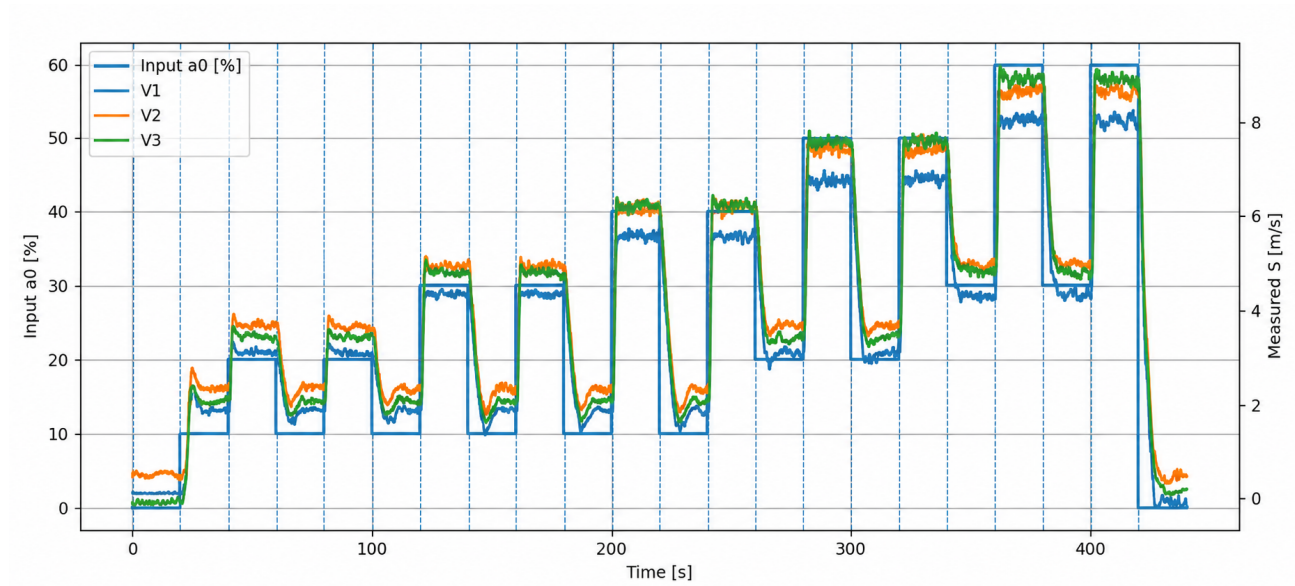


Figure 2-4: Measured velocity response at the three sensor locations to repeated uniform WindShaper step commands. Each input is applied twice, providing an initial indication of short-term within-run consistency and revealing persistent differences between the measured sensor outputs.

are switched from position, such that each sensor is placed at one of the sensor locations once. The sensors are identified as: S_1 , S_2 and S_3 while the fixed locations are defined by T, L and R. The resulting mean velocities are summarized in Table 2-1.

When no input is applied, the sensor at location R consistently measures the highest mean velocity, with values ranging between 0.3888 - 0.4501 m s^{-1} , suggesting local flow contribution. However, at 20 % WindShaper input, sensor S_1 consistently measures the lowest velocity at each location. These observations suggest the presence of sensor-dependent effects. However, each configuration was only tested once, and the magnitude of the observed differences lies within the stated absolute sensor accuracy. Therefore, these experiments can not determine if these observations are repeatable, nor whether these differences have explicit sensor or location dependencies.

Even though the exact cause behind these irregularities cannot be isolated, these observed differences between sensor configurations and locations do not prevent the identification of an approximate input-output model of the WindShaper facility. Although the offsets cannot be represented exactly by a single constant, a constant offset for a fixed sensor configuration can still be used as an approximation. Any remaining operating-dependent differences can then be modelled by the input-output relationship or left as modelling errors.

2-3 Control Problem Formulation

Fan-array wind generators provide significant flexibility because they offer individual control of the fans. This allows them to generate spatially and temporally varying flow conditions. Babu et al. [15] review facilities with these capabilities and show that programmable and coordinated fan actuation can be used to achieve a wide range of flow profiles. For this

Table 2-1: Mean measured velocities for three sensor configurations under uniform WindShaper inputs. The sensors are denoted by S_1 , S_2 , and S_3 , while T , L , and R denote the fixed measurement locations.

Sensor configuration			Input [%]	Mean velocity [m/s]		
T	L	R		T	L	R
S_1	S_3	S_2	0	0.0679	-0.0092	0.3888
S_1	S_3	S_2	10	2.6224	2.9144	3.1201
S_1	S_3	S_2	20	5.4450	6.1044	6.0508
S_3	S_2	S_1	0	0.1415	-0.0369	0.4464
S_3	S_2	S_1	10	3.0180	2.7779	2.9266
S_3	S_2	S_1	20	6.1233	5.8046	5.5711
S_2	S_1	S_3	0	0.2101	0.0424	0.4501
S_2	S_1	S_3	10	2.8793	2.6160	3.1877
S_2	S_1	S_3	20	5.7848	5.5045	6.1966

research, this actuator flexibility is explicitly considered for controlled flow for wind energy control experiments, where heterogeneous and temporally varying inflow conditions need to be generated repeatedly.

The downside to this actuator flexibility is that independent fan actuation results in a high dimensional actuator space. The WindShaper in the WindAI Lab consists of 729 individually controllable fans, which results in the high-dimensional input-space:

$$\mathbf{u}_k = [u_{1,k} \dots u_{729,k}] \in \mathbb{R}^{729}$$

Since there are only three TriSonics, the mean output-space is $\mathbf{y}_k \in \mathbb{R}^3$. For tracking prescribed reference profiles, the control objective at time-step k is generally described as minimizing a cost function such as

$$J_k = \frac{1}{2} \|\mathbf{r}_k - \mathbf{y}_k\|_2^2 \quad (2-2)$$

where $\mathbf{r}_k$ is the desired reference profile. If 729 actuators control the three measured outputs, the control problem is inherently underdetermined: many different fan commands can lead to similar outputs. Therefore, direct optimization of the full input space would require regularization and constraints to find unique and feasible solutions. The WindShaper GUI provides the capability to prescribe input functions in a form of spatial functions. The GUI in Figure 2-5 shows that the WindShaper can be controlled through input functions described through a time variable t and two position variables (x, y) . The input function can therefore be described as a sum of scalable basis functions. To define these input functions more conveniently, the WindShaper coordinate system is normalized, as shown in Figure 2-5. The physical coordinates $(x, y) \in [0, L_x] \times [0, L_y]$ are therefore mapped to normalized coordinates:

$$\zeta = \frac{x - 0.5L_x}{0.5L_x}, \quad \eta = \frac{y - 0.5L_y}{0.5L_y}$$

such that $(\zeta, \eta) \in [-1, 1]$. In order to generate flow capable of producing uniform, x-gradient and y-gradient flow, the following basis functions were chosen: $\phi_1 = 1$, $\phi_2 = \zeta$ and $\phi_3 = \eta$.

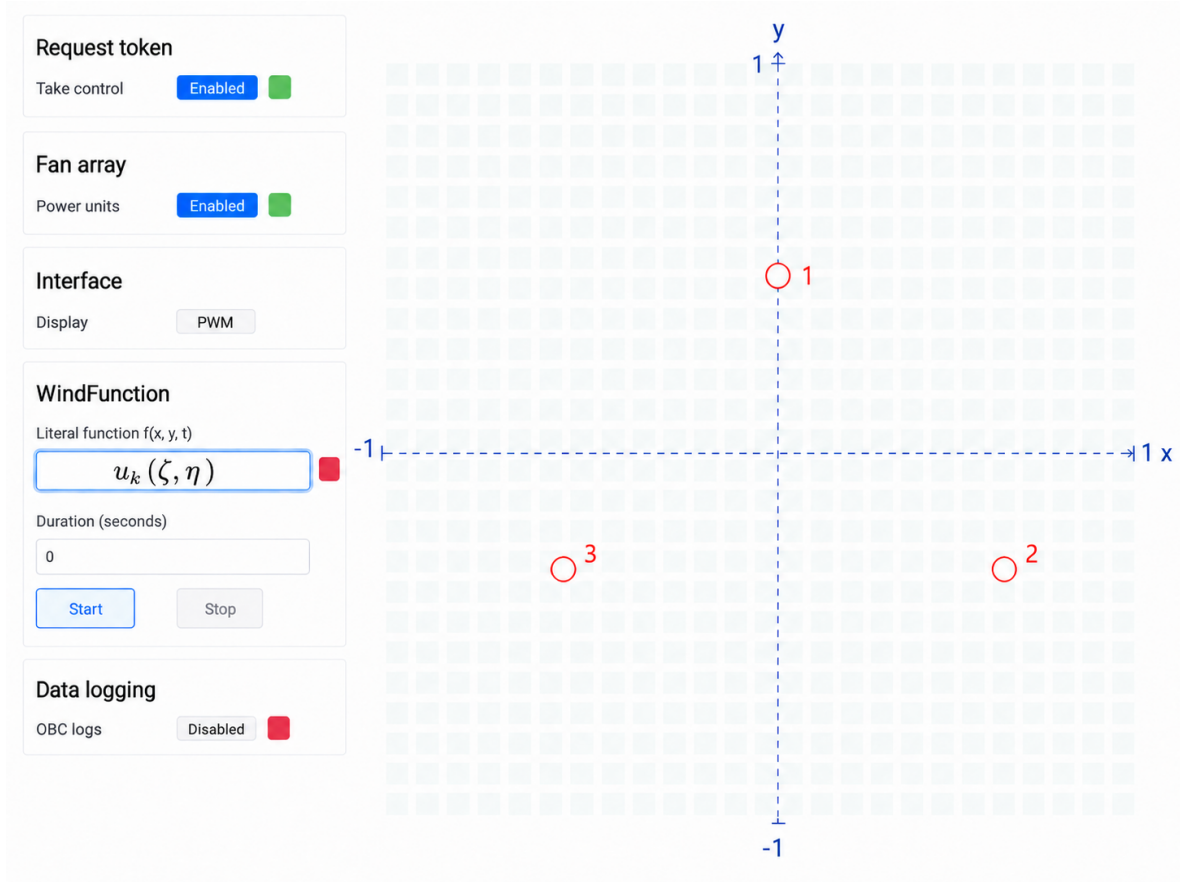


Figure 2-5: The WindShaper GUI showing the normalized coordinate system used to define spatial input functions. The fan-array plane is normalized to $\zeta, \eta \in [-1, 1]$, and the approximate locations of the three measurement sensors are indicated as viewed from the front of the WindShaper.

By scaling these basis functions with scaling coefficients

$$\boldsymbol{\alpha}_k = \begin{bmatrix} \alpha_{0,k} \\ \alpha_{x,k} \\ \alpha_{y,k} \end{bmatrix} \quad (2-3)$$

it is possible to produce different kinds of uniform, x-gradient and y-gradient profiles and combine these flow patterns. By defining this parametrized input function, the decision variable space is reduced from $\mathbf{u} \in \mathbb{R}^{729}$ to $\boldsymbol{\alpha} \in \mathbb{R}^3$. As a result, the parametrized input function can be rewritten as:

$$u_k(\zeta, \eta) = \alpha_{0,k} + \alpha_{x,k}\zeta + \alpha_{y,k}\eta \quad (2-4)$$

The controller will therefore optimize the three physically interpretable decision variables rather than the full 729 independent actuators. This makes the control problem more structured, although it restricts the possible commands to the possible combinations of the basis functions.

2-4 Static Map

For high-dimensional input-output models, black-box models provide an alternative when the input-output dynamics are difficult to derive from first principles [16]. Zavala et al. [12] identify a surrogate model by fitting a regularised linear map using fan-array measurements and use this for open-loop control. They showed that the relation between constant fan-array speeds and time-averaged stream wise velocities are dominantly linear. This finding motivates an approach of modelling the input-output system through a static affine input-output model. However, rather than identifying the input-output map directly from fan commands to mean velocities, it will be identified using the reduced actuator space $\alpha \in \mathbb{R}^3$ introduced in the previous section. This is done directly through the input variables α and the corresponding mean velocity measurements at the three sensor locations. The model at time-step k is defined as

$$\mathbf{y}_k = \mathbf{G}\alpha_k + \mathbf{b} + \mathbf{e}_k \quad (2-5)$$

where $\mathbf{y}_k \in \mathbb{R}^3$ is the mean velocity, $\mathbf{G} \in \mathbb{R}^{3 \times 3}$ the input-output mapping matrix, $\mathbf{b} \in \mathbb{R}^3$ a constant offset vector and $\mathbf{e}_k \in \mathbb{R}^3$ measurement and modelling errors. The fully written out model is defined as

$$\begin{bmatrix} y_{1,k} \\ y_{2,k} \\ y_{3,k} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} \alpha_{0,k} \\ \alpha_{x,k} \\ \alpha_{y,k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \mathbf{e}_k \quad (2-6)$$

This model is not intended to capture any transient or delay effects but captures the approximate relation between the parametrized inputs and mean-velocity measurements.

The static model is identified using 26 constant spatial commands spanning between 5 % to 75 % of the WindShaper command range in order to have enough variation in the input directions. The operating range of these input commands contain uniform, x-gradient, y-gradient and combinations of these command patterns. Each input was applied for 20 s but only the averaged middle ten seconds were used to reduce the effects of transient behaviour. Each input command therefore contributed to one input-output pair:

$$(\alpha^{(j)}, \bar{\mathbf{y}}^{(j)}), \quad j = 1, \dots, N \quad (2-7)$$

The obtained input-output pairs were divided into training and validation data, a complete overview of the applied inputs can be found in Appendix C. It was made sure each group consisted of similar ranges of $(\alpha_0, \alpha_x, \alpha_y)$. In addition, the input-output pairs were distributed equally to reduce the probability that one group contains slow temporal variations.

The model parameters were estimated using a linear least-squares approach. For operating point j , the regressor was defined as

$$\phi^{(j)} = \begin{bmatrix} \alpha_0^{(j)} \\ \alpha_x^{(j)} \\ \alpha_y^{(j)} \\ 1 \end{bmatrix} \quad (2-8)$$

The regressor corresponding to the training data was found to be full rank, satisfying the rank condition required for unique least-squares estimation of the affine model parameters.

The corresponding parameter matrix is defined as

$$\Theta = \begin{bmatrix} \mathbf{G} & \mathbf{b} \end{bmatrix} \quad (2-9)$$

For a single operating point, the model can be written as

$$\mathbf{y}^{(j)} = \Theta \phi^{(j)} + \mathbf{e}^{(j)} \quad (2-10)$$

Stacking this for all operating points N gives:

$$\mathbf{Y} = \Theta \Phi + \mathbf{E} \quad (2-11)$$

where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}^{(1)} & \dots & \mathbf{y}^{(N)} \end{bmatrix} \quad (2-12)$$

and

$$\Phi = \begin{bmatrix} \phi^{(1)} & \dots & \phi^{(N)} \end{bmatrix} \quad (2-13)$$

The corresponding least-squares estimate is then given by

$$\hat{\Theta} = \mathbf{Y} \Phi^T (\Phi \Phi^T)^{-1} \quad (2-14)$$

The resulting parameter estimates are

$$\hat{\mathbf{G}}_0 = \begin{bmatrix} 0.1459495 & 0.004413 & 0.082286 \\ 0.147269 & 0.065835 & -0.050907 \\ 0.143442 & -0.061525 & -0.058173 \end{bmatrix}, \quad \hat{\mathbf{b}}_0 = \begin{bmatrix} -0.363964 \\ 0.011524 \\ 0.368323 \end{bmatrix} \quad (2-15)$$

where the subscript 0 denotes the initial offline model estimate.

The obtained parameter estimates are consistent with the input commands. Each entry for the first column of $\hat{\mathbf{G}}_0$ is approximately equal, indicating that when only a uniform input (α_0) is applied, approximately equal measurements are observed. The second column represents the response to the x-gradient (α_x) command. This shows that the second and third entries are centered approximately equally around the first entry. This is consistent with the way the sensor rake is positioned in box, see Figure 2-5, where sensor 1 is located near the horizontal centre and sensors 2 and 3 at equal opposite sides. A similar pattern can be observed for the third column of $\hat{\mathbf{G}}_0$, which represents the response to (α_y). Sensors 2 and 3 are placed below the vertical centre and have similar magnitudes and sensor 1 above the vertical centre.

The unequal entries in the estimated bias vector $\hat{\mathbf{b}}_0$ represent output dependent contributions in the affine model that can not be described by $\hat{\mathbf{G}}_0 \alpha$. They may be dependent on modelling errors, background flow or sensor location dependencies. However, section 2-2 showed that the exact reason can not be isolated. Since the entries of $\hat{\mathbf{b}}$ are the result of estimating both $\hat{\mathbf{b}}$ and $\hat{\mathbf{G}}$ they cannot be directly explained as sensor dependent offsets.

The obtained matrix has full rank, and its singular values are $\sigma = [0.2528 \ 0.1121 \ 0.089]$. This corresponds to a condition number of $\kappa(\hat{\mathbf{G}}) = 2.8277$. The three parametrized input directions therefore produce linearly independent modelled output responses, and the resulting estimated input-output map $\hat{\mathbf{G}}_0$ is not ill-conditioned. The performance metrics of the parameter estimation are given in Table 2-2. The high R^2 -score and relatively low RMSE for

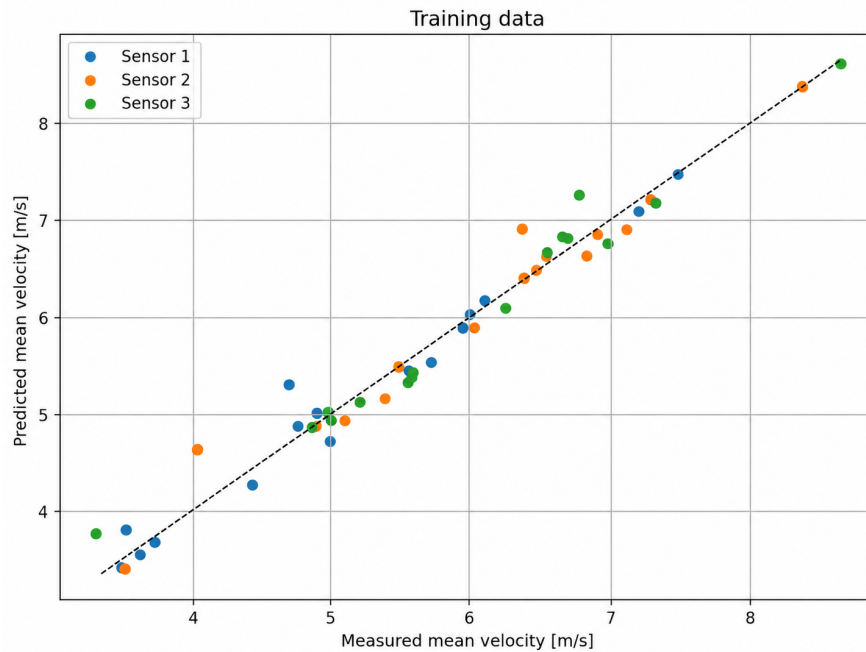
Sensor	Dataset	Runs	R2	RMSE [m s ⁻¹]	MAE [m s ⁻¹]
1	Training	16	0.971647	0.202134	0.146583
2	Training	16	0.963262	0.232413	0.156375
3	Training	16	0.970403	0.207110	0.165801
1	Validation	10	0.431488	0.557916	0.320035
2	Validation	10	0.825437	0.360179	0.195260
3	Validation	10	0.925120	0.352286	0.229270

Table 2-2: Training and validation performance of the identified static affine model at each sensor location, evaluated using R^2 , $RMSE$ and MAE .

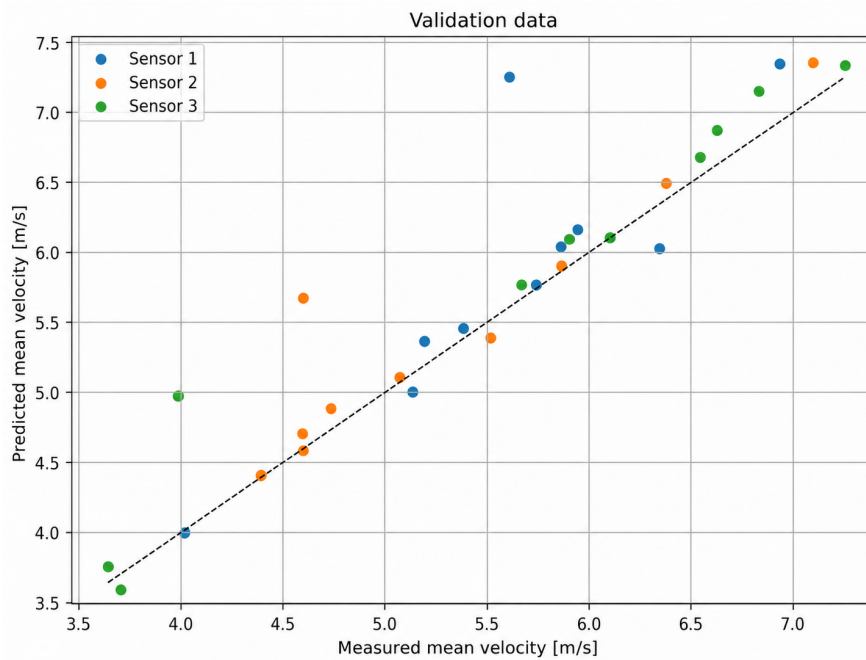
the training data indicate the model fits the training data well. These metrics, in combination with Figure 2-6 show that the the input-output relation is approximately linear.

The validation performance is clearly weaker, with RMSE values ranging between approximately 0.35-0.56 m s⁻¹ and a lower R^2 -score. The R^2 score for sensor 1 on the validation data indicates that the static model explains only a limited fraction of the observed validation-output variation. This is in line with the relatively high RMSE and MAE values for this case. Figure 2-6b shows several validation operating points with substantially larger prediction errors. However, without further validation it can not be confirmed whether these offsets are caused by temporal changes, measurement variability or other forms of model mismatch. The performance metrics support the use of the affine static model as an approximate representation of the dominant input-output relationship.

Overall, the static model represents the mean input-output behaviour relatively well. Despite this linear relation, significant offsets observed with respect to the validation data suggest the static model is not accurate enough for open-loop tracking. Before introducing control approaches that compensate for this model uncertainty for accurate reference tracking, it is necessary to investigate whether the input-output behaviour of the WindShaper experiences changes over the duration of an experiment. Therefore, the next chapter will assess within-run temporal stability of the WindShaper by analysing the output behaviour under fixed stationary and non-stationary inputs



(a) Measured and model-predicted mean velocities for the training dataset.



(b) Measured and model-predicted mean velocities for the validation dataset.

Figure 2-6: Comparison of measured and static model-predicted mean velocities at the three sensor locations for the training and validation datasets. The dashed line denotes $y_{pred} = y_{meas}$. The training data lies close to this line while the validation data shows larger prediction errors, illustrating reduced model accuracy outside the fitted dataset.

Within-run Temporal Stability

Experimental facilities used for wind energy control experiments not only require prescribed flow conditions but also sufficiently stable behaviour over the duration of an experiment. Chapter 2 showed that an affine static model can represent the input-output relation well, but is not accurate enough for open-loop tracking. Before introducing control architectures to compensate for this model uncertainty, it is valuable to know whether the input-output system experiences any systematic changes over the duration of an experiment. Therefore, this chapter will assess the temporal stability of the input-output relation under fixed stationary and periodic input sequences.

Drift in this context is considered as a systematic temporal change in the measured output while the prescribed input sequence remains unchanged. Rather than comparing measurements between two selected intervals, drift is quantified primarily through a fitted rate of change over the duration of each experiment. The accumulated fitted change over the complete run is then used to determine whether the observed trend fits actual drift.

Both stationary and non-stationary fixed input sequences are considered, ranging from uniform, x-and y-gradient to periodic input sequences. Each drift assessment has a duration of 1800 s and is done once. Since these assessments are only done once, they are only useful to investigate behaviour over that duration and cannot be used to assess run-to-run repeatability. For a complete overview of the different experimental runs, see Table 3-1. Even though temperature was monitored, it is not independently controlled, therefore no causal relationship can be established between temperature and velocity measurements. Temperature is therefore treated as supporting information, while temporal stability is only assessed from the fitted change with respect to time. For the temperature evolution over the different experiments, see Figure D-1 in Appendix D-1.

The stationary experiments were averaged over a duration of 0.75 s. The first 80 samples, corresponding to the first 60 s of each run, were discarded to remove any transient start up effects. Using this data, a linear time-dependent model for each sensor was defined by an initial fitted velocity, a constant drift rate and a residual term. The slope therefore represents velocity change per unit of time. The periodic flow used raw data sampled at 100 Hz and was

Table 3-1: Overview of the ten fixed-input experiments used for the temporal-stability assessment. Each individual input setting was applied once for a duration of 1800 s. The response was analysed separately at each of the three sensor locations.

Flow category	Individual input settings	Runs	Analysis performed
Uniform	$\alpha_0 \in \{35, 45, 55, 65\}\%$, with $\alpha_x = \alpha_y = 0$	4	One linear velocity-drift model fitted per sensor and experiment.
x -gradient	$\alpha_x = 30\%$, with $\alpha_0 = \alpha_y = 0$	1	One linear velocity-drift model fitted per sensor.
y -gradient	$\alpha_y = 30\%$, with $\alpha_0 = \alpha_x = 0$	1	One linear velocity-drift model fitted per sensor.
Periodic	$T_s \in \{7.5, 15, 30, 60\}$ s, periodic input with an amplitude of 5%	4	Mean level, amplitude and phase estimated for every complete period, followed by a temporal trend fit for each sensor.
Total		10	

filtered using a Butterworth filter at 1 Hz. This cut-off frequency lies well above the highest imposed periodic frequency of 0.133 Hz ($T_s = 7.5$ s). However the goal is not to isolate a specific frequency range, more so to remove high frequency noise. The data was also filtered with a cut-off frequency of 0.5 Hz and showed negligible differences in the filtered signal. For the non-stationary experiments the first 60 s of each run was also discarded to remove any transient effects.

A periodic response can be defined in mean value, amplitude and phase, meaning its temporal stability cannot be determined using a single slope. Therefore the response is fitted for each individual period. The obtained parameter estimates from each individual period are then fitted over the whole duration, resulting in linear rates of change for the mean value, amplitude and phase. The uncertainty of each fitted rate is quantified using a 95 % confidence interval (CI) [17], see Equation 3-1.

$$CI = \tilde{c} \pm t_{0.975, N-2} SE(\tilde{c}) \quad (3-1)$$

This interval is obtained from the estimated rate ($\tilde{c}$), the corresponding critical Student's t -distribution (t) and the standard error of the fitted rate ($SE(\tilde{c})$), which is defined by

$$SE(\tilde{c}) = \frac{s_e}{\sqrt{\sum_{n=1}^N (t_n - \bar{t})^2}} \quad (3-2)$$

where s_e is the residual standard deviation defined by

$$s_e = \sqrt{\frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{N - 2}} \quad (3-3)$$

Since for every linear fitted model an intercept and slope are estimated, the regression has $N - 2$ degrees of freedom, where N is the number of observations used for the linear fit. For the stationary experiments N represents the number of averaged velocity measurements. For the periodic experiments N represents the number of complete periods used to estimate the change in corresponding sinusoidal parameter value. When the confidence interval excludes

zero, the fitted rate is statistically distinguishable from zero at the 95 % confidence level under the the assumptions of the linear regression model. If the confidence interval contains zero, the available data do not provide sufficient evidence to determine the sign of the rate at that confidence level.

The obtained *CI* need a slight correction in their interpretation. Since it is based on a ordinary least-squares formulation, it assumes temporally uncorrelated residuals. When this assumption is violated, the estimated standard errors and corresponding confidence intervals may no longer accurately represent the uncertainty of the regression coefficients [18]. To verify this assumption, the autocorrelation of the different fitted rates can be found in Appendix D.

3-1 Stationary Input Sequences

The main way of determining drift of the measured velocity is fitting the velocity against time. For sensor i and time-step k , this relation can be written as

$$y_{i,k} = \hat{y}_{i,0} + c_{i,t}(t_k - t_0) + \epsilon_{k,i} \quad (3-4)$$

Here $\hat{y}_{i,0}$ is the initial fitted measurement, c_t the the rate of change in $[\text{m s}^{-1}]$ per sample, $t - t_0$ the elapsed time in samples and ϵ the regression residual. Parameters $\hat{y}_{i,0}$ and $c_{i,t}$ can be found by treating this as a linear regression problem and are therefore found through ordinary least-squares. The measured flow was fitted in m s^{-1} per sample and later converted to $\text{m s}^{-1} \text{min}^{-1}$.

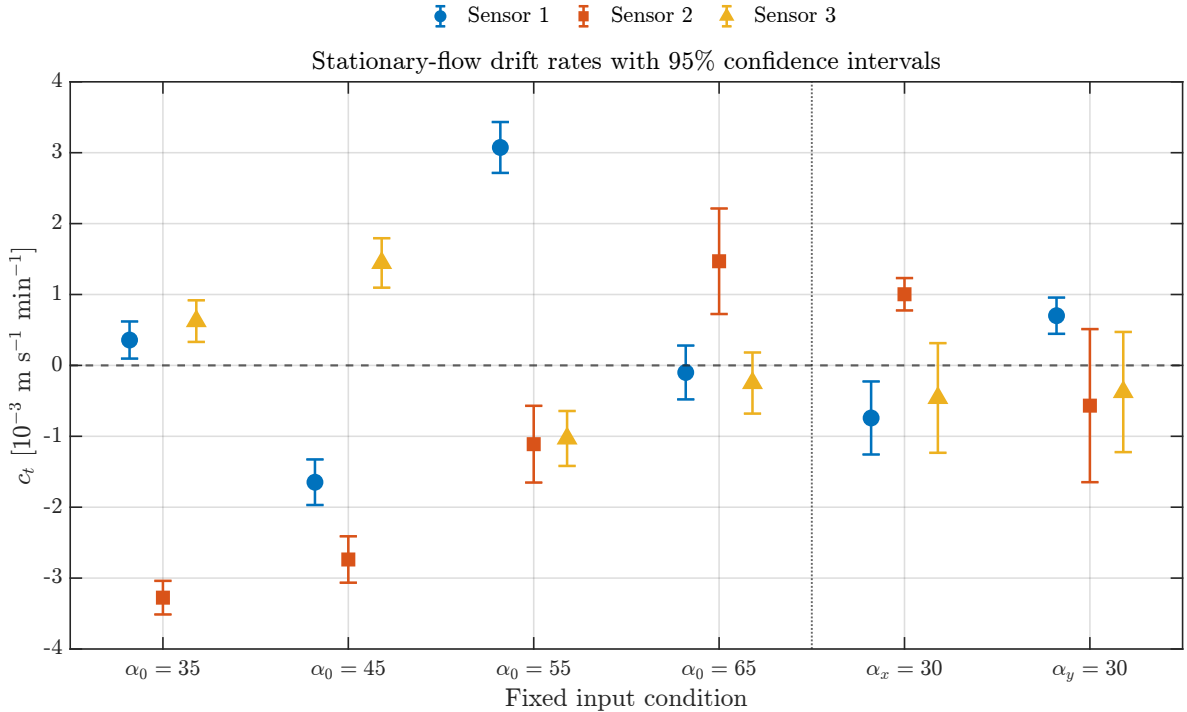


Figure 3-1: Estimated velocity drift rates for the stationary input experiments after discarding the first 60 s. Error bars denote the 95 % confidence intervals of the fitted rates, while the dashed horizontal line indicates zero drift.

The obtained $c_{i,t}$ and confidence interval for each run and sensor are given in Figure 3-1, for a more detailed overview see Table D-1 in Appendix D. A total of five out of 18 confidence intervals contain zero, while the remaining 13 intervals exclude zero. This indicates that for 13 out of 18 intervals, there is sufficient confidence in their direction based on the linear regression model obtained from the available data. However, it was found that the residual independence assumption was not fully satisfied. Autocorrelation is especially present between closely spaced measurement. This is visualised in Figure D-2 in Appendix D. Therefore the *CI*s should be treated as supporting rather than decisive evidence.

The largest observed rate of change was at sensor y_2 for the uniform $\alpha_0 = 35$ run, with a rate of $-3.2757 \times 10^{-3} \text{ [ms}^{-1} \text{ min}^{-1}]$. This trend for y_2 can clearly be observed in Figure 3-2. According to the linear drift model in Equation 3-4 this would result in $\Delta y_2 = -0.0946 \text{ [ms}^{-1}]$ over the whole duration excluding the first 60 s. This is only a decrease of approximately -1.73% with respect to the fitted initial velocity. This combined with the confidence intervals support that the most extreme changes remain below approximately 2% of the initial fitted velocity. It must be stated, however, that the observed autocorrelation also introduces uncertainty in the absolute fitted changes.

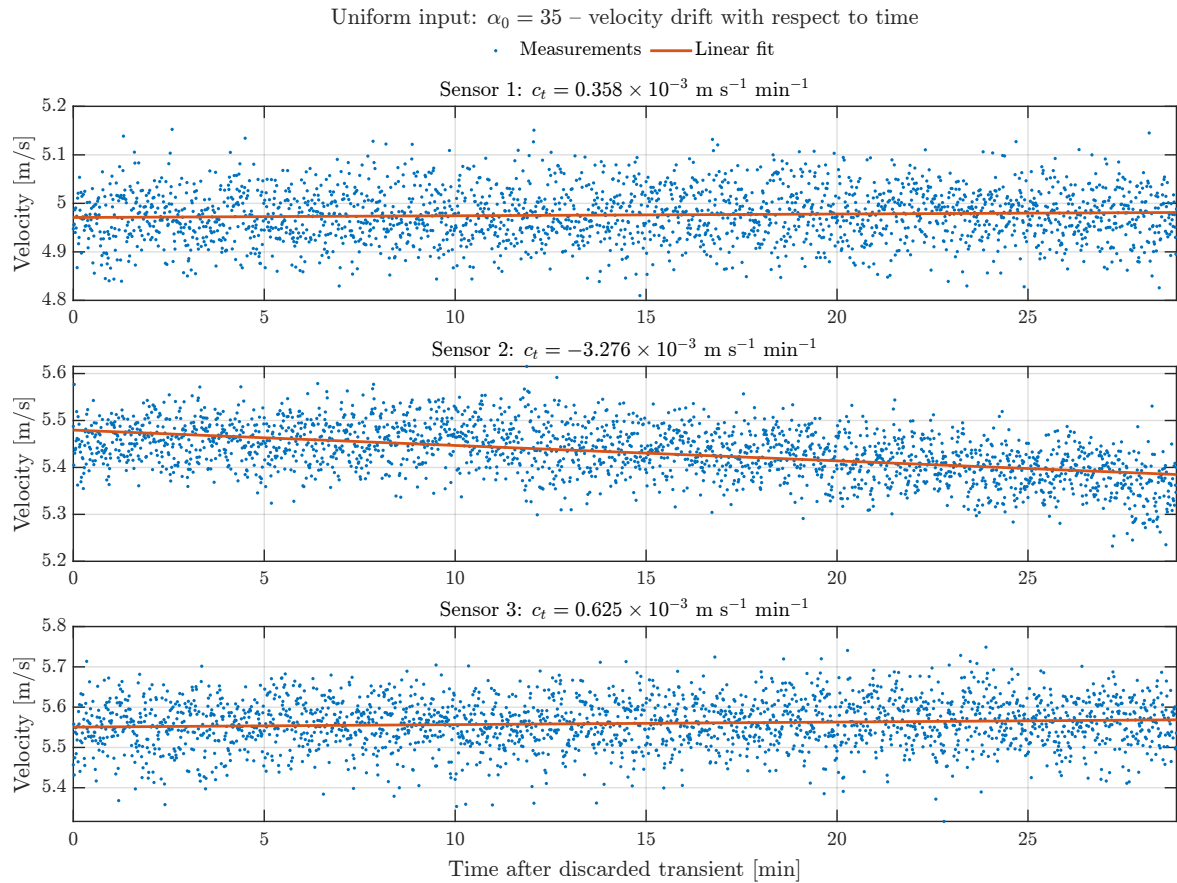


Figure 3-2: Linear drift models fitted to the 0.75 s averaged velocity measurements for the $\alpha_0 = 35$ uniform input experiment after discarding the first 60 s. The measured flow was fitted in ms^{-1} per sample and later converted to $\text{ms}^{-1} \text{ min}^{-1}$. Results are shown separately for the three sensor outputs.

3-2 Periodic Input Sequences

For the temporal stability assessment of periodic flow, the emphasis is limited to sinusoidal input sequences. Tests were done for sinusoidal input functions with different periods and a fixed amplitude. Determining drift is less straight forward compared to the stationary input signals. A sinusoidal function can generally be described with three distinct traits: its amplitude, phase and mean offset. For each sensor i , a sinusoidal output is therefore described as

$$y_i(t) = A_i + R_i \sin(\omega t + \phi_i) \quad (3-5)$$

Where A [m s^{-1}] is the mean velocity, R [m s^{-1}] is the amplitude, ϕ [rad] the phase and $\omega = 2\pi/T_s$ [rad s^{-1}]. This can then be rewritten in a form that is linear in parameters A , B and C :

$$y_i = A_i + B_i \sin(\omega t) + C_i \cos(\omega t) \quad (3-6)$$

The amplitude and phase can then be defined by

$$R_i = \sqrt{B_i^2 + C_i^2}, \quad \phi_i = \text{atan2}(C_i, B_i) \quad (3-7)$$

For every completed period p with N_p measurements, the parameters can be estimated using linear least-squares, by defining an estimator $\theta_i^{(p)}$ and a regressor $\Phi^{(p)}$:

$$\mathbf{y}_i^{(p)} = \Phi^{(p)} \theta_i^{(p)} + \epsilon_i^{(p)}, \quad (3-8)$$

where

$$\mathbf{y}_i^{(p)} = \begin{bmatrix} y_i(t_1) \\ \vdots \\ y_i(t_{N_p}) \end{bmatrix} \in \mathbb{R}^{N_p} \quad (3-9)$$

and

$$\theta_i^{(p)} = \begin{bmatrix} A_i^{(p)} & B_i^{(p)} & C_i^{(p)} \end{bmatrix}^T \quad (3-10)$$

and

$$\Phi^{(p)} = \begin{bmatrix} 1 & \sin(\omega t_1) & \cos(\omega t_1) \\ \vdots & \vdots & \vdots \\ 1 & \sin(\omega t_{N_p}) & \cos(\omega t_{N_p}) \end{bmatrix} \in \mathbb{R}^{N_p \times 3} \quad (3-11)$$

This gives one estimate for A , R and ϕ for each period p . Before estimating the phase drift, each identified phase sequence is unwrapped to prevent discontinuities at $\pm\pi$. Since the sinusoidal amplitude is kept fixed for all experiments and only the excitation period T_s is varied, differences in fitted amplitude and phase can be used to assess frequency dependencies of the WindShaper. Table 3-2 shows the mean estimated values for $\hat{A}$, $\hat{R}$ and $\hat{\phi}$ of all fitted periods for each experimental run. The mean velocity stays approximately constant across all different sinusoidal frequencies. The mean amplitude on the other hand sees a decreasing trend as the frequency increases. For $T_s = 60$ s and $T_s = 30$ s mean fitted amplitude remains approximately constant but for higher frequencies the mean fitted amplitude decreases. The reduced fitted amplitude at shorter periods suggests frequency dependent attenuation of the WindShapers input-output response.

Table 3-2: Mean fitted sinusoidal parameters for each excitation period and sensor. The first 60 s of each experiment were discarded, after which the reported values were averaged over all remaining complete periods. The phase difference is defined relative to the zero-phase sinusoidal input.

T_s [s]	Sensor	Mean level $\bar{A}$ [m s ⁻¹]	Mean amplitude $\bar{R}$ [m s ⁻¹]	Mean phase difference $\bar{\Delta\phi}$ [rad]
30	1	6.1359	0.58388	-0.50053
	2	6.5497	0.61715	-0.51191
	3	6.9227	0.67137	-0.49456
60	1	6.1355	0.58627	-0.30358
	2	6.5125	0.62111	-0.30274
	3	6.9315	0.67176	-0.29603
15	1	6.1216	0.44917	-1.35590
	2	6.5236	0.45344	-1.36210
	3	6.9533	0.50725	-1.38250
7.5	1	6.2384	0.30324	-2.39970
	2	6.6363	0.30025	-2.35520
	3	7.0766	0.32792	-2.37850

To quantify changes in these model parameters over the duration of each experimental run, for each sensor i and period p , they can be fitted onto the following linear models.

$$A_i^{(p)} = A_{0,i} + c_{A,i}(t_p - t_0) + \epsilon_{A,i}^{(p)} \quad (3-12)$$

$$R_i^{(p)} = R_{0,i} + c_{R,i}(t_p - t_0) + \epsilon_{R,i}^{(p)} \quad (3-13)$$

$$\phi_i^{(p)} = \phi_{0,i} + c_{\phi,i}(t_p - t_0) + \epsilon_{\phi,i}^{(p)} \quad (3-14)$$

Here, $c_{A,i}$, $c_{R,i}$ and $c_{\phi,i}$ represent the fitted rates of change of the mean offset, amplitude and phase for each sensor respectively, and t_p the midpoint time of period p . For each run and sensor the different rates and their corresponding confidence intervals are given in Figure 3-3. For c_A seven out of twelve confidence intervals exclude zero, indicating that changes in the mean-output can be resolved for several instances. For c_R only two out of twelve confidence intervals exclude zero. For c_ϕ all of the confidence intervals contain zero. Therefore, no systematic phase drift can be resolved at the 95 % confidence interval. The intervals for all rates, however, need to be treated with caution. While the residual autocorrelation for the amplitude and phase remain limited, this is more pronounced for the mean velocity level. This indicates that the reliability of the obtained *CIs* varies between the different rates and experiments.

In absolute terms, however, Table 3-3 shows that the largest observed amplitude change is -5.15 %, largest mean velocity change is 0.69 % and largest phase difference -0.0956 s. However, especially for the mean velocity trend, the autocorrelation between fitted residuals also introduce uncertainty in the absolute fitted changes. The trends for the input sequence with $T_s = 30$ s is given in Figure 3-4.

Overall, the experiments show that the WindShaper is approximately temporally stable for fixed stationary and periodic inputs over a duration of 1800 s. Several fitted rates visibly differ

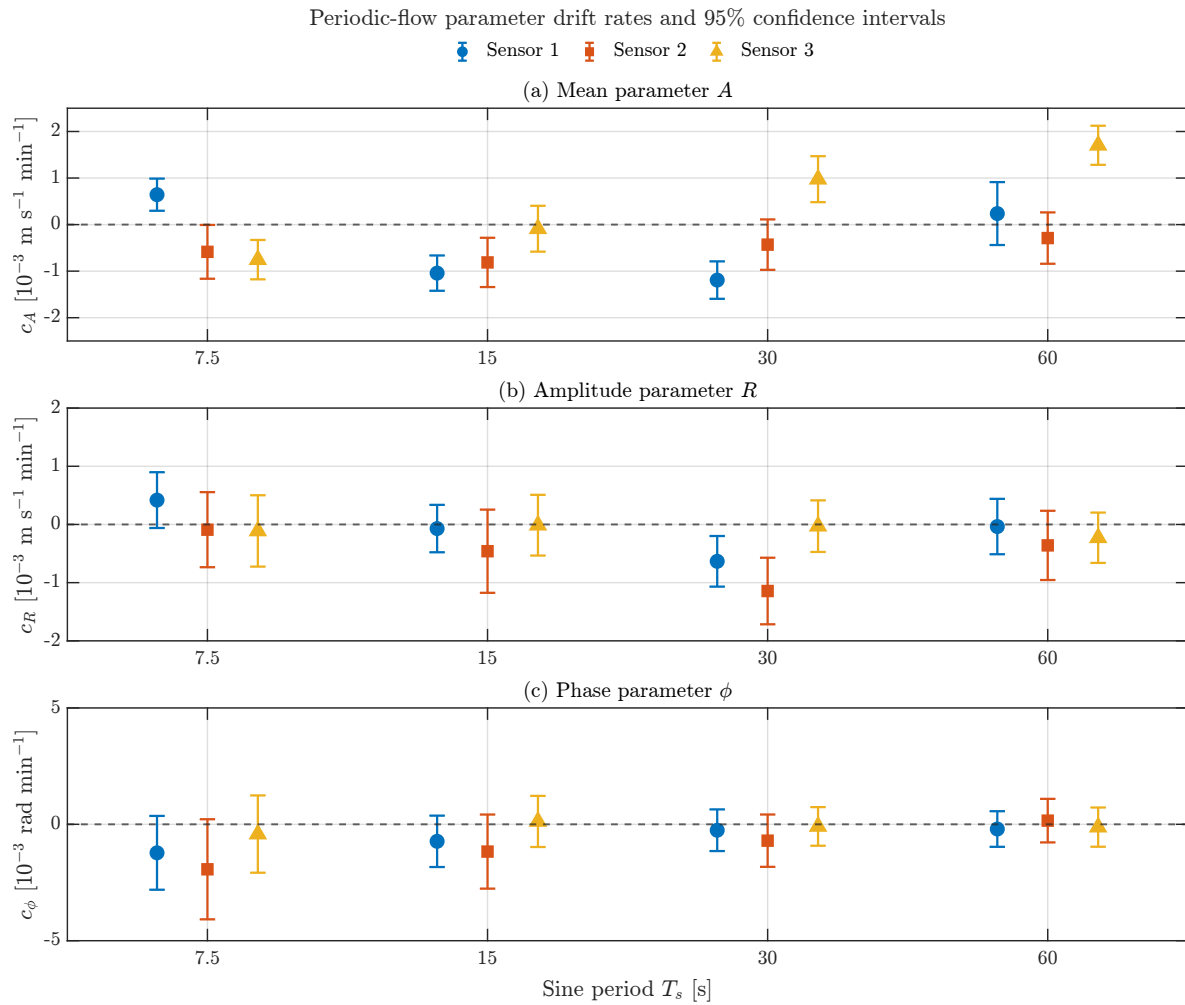


Figure 3-3: The estimated temporal drift rates of the period wise fitted mean level A , amplitude R and unwrapped phase ϕ for all periodic input experiments. Markers denote the three sensor locations and error bars indicate the 95 % confidence intervals of the fitted rates. The dashed horizontal line denotes zero drift.

from zero, but the overall accumulated changes remained relatively small. For the periodic input experiments, the amplitude and phase offset showed only limited systematic change over the run. Residual autocorrelation was present in several occasions, reducing the certainty

Parameter	Input and sensor	Fitted change	Relative interpretation
Mean A	$T_s = 60$ s, sensor 3	$+0.0477 \text{ m s}^{-1}$	0.69 %
Amplitude R	$T_s = 30$ s, sensor 2	-0.0326 m s^{-1}	-5.15 %
Phase ϕ	$T_s = 30$ s, sensor 2	-0.0200 rad	-0.0956 s

Table 3-3: Largest accumulated fitted changes in mean level, amplitude and phase among the 12 excitation period-sensor combinations while excluding transient effects. Changes in mean level and amplitude are also expressed relative to their fitted initial values. The phase shift is converted to the corresponding time shift.

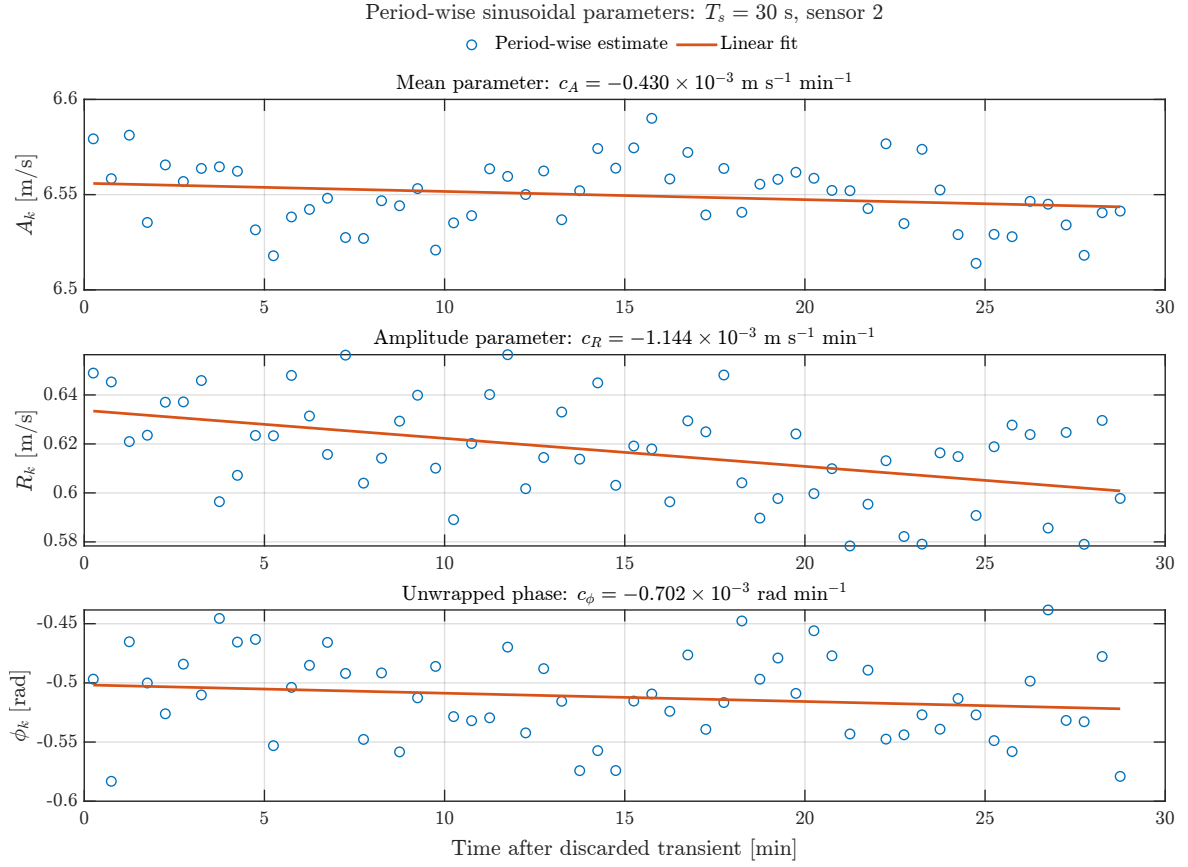


Figure 3-4: Period-wise drift rates of the fitted mean level A , amplitude R and unwrapped phase ϕ for sensor 2 under the $T_s = 30$ s periodic input. Solid lines show the fitted linear temporal trends after removal of the first 60 s.

with which the fitted rates and corresponding accumulated changes can be interpreted as underlying drift, and may cause the ordinary least-squares confidence intervals to underestimate their uncertainty. Nevertheless, the observed fitted changes remained small over the investigated 1800 s duration. The WindShaper is therefore considered approximately temporally stable within the scope of these experiments. The claim of within-run temporal stability over the duration of the experiments must not be confused with run-to-run repeatability, which will be further investigated in Chapter 7

From a control perspective, these results suggest that any tracking errors observed in future experiments cannot be dominantly explained by within-run drift. Model mismatch, unmodelled dynamics and disturbances remain possible contributors. This motivates the investigation of different control architectures to compensate for this uncertainty in input-output behaviour in Chapter 4.

Control Architectures

Chapter 2 showed that it is justifiable to model the parametrized input and the mean velocity measurements with an affine static input-output model. Even though this model captures input-output sensitivities well, due to the remaining prediction error, it was not found to be accurate enough for open-loop reference tracking. Chapter 3 showed that under fixed stationary and periodic input sequences, the WindShaper shows approximate within-run temporal stability over the duration of the experiments. Even though clear trends were visible, the absolute differences remained small. This indicates that within-run temporal drift alone cannot explain the remaining prediction errors of the identified static model, motivating control approaches that can compensate for the remaining plant-model mismatch.

In this research, two conceptually different approaches are investigated. First, the input-output sensitivities will be kept fixed, and tracking errors will be compensated for by using measured tracking errors in a feedback controller. The second approach uses past input-output data to update the model online and determines the control input from the updated static model. These approaches will be referred to as: fixed-model feedback control and online model adaptation with inverse control.

4-1 Fixed-Model Feedback Control

With the objective being accurate reference tracking, open-loop tracking can only be applied when the model describes the system with sufficient accuracy. If the model is inaccurate, directly inverting the static model could lead to persistent tracking errors. By applying feedback laws based on real-time measurements, it is possible to compensate for those errors despite using an inaccurate model.

Even though the static model is not accurate enough for open-loop tracking, the static map $\mathbf{G}$ may still be very useful for feedback purposes, since it captures the dominant input-output directions. The first feedback approach investigated in this thesis is Online Feedback Optimization (OFO), which is an optimization based feedback controller that relies on the identified steady-state input-output sensitivity of the system in its update law. By relying on

this, OFO does not require a detailed dynamic model, but instead uses the static input-output direction $\mathbf{G}$ to determine how the input should be updated. The resulting tracking performance is a direct indicator whether $\mathbf{G}$ provides a sufficiently representative approximation of the steady-state input-output sensitivity for feedback control. A basic block-scheme of this algorithm is given in Figure 4-1. This shows how real-time measurements from the plant are fed back into the update law. Applying operational constraints restricts the control input to the admissible input set. By mapping the tracking error on the input-output sensitivity, this algorithm can compensate for model mismatch and disturbances that appear as steady-state output errors.

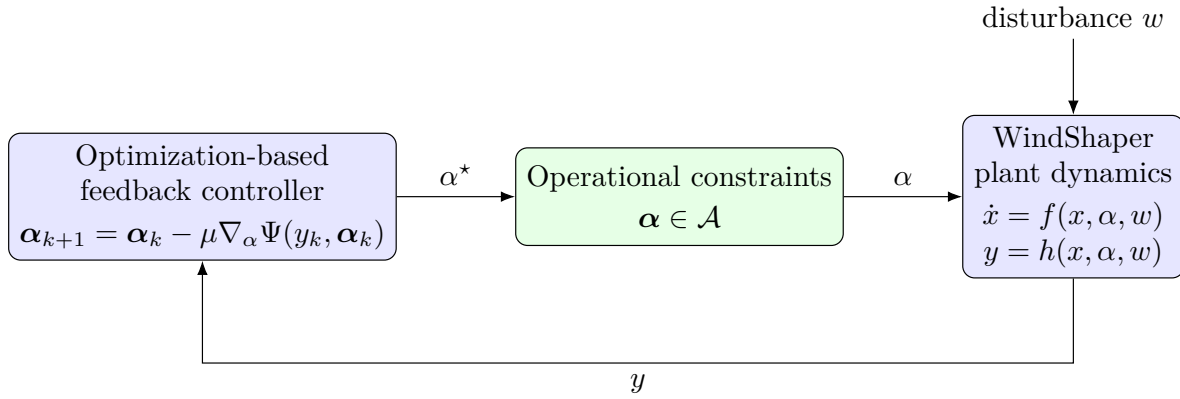


Figure 4-1: General optimization-based feedback control architecture for the WindShaper facility. The controller uses the measured flow output and the identified static map G to update the parametrized input online, while control inputs are restricted to the operational constraints of the admissible input set.

Secondly, a model predictive controller (MPC) will be implemented. In contrast to OFO, MPC is investigated to include the temporal response of the WindShaper. Where OFO uses a static map and therefore does not explicitly account for dynamics, MPC uses a dynamic model over which it can predict behaviour over a receding horizon. This allows temporal effects and input constraints to be incorporated into the control problem. Therefore, MPC is used to investigate whether the temporal dynamics left out in the static model become relevant for non-stationary reference tracking. Figure 4-2 shows a block-scheme of an MPC architecture. Since the prediction model is not accurate enough for open-loop tracking, the bias is updated by using real-time prediction errors. The updated bias is then fed to the MPC.

The goal of these approaches is not to provide an extensive comparison between these and future approaches, but to explore the capabilities and limitations of fixed-model feedback of the WindShaper. Their mathematical description and experimental tracking results are presented in Chapter 5.

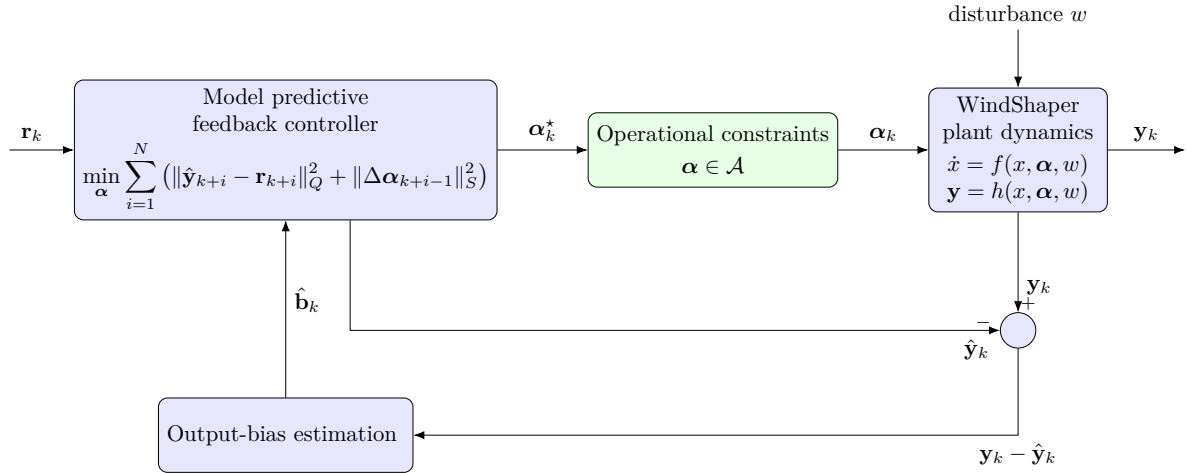


Figure 4-2: General model predictive feedback control architecture for the WindShaper facility. A fixed dynamic model is used to predict the system response and determine the control action over a receding horizon. The prediction error $y_k - \hat{y}_k$ is used to update the output-bias estimate $\hat{\mathbf{b}}_k$, allowing part of the plant-model mismatch to be compensated without adapting the identified dynamic sensitivities.

4-2 Online Model Adaptation and Inverse Control

Initial model fitting in Chapter 2 has shown that a static model fitted over a finite dataset is not able to represent the investigated operating conditions well enough for reference tracking. Therefore, the second architecture proposes to learn these model parameters online through an adaptive least-squares approach. The control input is then defined by using the current model estimates. The static model using estimated model parameters $\hat{\mathbf{G}}$ and $\hat{\mathbf{b}}$ is then written as

$$\hat{\mathbf{y}}_k = \hat{\mathbf{G}}_k \alpha_k + \hat{\mathbf{b}}_k \quad (4-1)$$

the control law can then simply be defined as an inverse control law

$$\alpha_k = \hat{\mathbf{G}}_k^{-1} (\mathbf{r}_k - \hat{\mathbf{b}}_k) \quad (4-2)$$

given that the estimated model parameters are sufficiently conditioned and invertible. Although the physical input-output system remains coupled, since each parametrised input generally affects multiple outputs, model inversion provides a way of accounting for this coupling. Instead of assigning an individual input to each output, the inverse determines a combination of α_0 , α_x and α_y such that the desired output is produced. For a square and invertible $\hat{\mathbf{G}}_k$, substituting Equation 4-2 into 4-1 gives $\hat{\mathbf{y}}_k = \mathbf{r}_k$ since $\hat{\mathbf{G}}_k \hat{\mathbf{G}}_k^{-1} = \mathbf{I}$. In this model-based sense, the outputs are decoupled through the control law based on model inversion. In practice however, model mismatch means that exact decoupling can not be guaranteed, while poor conditioning can make model inversion highly sensitive.

Poor conditioning, however, can be accounted for through regularization. The inverse control law is then defined as

$$\alpha_k = (\hat{\mathbf{G}}_k^T \hat{\mathbf{G}}_k + \rho_G \mathbf{I})^{-1} \hat{\mathbf{G}}_k^T (\mathbf{r}_k - \hat{\mathbf{b}}_k) \quad (4-3)$$

where $\mathbf{I}$ is the identity matrix and ρ_G the regularization term. The regularised inverse control law no longer provides exact model decoupling since the regularization introduces a bias. It

deliberately sacrifices exact inversion to suppress large control actions associated with small singular values of $\hat{\mathbf{G}}_k$.

Compared with the fixed-model feedback controllers, real-time measurements are used to improve the model estimates to obtain a control input rather than updating the control input directly. How well this approach works is dependent on the quality of the online parameter estimates. If certain input directions are insufficiently excited, the data contain insufficient information to estimate the associated model parameters reliably. Therefore, sufficient input excitation is a key requirement for reliable parameter estimates.

Online model adaptation naturally fits in a least-squares framework. Least-squares estimation can be implemented in various ways, differing in how data is treated. This offers a trade-off in adaptation speed to a changing plant and in estimation reliability. A basic block scheme of an online model adaptation algorithm with an adaptive least-squares approach is given in Figure 4-3. The different least-squares approaches and simulation performances are described in Chapter 6.

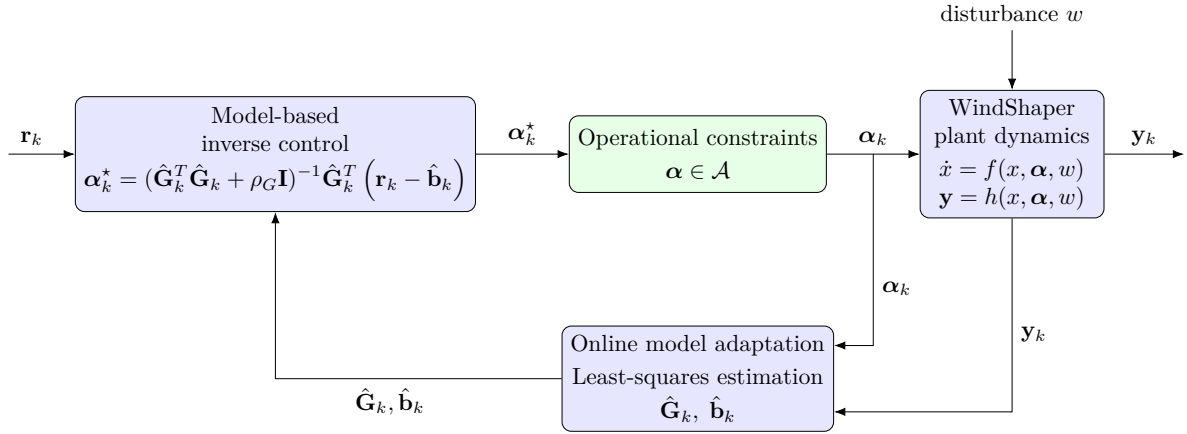


Figure 4-3: Online model adaptation and inverse-control architecture for the WindShaper facility. Measured input-output data are used to update the estimated static map $\hat{\mathbf{G}}_k$ and offset $\hat{\mathbf{b}}_k$ online. The updated model is subsequently used to determine the control input required for tracking the prescribed mean-velocity reference.

The two architectures discussed in this chapter tackle the same underlying problem in different ways: minimizing tracking error by compensating for input-output uncertainty. The following chapters are aimed at implementing both architectures and assessing their performance with respect to reference tracking.

Chapter 5

Feedback Control

This chapter experimentally investigates the fixed-model feedback controllers introduced in Chapter 4. The objective here is not to provide an extensive comparison between controllers, but to provide insight into what the fixed input-output model has to offer in a feedback architecture. First, OFO is implemented to determine whether the static map is sufficiently informative for closed-loop reference tracking. Next, an MPC architecture is used to investigate the relevance of including temporal dynamics.

5-1 Online Feedback Optimization

OFO allows us to assess whether the input-output behaviour is informative enough for closed-loop reference tracking. This algorithm relies on a steady-state map to drive the optimization problem towards the solution. Such a steady-state input-output relation is generally described as

$$y = h(\alpha) + w \quad (5-1)$$

where $h(\alpha)$ is the steady-state map and w represents unknown disturbances. Therefore OFO does not rely on a complicated dynamic model. The control objective for OFO is generally described as

$$\begin{aligned} \min_{\alpha, y} \quad & \Psi(y) \\ \text{s.t.} \quad & y = h(\alpha) + w, \\ & \alpha \in \mathcal{A}, \\ & y \in \mathcal{Y}. \end{aligned}$$

where Ψ is the objective function and $\mathcal{A}$ and $\mathcal{Y}$ represent the admissible input and output space. The actual control update is done through a gradient descent update, which modifies the optimization variable with respect to the cost function. The gradient descent update law is written as

$$\alpha_{k+1} = \Pi_{\mathcal{A}}[\alpha_k - \mu \nabla_{\alpha} \Psi(y_k, \alpha_k)] \quad (5-2)$$

where μ is the learning rate and $\Pi_{\mathcal{A}}$ the projector operator onto the admissible input set. The full derivation with the static model is done as follows. For the static model

$$\mathbf{y} = \mathbf{G}\boldsymbol{\alpha} + \mathbf{b} \quad (5-3)$$

the cost function in the optimization problem is defined as

$$\Psi = \frac{1}{2}(\mathbf{r} - \mathbf{y})^T(\mathbf{r} - \mathbf{y}) \quad (5-4)$$

By using the chain rule gradient, the update part of Equation 5-2 can then be written as

$$\nabla_{\boldsymbol{\alpha}} \Psi = -\left(\frac{\partial h}{\partial \boldsymbol{\alpha}}\right)^T(\mathbf{r} - \mathbf{y}) \quad (5-5)$$

where the steady-state map is approximated as

$$\frac{\partial h}{\partial \boldsymbol{\alpha}} \approx \mathbf{G} \quad (5-6)$$

Then the online feedback optimization update rule is written as

$$\boldsymbol{\alpha}_{k+1} = \Pi_{\mathcal{A}}[\boldsymbol{\alpha}_k + \mu \mathbf{G}^T(\mathbf{r}_k - \mathbf{y}_k)] \quad (5-7)$$

For the WindShaper the following input constraints are applied.

$$\begin{bmatrix} 20 \\ -15 \\ -15 \end{bmatrix} \leq \begin{bmatrix} \alpha_0 \\ \alpha_x \\ \alpha_y \end{bmatrix} \leq \begin{bmatrix} 80 \\ 15 \\ 15 \end{bmatrix} \quad (5-8)$$

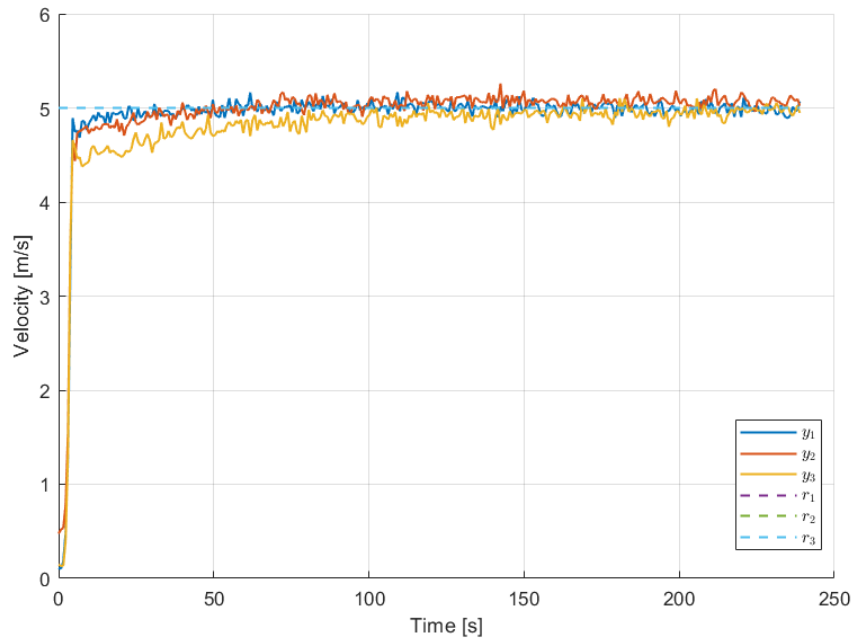


Figure 5-1: Measured mean-velocity tracking of a homogeneous stationary reference using online feedback optimization with the fixed static input-output map. Dashed lines denote the prescribed references and solid lines the three measured sensor outputs.

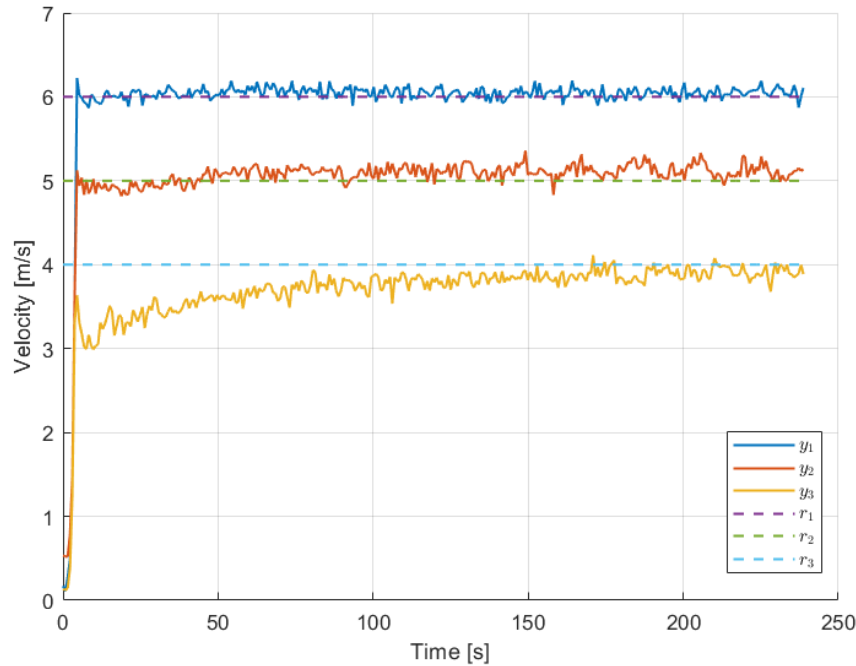


Figure 5-2: Measured mean-velocity tracking of a heterogeneous stationary reference using online feedback optimization with the fixed static input-output map. Dashed lines denote the prescribed references at the three sensor locations and solid lines the corresponding measured outputs.

Figures 5-1 and 5-2 visualize the tracking performance of the online feedback optimization algorithm to a homogeneous and heterogenous reference profile respectively. This shows that the algorithm is well able to track stationary reference profiles.

These experiments show that online feedback optimization is well able to account for the inaccuracies of the static model and track prescribed stationary reference profiles accurately. The formulation of this algorithm, however, relies on a steady-state input-output map and does not describe transient behaviour between the WindShaper and the measurement system.

This is a prominent limitation when it comes to tracking dynamic reference profiles. Since the flow produced by the WindShaper needs to travel towards the measurement system, this introduces a delay d and means that measurements at some time-step k are influenced by inputs at time-step $k - d$. Therefore instantaneous tracking no longer corresponds to the current control input. This is visualised in Figure 5-3, showing that although the output follows the variation of the reference well, a phase offset remains. This motivates explicitly accounting for the delay and transient response when tracking dynamic references. The following section briefly goes into this dynamic relation in more detail.

5-2 Dynamic Model

The static model introduced in Section 2-4 described an input-output model that does not incorporate any transient effects. The previous section showed this works well for stationary

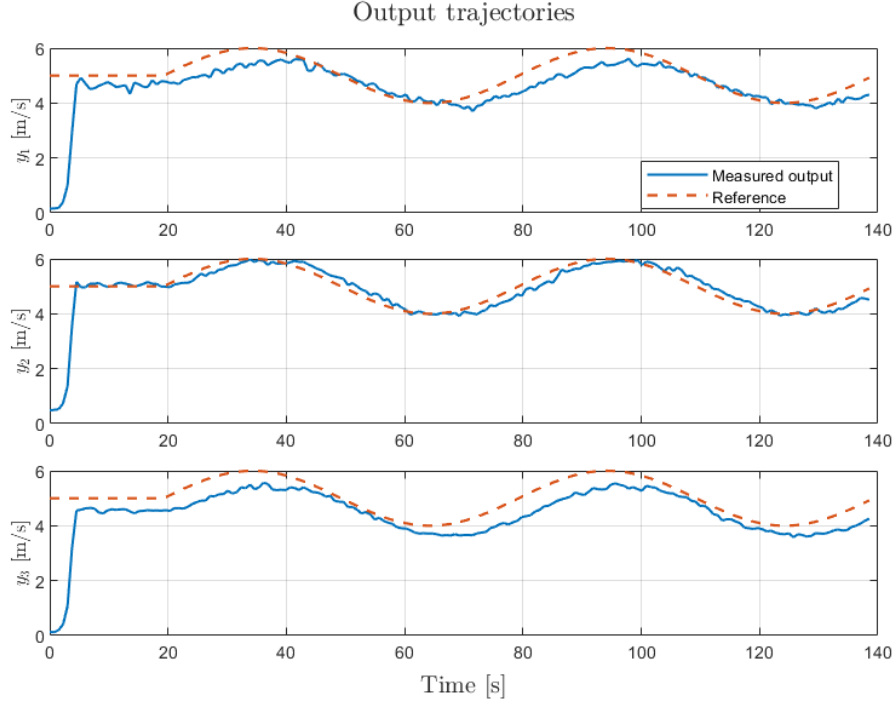


Figure 5-3: Measured tracking of a periodic reference with $T_s = 30$ s using online feedback optimization based on the fixed static input-output map. The measured outputs follow the periodic reference variation but exhibit a phase offset, illustrating the limitation of a static model for non-stationary reference tracking.

reference signals. However, for dynamic reference signals, transient effects become relevant.

A Finite Impulse Response (FIR) model can be used as the prediction model for MPC [19]. Using the same input vector α and output vector $\mathbf{y}$, such a model is generally written as

$$\mathbf{y}_k = \mathbf{b} + \sum_{i=0}^{m-1} \mathbf{H}_i \alpha_{k-d-i} \quad (5-9)$$

where H_i describes the effect of previous inputs on the current output, $\mathbf{b}$ is a bias term, d the delay and m determines the finite memory of the FIR model. The update law for $\hat{\mathbf{b}}$ is defined as

$$\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_{k-1} + \lambda_b (\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \quad (5-10)$$

where $\hat{\mathbf{y}}_k^-$ is the prediction before updating the bias, This is defined as

$$\hat{\mathbf{y}}_k^- = \hat{\mathbf{b}}_{k-1} + \sum_{i=0}^{m-1} \mathbf{H}_i \alpha_{k-d-i} \quad (5-11)$$

A Pseudo Random Binary Signal was designed over a duration of 1200 s to excite the system. The different input coefficients varied between $35 \leq \alpha_0 \leq 45$, $-10 \leq \alpha_x \leq 10$ and $-10 \leq \alpha_y \leq 10$. Each constant level was held for 6 s to let the flow settle.

A model memory of three time-steps was selected, such that

$$\Theta = \begin{bmatrix} \mathbf{H}_0 & \cdots & \mathbf{H}_{m-1} & b \end{bmatrix}$$

and a corresponding regressor vector

$$\phi_k = \begin{bmatrix} \alpha_{k-d}^T & \alpha_{k-d-1}^T & \cdots & \alpha_{k-d-m+1}^T & 1 \end{bmatrix}^T$$

A single sample therefore simply is

$$\mathbf{y}_k = \Theta \phi_k + \epsilon_k \quad (5-12)$$

By stacking all samples for which the complete delayed input history is available, the regression matrices are defined as

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} \mathbf{y}_{d+m-1} & \mathbf{y}_{d+m} & \cdots & \mathbf{y}_{N-1} \end{bmatrix}, \\ \Phi &= \begin{bmatrix} \phi_{d+m-1} & \phi_{d+m} & \cdots & \phi_{N-1} \end{bmatrix}, \\ \mathbf{E} &= \begin{bmatrix} \epsilon_{d+m-1} & \epsilon_{d+m} & \cdots & \epsilon_{N-1} \end{bmatrix}. \end{aligned} \quad (5-13)$$

allowing the FIR model to be written in the linear regression form of

$$\mathbf{Y} = \Theta \Phi + \mathbf{E}$$

The FIR coefficients were estimated from input-output data using a least-squares approach. The length of the memory was selected by comparing the magnitude of the entries in each $\mathbf{H}_i$ for large m . It was found that for a sampling time of 0.375 s, a memory of $m = 3$ represented the system well and the magnitudes of $\mathbf{H}_i$ for larger m became negligible. It must be noted that a different sampling time is used here compared to other experiments.

To assess whether incorporating temporal dynamics can compensate for the limitations observed with the static model OFO approach, the FIR model was applied in a model predictive framework. This means that at each time-step, the controller predicts the future output over a predefined horizon and determines the optimal control input based on minimizing the tracking error over the horizon. To compensate for persistent model inaccuracies, the bias estimate was updated online using the prediction error, see Equation 5-10.

The optimization problem selected for the MPC is

$$\min_{\alpha} \sum_{n=1}^N \left(\|\hat{\mathbf{y}}_{k+n} - \mathbf{r}_{k+n}\|_{Q_y}^2 + \|\Delta \alpha_{k+n-1}\|_{S_\alpha}^2 \right) \quad (5-14)$$

where $\hat{\mathbf{y}}$ is the predicted output, $\mathbf{r}$ the desired reference and $\Delta \alpha$ the input rate of change. The MPC has the following controller settings

$$\mathbf{Q}_y = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \mathbf{S}_\alpha = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}, N = 10, \lambda_b = 0.2, d = 3 \quad (5-15)$$

Where $\mathbf{Q}_y$ is the output-tracking weight matrix, $\mathbf{S}_\alpha$ the input-rate weight matrix, N the

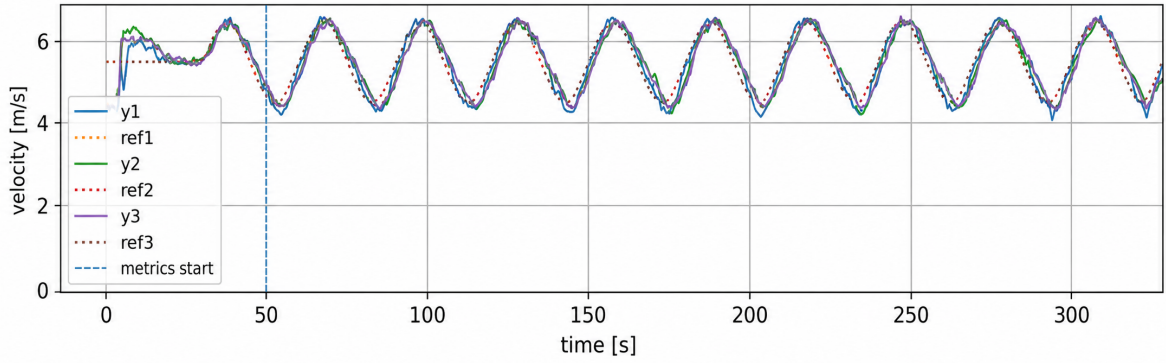


Figure 5-4: Measured tracking of a sinusoidal reference with $T_s = 30$ s using the FIR-based model predictive controller with online output-bias adaptation. The dynamic model explicitly accounts for the identified input-output delay and finite response memory.

prediction horizon, d the delay and λ_b the learning rate for online bias adaptation. With input-rate constraints of

$$\begin{bmatrix} -0.5 \\ -0.25 \\ -0.25 \end{bmatrix} \leq \begin{bmatrix} \Delta\alpha_0 \\ \Delta\alpha_x \\ \Delta\alpha_y \end{bmatrix} \leq \begin{bmatrix} 0.5 \\ 0.25 \\ 0.25 \end{bmatrix} \quad (5-16)$$

Figure 5-4 shows the tracking for a periodic reference with $T_s = 30$ s. When comparing this to the static-model-based approach, the MPC controller can explicitly account for the temporal variations through the identified FIR model. The accurate tracking performance indicates that the dynamic behaviour that is left out of the static model is actually relevant when it comes to non-stationary reference profiles. This result, however, should be considered as a proof of concept for the control relevance of the identified dynamic model instead of an assessment of dynamic reference tracking. No systematic analysis of the FIR model in a MPC framework was done since only a limited set of reference profiles and controller settings were considered. Therefore, the dynamic controller is not further considered in this research.

5-3 Final Remarks

The experimental results in this chapter have shown that the fixed identified static model is sufficiently informative for closed-loop reference tracking in a OFO framework. The accurate tracking of stationary reference profiles within this framework also supports interpreting the identified map $\mathbf{G}_0$ as a representative steady-state input-output sensitivity matrix under the investigated operating conditions. The MPC experiments additionally showed that the temporal dynamics omitted by the static model become relevant for non-stationary reference tracking and that the FIR model can be used in a predictive way within a constrained feedback architecture.

Ultimately, a feedback architecture is undesired for future wind energy control experiments. If a sensor setup is placed in the wind box, this can provide spatial and temporal control at the sensor locations but will also result in disturbed flow further downstream. Therefore a

feedback architecture will not be investigated further. Instead, the next Chapter will investigate the online model adaptation with an inverse control architecture further. This could fit the facility better since the most suitable parameters can be learned during a calibration phase for certain operating conditions, after which the sensors can be removed.

Online Model Estimation and Inverse Control

Chapter 4 introduces online model adaptation with inverse control as an alternative approach to fixed-model feedback control. Instead of using the measured tracking error to directly update the control input, input-output data is used to update the model parameters online and apply them in the inverse control law. For the static input-output model

$$\mathbf{y}_k = \mathbf{G}\boldsymbol{\alpha}_k + \mathbf{b} \quad (6-1)$$

the control input can be determined from the estimates $\hat{\mathbf{G}}_k$ and $\hat{\mathbf{b}}_k$ using the regularized inverse control law in Equation 6-2.

$$\boldsymbol{\alpha}_k = (\hat{\mathbf{G}}_k^T \hat{\mathbf{G}}_k + \rho_G \mathbf{I})^{-1} \hat{\mathbf{G}}_k^T (\mathbf{r}_k - \hat{\mathbf{b}}_k) \quad (6-2)$$

When a known input-output delay of d samples is included, the input applied at time-step k is computed using the future reference r_{k+d} . However, this chapter solely discusses the mathematical formulation of the different online model estimation approaches and implements them in a simulation model which did not include any delays. The delays were accounted for during the experimental validation in Chapter 7

If the estimates are an accurate representation of the system, the control law becomes a model-based feedforward action for tracking prescribed references. This means that the accuracy of the reference tracking is directly dependent on the quality of the online estimates. The relevance of this is accurately depicted in Figure 6-1. Here the offline estimated model parameters are used in the inverse control law and applied to the actual WindShaper, resulting in persistent tracking errors.

The reliability of these estimates is not merely determined by the amount of data. It must be made sure that the applied inputs are sufficiently excited to distinguish input-output sensitivities from the measured data. Online model adaptation will be implemented with several different applications of an adaptive least-squares algorithm. By introducing trade-offs

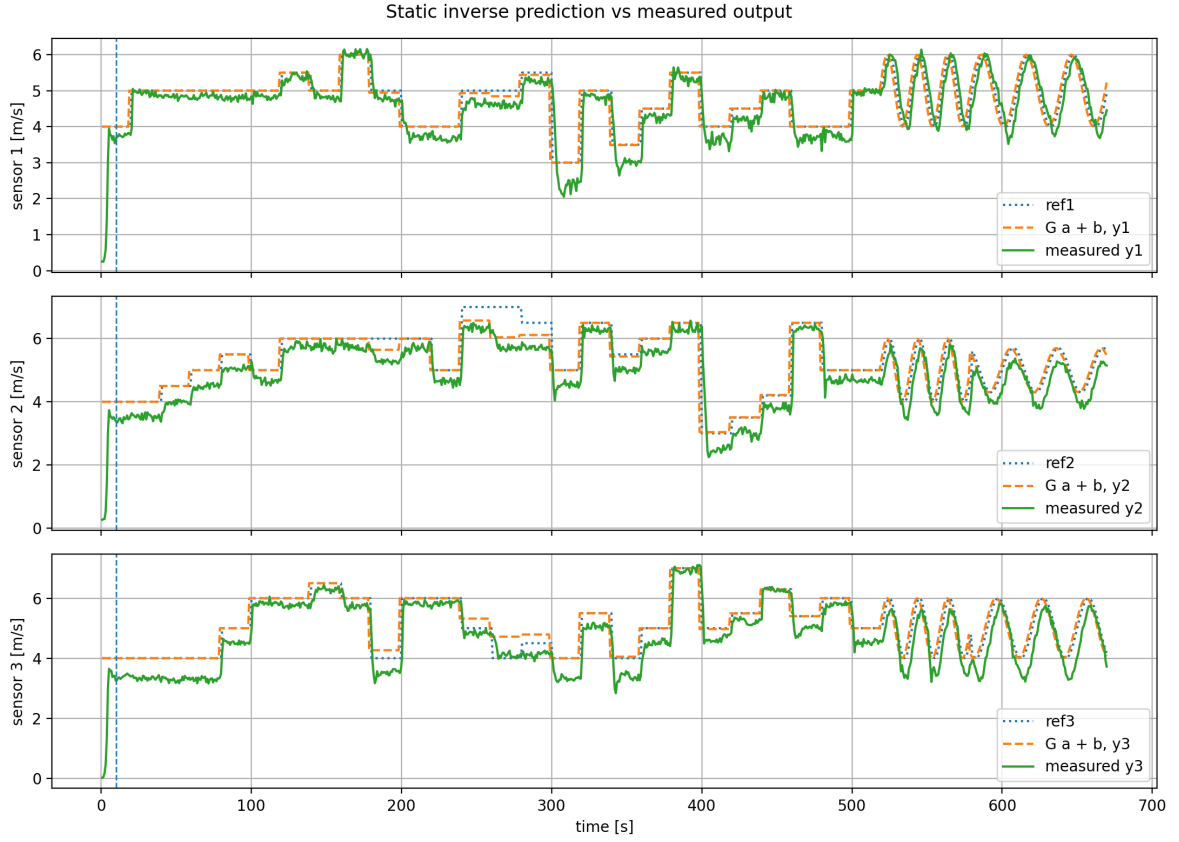


Figure 6-1: Open-loop inverse control tracking using the fixed identified static model. The prescribed reference, model predicted output $\mathbf{G}\mathbf{a} + \mathbf{b}$ and measured output are shown for each sensor channel

in adaptability and estimation reliability, there are different ways of treating past data. This chapter investigates whether adaptive least-squares algorithms can improve reference tracking compared to the fixed nominal model in open-loop under different operating conditions in a simulation environment. Furthermore, it will investigate under which excitation conditions and data-usage the online estimations remain reliable. The resulting insights are used to define the settings for experimental validation under nominal and enforced model mismatch conditions.

6-1 Adaptive Least-Squares

The online estimation problem uses the same structure as the offline identification problem. For each time-step k , the input-output model is written as

$$\mathbf{y}_k = \mathbf{\Theta}_k \phi_k + \mathbf{e}_k \quad (6-3)$$

where

$$\phi_k = \begin{bmatrix} \alpha_k \\ 1 \end{bmatrix} \in \mathbb{R}^4, \quad \mathbf{\Theta} = \begin{bmatrix} \hat{\mathbf{G}} & \hat{\mathbf{b}} \end{bmatrix} \in \mathbb{R}^{3 \times 4}. \quad (6-4)$$

For a selected dataset D_k at time-step k with N_D samples, the stacked representation of the regression problem becomes

$$\mathbf{Y}_D = \mathbf{\Theta}\mathbf{\Phi}_D + \mathbf{E}_D$$

with

$$\mathbf{Y}_D = \begin{bmatrix} \mathbf{y}_{d_1} & \mathbf{y}_{d_2} & \cdots & \mathbf{y}_{d_{N_D}} \end{bmatrix} \in \mathbb{R}^{3 \times N_D}, \quad (6-5)$$

and

$$\mathbf{\Phi}_D = \begin{bmatrix} \phi_{d_1} & \phi_{d_2} & \cdots & \phi_{d_{N_D}} \end{bmatrix} \in \mathbb{R}^{4 \times N_D}. \quad (6-6)$$

Here, the notation $d_1, \dots, d_{N_D}$ stands for time-steps $1, \dots, N_D$ within dataset D_k . For simplicity, $\mathbf{Y}_D$, $\mathbf{\Phi}_D$ and $\mathbf{E}_D$ denote matrices constructed from the currently selected dataset D_k . The regularised least-squares estimate is then given by

$$\hat{\mathbf{\Theta}}_k = \mathbf{Y}_D \mathbf{\Phi}_D^T \left(\mathbf{\Phi}_D \mathbf{\Phi}_D^T + \rho_{\Phi} \mathbf{I} \right)^{-1}, \quad (6-7)$$

where ρ_{Φ} is the least-squares regularization parameter and $\mathbf{I}$ the identity matrix. Since the regularisation term in the estimate of $\mathbf{\Phi}_D$ only helps for numerical stability and does not make $\hat{\mathbf{G}}$ well conditioned, regularization is also done for the inverse control law as is defined in Equation 6-2. The updated model parameters from the least-squares estimate are extracted as

$$\hat{\mathbf{\Theta}}_k = \begin{bmatrix} \hat{\mathbf{G}}_k & \hat{\mathbf{b}}_k \end{bmatrix}. \quad (6-8)$$

The following sections define four different ways of implementing an adaptive least-squares algorithm

6-1-1 Cumulative least-squares

In cumulative least-squares, all samples collected since the start of each run are used for each model update. After a minimum of N_{min} samples are collected, the cumulative least-squares approach updates its estimate at time-step k by defining dataset

$$D_k = \{1, 2, \dots, k\}$$

This means the amount of data used for the cumulative least-squares approach grows continuously throughout each experimental run. The advantage of this approach is that throughout an experiment the estimation becomes less sensitive to measurement noise. The disadvantage is that old data remains dominant. This makes it less adaptive to changing model parameters. The algorithm for the cumulative least-squares uses the general solution as given in Equation 6-7. The whole algorithm is given below.

$$\begin{aligned} & \text{if } k < N_{min} \text{ then} \\ & \quad \hat{\mathbf{G}} \leftarrow \mathbf{G}_{nom} \\ & \quad \hat{\mathbf{b}} \leftarrow \mathbf{b}_{nom} \\ & \quad \text{else} \\ & \quad D_k \leftarrow \{1, 2, \dots, k\} \\ & \quad [\hat{\mathbf{G}}, \hat{\mathbf{b}}] \leftarrow \text{LS}(D_k) \\ & \quad \text{end if} \\ & \alpha_k \leftarrow (\hat{\mathbf{G}}_k^T \hat{\mathbf{G}}_k + \rho_G \mathbf{I})^{-1} \hat{\mathbf{G}}_k^T (\mathbf{r}_{k+d} - \hat{\mathbf{b}}_k) \end{aligned}$$

6-1-2 Batch-wise least-squares

In batch-wise least-squares, the model is only updated after a new batch of N_b samples is collected. After a minimum of N_{min} samples, at time-step k , where k is an integer multiple of N_b the dataset is defined as

$$D_k = \{k - N_b + 1, \dots, k\}$$

Only this most recent non-overlapping batch D_k is used for the parameter estimate. Compared to cumulative least-squares, it is more sensitive to measurement noise and poor excitation since it uses less data points. However, it is more adaptive to changes in the plant since it forgets old data after each new batch. The algorithm for the batch-wise least-squares uses the general solution as given in Equation 6-7. The whole algorithm is given below.

$$\begin{aligned} & \text{if } k < N_{\min} \text{ then} \\ & \quad \hat{\mathbf{G}} \leftarrow \mathbf{G}_{\text{nom}} \\ & \quad \hat{\mathbf{b}} \leftarrow \mathbf{b}_{\text{nom}} \\ & \text{else if } \text{mod}(k, N_b) = 0 \text{ then} \\ & \quad D_k \leftarrow \{k - N_b + 1, \dots, k\} \\ & \quad [\hat{\mathbf{G}}, \hat{\mathbf{b}}] \leftarrow \text{LS}(D_k) \\ & \quad \text{end if} \\ \alpha_k & \leftarrow (\hat{\mathbf{G}}_k^T \hat{\mathbf{G}}_k + \rho_G I)^{-1} \hat{\mathbf{G}}_k^T (\mathbf{r}_{k+d} - \hat{\mathbf{b}}_k) \end{aligned}$$

6-1-3 Moving-window least-squares

In moving-window least-squares, the model is updated continuously after each window of sample size N_w is collected. Therefore, after a minimum of N_w samples, at time-step k the dataset D_k is defined as

$$D_k = \{k - N_w + 1, \dots, k\}$$

This approach has each consecutive window overlap with each other. This means one old data point is removed while one is added. This makes the parameter updates more continuous than the batch-wise update, while still allowing the old data to be forgotten. This approach is therefore expected to adapt smoother. The disadvantage is that the estimate can become sensitive to noise or poor excitation if the window is too short. The algorithm for the moving-window least-squares uses the general solution as given in Equation 6-7. The whole algorithm is given below.

$$\begin{aligned} & \text{if } k < N_w \text{ then} \\ & \quad \hat{\mathbf{G}} \leftarrow \mathbf{G}_{\text{nom}} \\ & \quad \hat{\mathbf{b}} \leftarrow \mathbf{b}_{\text{nom}} \\ & \quad \text{else} \\ & \quad D_k \leftarrow \{k - N_w + 1, \dots, k\} \\ & \quad [\hat{\mathbf{G}}, \hat{\mathbf{b}}] \leftarrow \text{LS}(D_k) \\ & \quad \text{end if} \\ \alpha_k & \leftarrow (\hat{\mathbf{G}}_k^T \hat{\mathbf{G}}_k + \rho_G I)^{-1} \hat{\mathbf{G}}_k^T (\mathbf{r}_{k+d} - \hat{\mathbf{b}}_k) \end{aligned}$$

6-1-4 Recursive Least Squares

The recursive least-squares algorithm can be seen as a combination of the cumulative and moving-window least-squares approach. In essence, none of the data is forgotten but by applying an exponentially decaying forgetting factor λ to the cumulatively obtained data, it turns into a soft moving-window approach. Only now the data within this moving-window is not equally dominant due to this exponentially decaying forgetting factor. The effective window is defined as approximately

$$N_{eff} \approx \frac{1}{1 - \lambda_{RLS}}$$

The derivation of the recursive least-squares algorithm goes as follows. The static input-output model at time-step k is written as

$$\mathbf{y}_k = \mathbf{\Theta} \phi_k + \mathbf{e}_k \quad (6-9)$$

where

$$\mathbf{\Theta} = [\mathbf{G} \quad \mathbf{b}] \in \mathbb{R}^{3 \times 4}, \quad \phi_k = [\alpha_{0,k} \quad \alpha_{x,k} \quad \alpha_{y,k} \quad 1]^T \in \mathbb{R}^4. \quad (6-10)$$

Since $\mathbf{G}$ and $\mathbf{b}$ have a total of 12 to be identified variables, at least four sufficiently exciting data points are required to reliably estimate $\mathbf{G}$ and $\mathbf{b}$ online. More specific, the weighted regression information matrix must have rank four for all parameter directions to be identifiable. The exponentially weighted least-squares cost function is defined as

$$J_k(\mathbf{\Theta}) = \sum_{i=1}^k \lambda_{RLS}^{k-i} \|\mathbf{y}_i - \mathbf{\Theta} \phi_i\|^2 \quad (6-11)$$

This can also be rewritten into a stacked representation as

$$\mathbf{Y}_k = [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{y}_k] \in \mathbb{R}^{3 \times k}, \quad (6-12)$$

$$\mathbf{\Phi}_k = [\phi_1 \quad \phi_2 \quad \cdots \quad \phi_k] \in \mathbb{R}^{4 \times k}, \quad (6-13)$$

and

$$\mathbf{W}_k = \text{diag}(\lambda_{RLS}^{k-1}, \lambda_{RLS}^{k-2}, \dots, \lambda_{RLS}, 1), \quad (6-14)$$

This allows the cost function to be written as

$$J_k(\mathbf{\Theta}) = \|(\mathbf{Y}_k - \mathbf{\Theta} \mathbf{\Phi}_k) \mathbf{W}_k^{1/2}\|_F^2. \quad (6-15)$$

To find the optimal estimate, the derivative of $\mathbf{\Theta}$ is set to zero which gives

$$\hat{\mathbf{\Theta}}_k = \mathbf{Y}_k \mathbf{W}_k \mathbf{\Phi}_k^T (\mathbf{\Phi}_k \mathbf{W}_k \mathbf{\Phi}_k^T)^{-1}$$

This can be rewritten as

$$\hat{\mathbf{\Theta}}_k = \mathbf{S}_k \mathbf{R}_k^{-1} \quad (6-16)$$

where

$$\mathbf{S}_k = \mathbf{Y}_k \mathbf{W}_k \mathbf{\Phi}_k^T = \sum_{i=1}^k \lambda_{RLS}^{k-i} \mathbf{y}_i \phi_i^T. \quad (6-17)$$

and

$$\mathbf{R}_k = \Phi_k \mathbf{W}_k \Phi_k^T = \sum_{i=1}^k \lambda_{RLS}^{k-i} \phi_i \phi_i^T \quad (6-18)$$

Both $\mathbf{R}_k$ and $\mathbf{S}_k$ can be updated recursively

$$\mathbf{R}_k = \lambda_{RLS} \mathbf{R}_{k-1} + \phi_k \phi_k^T, \quad (6-19)$$

$$\mathbf{S}_k = \lambda_{RLS} \mathbf{S}_{k-1} + \mathbf{y}_k \phi_k^T. \quad (6-20)$$

The problem that arises with Equation 6-16 is that for models with large dimensions the inversion of $\mathbf{R}_k$ is very computationally intensive. By defining $\mathbf{P}_k = \mathbf{R}_k^{-1}$ and using the Matrix Inversion Lemma [20], it is possible to rewrite this inversion such that the inversion is done with a scalar instead of matrices.

$$\mathbf{P}_k = \frac{1}{\lambda_{RLS}} \left(\mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1} \phi_k \phi_k^T \mathbf{P}_{k-1}}{\lambda_{RLS} + \phi_k^T \mathbf{P}_{k-1} \phi_k} \right) \quad (6-21)$$

The corresponding recursive least-squares gain can then be written as

$$\mathbf{K}_k = \frac{\mathbf{P}_{k-1} \phi_k}{\lambda_{RLS} + \phi_k^T \mathbf{P}_{k-1} \phi_k}. \quad (6-22)$$

Then finally, this results in $\hat{\Theta}_k$ being written as

$$\hat{\Theta}_k = \mathbf{S}_k \frac{1}{\lambda_{RLS}} \left(\mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1} \phi_k \phi_k^T \mathbf{P}_{k-1}}{\lambda_{RLS} + \phi_k^T \mathbf{P}_{k-1} \phi_k} \right) \quad (6-23)$$

The recursive update law for $\hat{\Theta}_k$ then becomes

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + (\mathbf{y}_k - \hat{\Theta}_{k-1} \phi_k) \mathbf{K}_k^T \quad (6-24)$$

The recursion must be initialized with an initial model estimate and covariance, for example

$$\hat{\Theta}_0 = \begin{bmatrix} \mathbf{G}_{\text{nom}} & \mathbf{b}_{\text{nom}} \end{bmatrix}, \quad \mathbf{P}_0 = p_0 \mathbf{I}_4. \quad (6-25)$$

The value of p_0 expresses the initial uncertainty in the model. A larger p_0 results in larger initial adaptation gains, whereas a smaller value places more confidence in the initial model estimate.

6-1-5 Expected performance and tuning trade-off

The four online least-squares approaches introduce a trade-off between noise rejection, adaptation speed and data richness. A large dataset generally improves noise rejection as individual noisy measurements have less influence on the least-squares estimate. Besides that, it also generally improves the conditioning of the regression matrix Φ_D , provided it has enough variation in α_0 , α_x and α_y . The downside is that larger batches are generally less effective at faster adaptation to time-varying plant behaviour, since they are more dependent on older data. Therefore, cumulative least-squares is expected to give the most robust parameter estimation for stationary model mismatch, while it is expected to adapt slowly when the true model

changes over time. Batch-wise least-squares, is expected to adapt faster to changing model parameters as it forgets old data after each new batch. That also means that its estimates are more sensitive to noisy measurements and poorly excited data. Therefore its performance is expected to be more robust to poor excitation for larger batch sizes. Moving-window least-squares also forgets old data but does so continuously. For identical dataset sizes it is expected to react smoother to changing model parameters than batch-wise least-squares as it only forgets one data sample at a time. Lastly, with an appropriately selected forgetting factor, the recursive least-squares algorithm is expected to adapt the fastest if the true model parameters change as its newest obtained sample is most dominant in the estimation. Unlike the moving-window approach, the RLS algorithm does not discard old data, instead their influence exponentially decays. This provides a smoother trade-off between retaining past information and adapting to recent data. Though sufficiently rich excitation remains necessary for reliable estimation.

For implementation of all four least-squares approaches the reliability not only depends on the size of dataset D , but also the richness of the data. If the input data is too poorly excited, the regression matrix Φ_D can become ill-conditioned resulting in an unreliable estimate of $\mathbf{G}$ and $\mathbf{b}$. In the simulations, the approaches are therefore compared primarily by their tracking accuracy and by the conditioning of the least-squares regression problem.

6-2 Simulations

To explore the expected limitations and advantages of each approach, their performance is simulated in identical simulation environments in Matlab. It must be noted that the simulations are not intended as a high-fidelity WindShaper model. Instead, they provide controlled test cases in which the behaviour of the least-squares adaptation strategies can be isolated. This allows the influence of data-window size, model mismatch, drift, and measurement noise to be studied before experimental validation. The controlled simulation cases are described in Table 6-1, the most important performance metrics in Table 6-2 and for an elaborate overview of the simulation settings, see Appendix E-1. For each case, $\mathbf{G}_{true}$ and $\mathbf{b}_{true}$ are substituted by the corresponding model parameters into Equation 6-26 that are in line with the simulation objective. A sampling time of $dt = 0.75$ s was used.

$$\mathbf{y}_k = \mathbf{G}_{true}\boldsymbol{\alpha}_k + \mathbf{b}_{true,k} + \boldsymbol{\epsilon}_{noise,k} \quad (6-26)$$

6-2-1 Can online least-squares damage performance when the model is already good?

Table 6-3 shows that for the nominal case, the open-loop controller shows a very high overall tracking accuracy. The simulations in this environment show that adaptive least-squares approaches that make use of datasets of preselected size, the batch-wise and moving-window approach, have worse tracking performance for smaller than for larger batch sizes. This is especially visible for the batch-wise update. At smaller batch sizes, the poor tracking coincides with insufficient regressor excitation. For $N_D = 60$, 13 out of 15 updates were rank deficient, while for $N_D = 100$, two out of nine were rank deficient. Overall, the nominal

Case	$\mathbf{G}_{\text{true}}$	$\mathbf{b}_{\text{true},k}$	Purpose
Nominal	$\mathbf{G}_{\text{ID}}$	$\mathbf{b}_{\text{ID}}$	Correct initial model
Bias mismatch	$\mathbf{G}_{\text{ID}}$	$\mathbf{b}_{\text{ID}} + \begin{bmatrix} 0.30 & -0.25 & 0.20 \end{bmatrix}^T$	Offset correction
Gain and bias mismatch	$\mathbf{G}_{\text{ID}} + \Delta\mathbf{G}$	$\mathbf{b}_{\text{ID}} + \begin{bmatrix} 0.25 & -0.15 & 0.20 \end{bmatrix}^T$	Full model adaptation
Slow drift	$\mathbf{G}_{\text{ID}} + \Delta\mathbf{G}$	$\mathbf{b}_{\text{ID}} + \begin{bmatrix} 0.15 \\ -0.10 \\ 0.10 \end{bmatrix} + \begin{bmatrix} 0.30 \\ -0.20 \\ 0.15 \end{bmatrix} \frac{t}{t_f}$	Time-varying mismatch

$$\Delta\mathbf{G} = \begin{bmatrix} 0.018 & -0.006 & 0.010 \\ -0.012 & 0.015 & -0.008 \\ 0.010 & 0.006 & -0.016 \end{bmatrix}, \quad t_f = 675.75 \text{ s.}$$

Table 6-1: Overview of the four different simulation cases.

Metric or setting	Implementation
Overall RMSE	Computed over all three outputs and all 902 samples.
Constant-reference RMSE	$\text{RMSE}_{\text{const}}$ computed over all three outputs using only the samples belonging to the constant-reference part of the reference sequence.
Sinusoidal-reference RMSE	$\text{RMSE}_{\text{sine}}$ computed over all three outputs using only the samples belonging to the sinusoidal-reference part of the reference sequence.
Sensor-wise RMSE	RMSE calculated separately for y_1 , y_2 , and y_3 .
Tracking improvement	$I_{\text{RMSE}} = 100(\text{RMSE}_{\text{open}} - \text{RMSE}_{\text{LS}})/\text{RMSE}_{\text{open}}$.
Conditioning	Median values of $\kappa(\Phi_D)$, together with the median value of $\sigma_{\min}(\Phi_D)$, evaluated at model-update instants.

Table 6-2: Evaluation settings and performance metrics used in the simulations.

case shows that online least-squares has little systematic error to correct when the model is already accurate. Any loss in tracking performance is therefore mainly caused by noisy or ill-conditioned parameter updates. Although tracking performance for small batch sizes is poor, for sufficiently large datasets, the updates become less sensitive and the tracking performance approaches that of the open-loop inverse control law.

6-2-2 Can online least-squares improve tracking when the model is wrong?

Figure 6-2 shows that online model adaptation becomes clearly beneficial when model mismatch is introduced. For all model mismatch cases the open-loop inverse control law performs worse than it did for the nominal case. The main exception is batch-wise least-squares with $D = 60$ samples, which sees poor tracking accuracy go hand in hand with poor regressor conditioning, see Figures 6-3 and 6-4. This demonstrates that online model adaptation can

Table 6-3: Simulation results for the nominal plant case, where the identified model parameters are equal to the true model parameters. For RLS, the conditioning is evaluated using the exponentially weighted regressor $\Phi_k \mathbf{W}_k^{-1/2}$. Rank-deficient evaluations correspond to regressors with rank smaller than four.

Algorithm	N_D / λ	RMSE	RMSE _{const}	RMSE _{sine}	median(κ_Φ)	Rank deficient / evaluations
Open-loop inverse	–	0.043	0.043	0.042	–	–
Cumulative LS	–	0.047	0.048	0.042	198.0	1 / 843
Batch-wise LS	60	0.826	0.807	0.891	1.01×10^{17}	13 / 15
Batch-wise LS	100	0.229	0.258	0.048	825.7	2 / 9
Batch-wise LS	140	0.044	0.044	0.044	300.4	0 / 6
Batch-wise LS	180	0.045	0.046	0.044	285.9	0 / 5
Batch-wise LS	220	0.043	0.043	0.045	271.1	0 / 4
Moving-window LS	60	0.061	0.060	0.065	3305.5	1 / 843
Moving-window LS	100	0.053	0.053	0.056	592.0	1 / 803
Moving-window LS	140	0.044	0.045	0.042	336.6	0 / 763
Moving-window LS	180	0.043	0.044	0.042	281.7	0 / 723
Moving-window LS	220	0.043	0.043	0.042	257.6	0 / 683
RLS	$\lambda = 0.95$	0.046	0.046	0.044	973.8	1 / 843
RLS	$\lambda = 0.98$	0.044	0.044	0.042	396.8	1 / 843

improve tracking accuracy under model mismatch, provided that the data used for every update is sufficiently informative. It also confirms that a time-varying reference does not guarantee useful identification data by itself. The resulting control inputs must sufficiently excite the independent regressor directions.

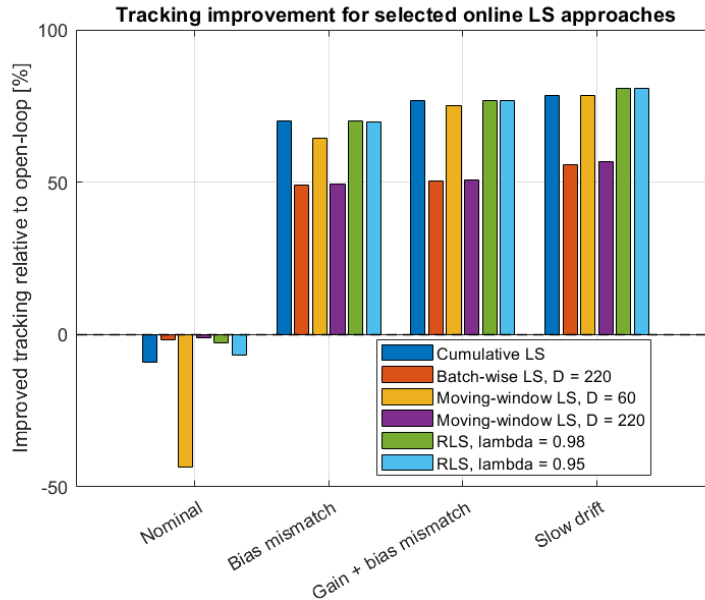


Figure 6-2: overall RMSE improvement I_{RMSE} of selected adaptive least-squares approaches relative to open-loop inverse control for the four simulation cases. Positive values indicate improved tracking, whereas negative values indicate degraded tracking.

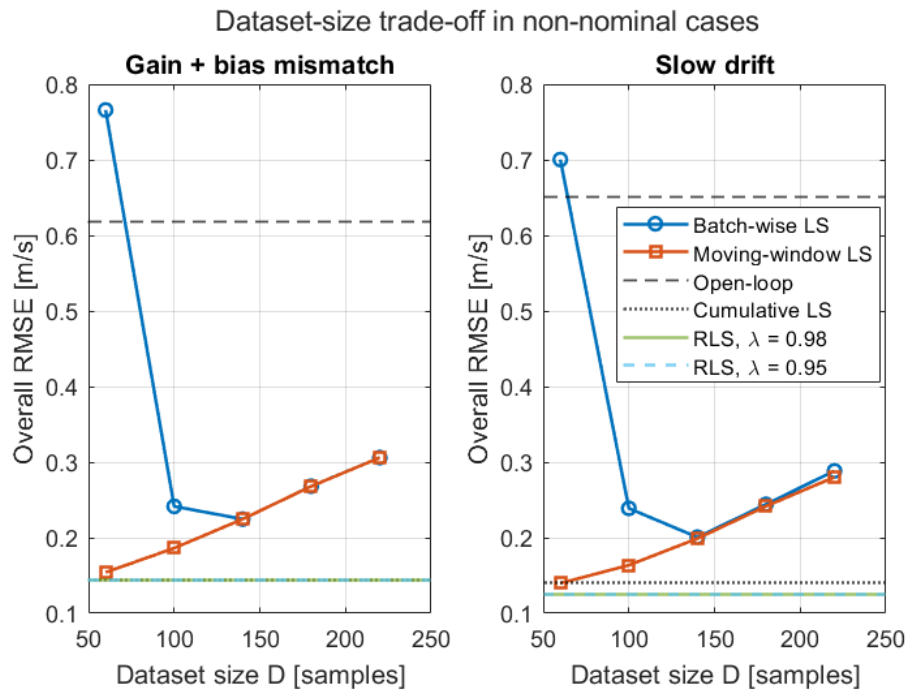


Figure 6-3: Overall RMSE versus dataset size for batch-wise and moving-window least-squares in the gain-and-bias-mismatch and slow-drift cases. The open-loop inverse controller, cumulative least-squares, and RLS results are included as reference levels. The opposing trends of the batch-wise and moving-window approaches illustrate the different role of dataset size in the two estimators.

6-2-3 What does the dataset size trade-off look like?

Figures 6-3 and 6-4 show how the dataset size affects batch-wise and moving-window least squares differently. The batch-wise approach sees a substantial increase in tracking accuracy for increasing batch-sizes in the non-nominal cases, although the tracking performance slightly deteriorates beyond approximately 140 samples, which may result from the reduced adaptability associated with larger batches.

The moving-window approach, on the other hand, shows decreasing tracking accuracy as the dataset size increases. Smaller windows generally improve tracking accuracy since it allows for faster adaptation to a changing plant because it is based more strongly on recent data. Though an increasing dataset size improves conditioning, it also uses more outdated data samples in its estimate. This shows that better conditioning does not necessarily result in more accurate tracking.

Very poor tracking for the batch-wise approach with $D = 60$ samples can be related to the way the reference sequence is designed. A batch of 60 samples corresponds to 45 s and therefore contains about 2.25 of the 20 s reference steps. This provides only a limited number of distinct input directions within a single batch. This can explain why the median value for those batch sizes in Figure 6-4 is extremely high.

Overall, this shows that the dataset size has a different role for the two approaches. In the batch-wise approach, the dataset size mainly determines whether each periodic update

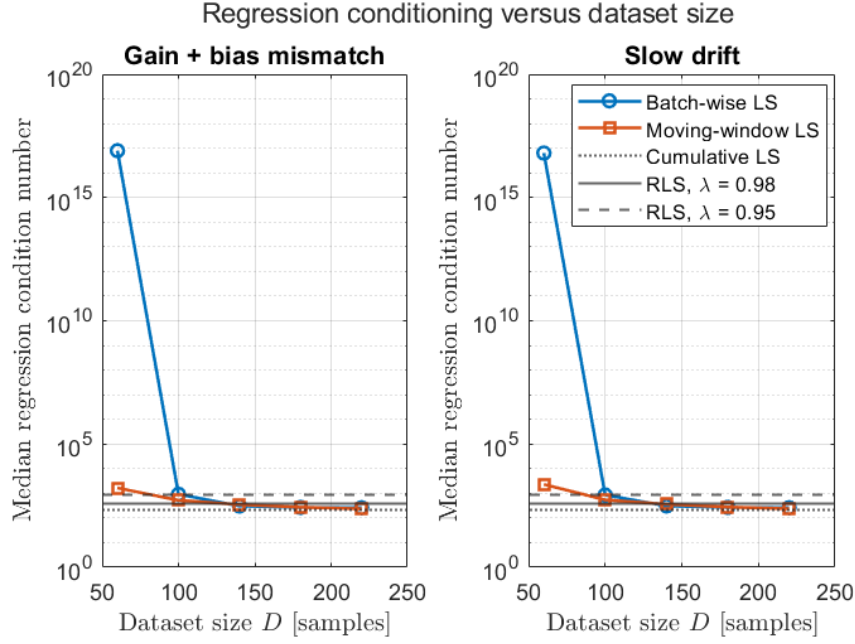


Figure 6-4: Median condition number of the regression matrix Φ_D at model-update instants versus dataset size for batch-wise and moving-window least-squares in the gain-and-bias-mismatch and slow-drift cases. RLS and cumulative results are included as reference levels, and the vertical axis is logarithmic.

is reliable. In the moving-window approach, the dataset size shows the trade-off between a well-conditioned estimate and a more accurate local estimate. The moving-window approach therefore offers a useful trade-off for slowly changing model errors, provided that ill-conditioned updates are detected and potentially rejected.

6-2-4 What is the effect of the regularisation terms?

Regularisation is used both in the inverse control law and in the model estimation. In the least-squares estimate, it ensures numerical stability for matrix inversion when the regressor becomes singular or nearly singular. It can also help attenuation error amplification in weakly excited directions. Although this only becomes significant when the corresponding squared singular values are comparable or smaller than the regularization term.

For the unregularised case, the derivation of the norm of the parameter estimation error is derived as follows. By substituting $\mathbf{Y} = \Theta\Phi + \mathbf{E}$ into the general unregularised least-squares solution gives

$$\hat{\Theta} = (\Theta\Phi + \mathbf{E})\Phi^T(\Phi\Phi^T)^{-1}$$

Assuming Φ has full rank, the parameter estimation error is then written as

$$\hat{\Theta} - \Theta = \mathbf{E}\Phi^\dagger \quad (6-27)$$

The two norm of $\Phi^\dagger$ is defined as

$$\|\Phi^\dagger\|_2 = \frac{1}{\sigma_{\min}(\Phi)}$$

This results in the following bound

$$\|\hat{\Theta} - \Theta\|_2 \leq \frac{\|E\|_2}{\sigma_{\min}(\Phi)} \quad (6-28)$$

This bound shows that the residual term can be strongly amplified for very small singular values. By adding the regularisation term the estimate can be written as

$$\hat{\Theta}_\rho = Y\Phi^T(\Phi\Phi^T + \rho_\Phi I)^{-1}$$

Substituting in $Y = \Theta\Phi + E$ gives

$$\hat{\Theta}_\rho - \Theta = -\rho_\Phi \Theta(\Phi\Phi^T + \rho_\Phi I)^{-1} + E\Phi^T(\Phi\Phi^T + \rho_\Phi I)^{-1} \quad (6-29)$$

Using the singular value decomposition, the regularised inverse of Φ can be written as

$$\Phi^T(\Phi\Phi^T + \rho_\Phi I)^{-1} = V_\Phi \text{diag}\left(\frac{\sigma_i}{\sigma_i^2 + \rho_\Phi}\right)U_\Phi^T$$

Without regularisation, the amplification factor would be $1/\sigma_i$. By regularising, whenever $\sigma_i^2 < \rho_\Phi$, these directions get attenuated by ρ_Φ while directions for which $\sigma_i^2 > \rho_\Phi$ remain amplified. Therefore, for certain bounds in weakly informed directions, the estimation residual gets attenuated. However this must not be confused with recovering information that is absent from the data. While ρ_Φ can have a suppressing effect on the residual, Equation 6-29 shows that ρ_Φ also results in a bias term on the estimation error.

In the simulations, for the cumulative, batch-wise and moving-window approach $\rho_\Phi = 10^{-6}$ was used. This means regularization only becomes significant for singular values of approximately 10^{-3} or smaller. The RLS singular values are also included to analyse the excitation of the weighted regressor. The minimum median values in table 6-4 show substantially higher values than this threshold. This implies that the regularisation has very little influence during typical model updates. In some cases however, rank deficient updates do occur, explaining the near zero minimum singular values. In these cases, the regularization term acts as an enforcer of inverting the regressor and ensure numerical stability. Its primary role during these simulations is therefore more being a safeguard rather than reducing residual amplification.

The same regularization logic applies to the inverse control law

$$\alpha_k = (\hat{G}_k^T \hat{G}_k + \rho_G I)^{-1} \hat{G}_k^T (r_{k+d} - \hat{b}_k)$$

The singular value decomposition of G is written as

$$\hat{G} = U_G \Sigma_G V_G^T$$

then the regularised inverse becomes

$$(\hat{G}_k^T \hat{G}_k + \rho_G I)^{-1} \hat{G}_k^T = V_G \text{diag}\left(\frac{\sigma_{G,i}}{\sigma_{G,i}^2 + \rho_G}\right)U_G^T$$

The role of the regularisation in the inverse control law is therefore to suppress large control actions in directions where $\hat{G}$ has small singular values.

Table 6-4: Worst-case smallest-singular-value and rank diagnostics across the nominal, bias-mismatch, gain-and-bias-mismatch, and slow-drift simulations. The minimum median is the lowest case-wise median value of $\sigma_{\min}(\Phi)$, whereas the overall minimum is the smallest value observed at any evaluated model-update instant. For RLS, Φ denotes the exponentially weighted regressor $\Phi_k W_k^{1/2}$.

Approach	Minimum median $\sigma_{\min}(\Phi)$	Overall minimum $\sigma_{\min}(\Phi)$	Rank deficient / evaluations
Cumulative LS	3.52	2.47×10^{-15}	4 / 3372
Batch-wise LS, $N_b = 220$	1.75	1.64	0 / 16
Moving-window LS, $N_w = 60$	7.98×10^{-2}	2.47×10^{-15}	4 / 3372
Moving-window LS, $N_w = 220$	1.81	0.857	0 / 2732
RLS, $\lambda = 0.98$	0.603	1.58×10^{-15}	4 / 3372
RLS, $\lambda = 0.95$	0.162	8.43×10^{-16}	4 / 3372

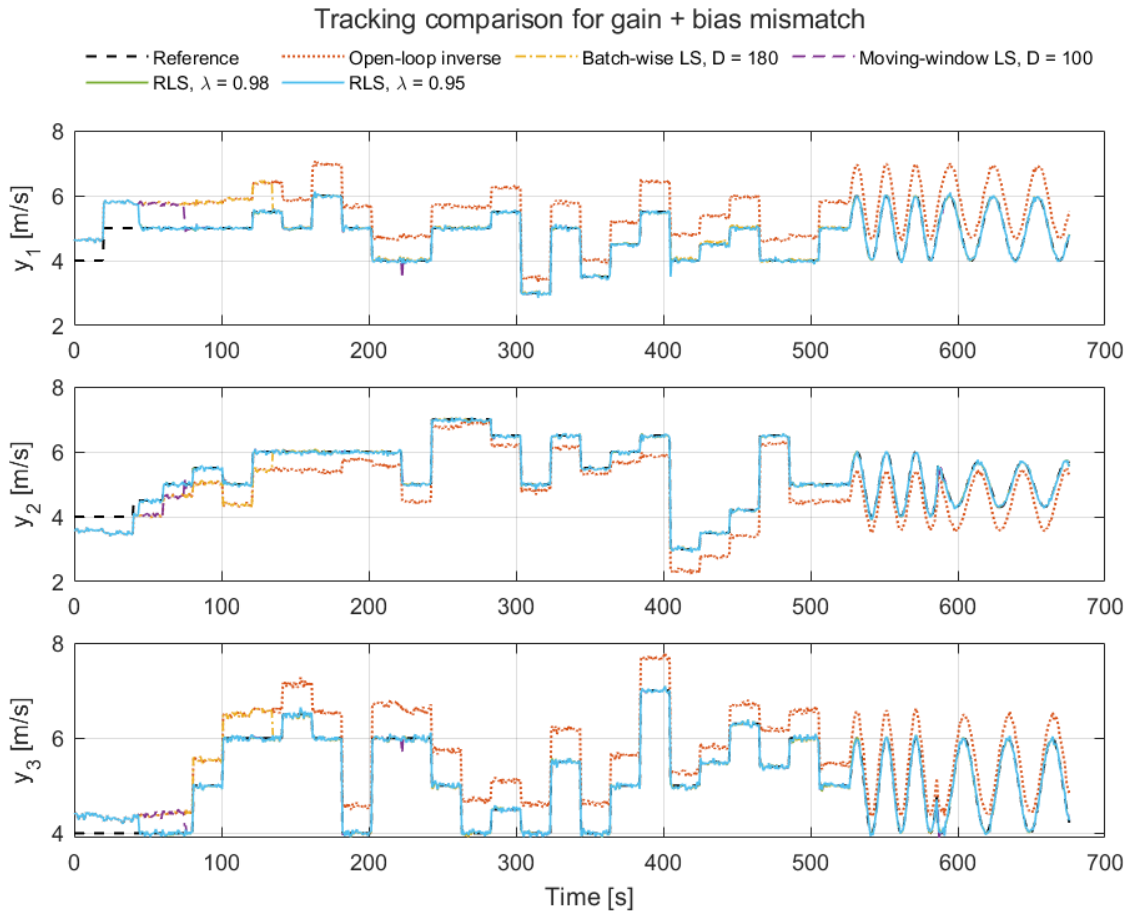


Figure 6-5: Reference tracking trajectories for the gain and bias mismatch simulations, comparing open-loop inverse control with selected adaptive least-squares approaches for the three measured outputs.

6-2-5 Takeaways for experimental validation

Figure 6-5 gives a visualisation of the proof that adaptive least-squares algorithms actually have the potential to reduce tracking errors. Other than that, the results of these simulations offer useful insights for an experimental validation phase. The following takeaways will be

taken into account for those experiments.

- **Unreliable updates:** Rank-deficient and strongly ill-conditioned regressors occurred for insufficiently excited datasets. Both regressor rank and conditioning should therefore be monitored.
- **Initial excitation phase:** To ensure there is enough variability in the regression matrix from the start. The input signal should be excited such that initial estimates of $\mathbf{G}$ and $\mathbf{b}$ are trustworthy.
- **Large datasets for Batch-wise:** The dataset must be sufficiently large to obtain informative full-rank regressors. However, unnecessarily large batches reduce adaptability, so dataset size must balance excitation and adaptation speed.
- **Window-size tuning for moving-window LS:** smaller windows can improve adaptation to changing model parameters, but require extra attention towards conditioning to take notice of unreliable estimates from poorly excited data.
- **Conditioning is a safeguard, not a performance objective:** lower $\kappa(\Phi_D)$ indicates a more reliable estimation problem, but the best-conditioned dataset does not necessarily give the best tracking when the plant changes over time.

6-3 Experimental Validation Setup

The simulations in Section 6-2 were used to investigate under which conditions online least-squares estimation can improve closed-loop tracking. In particular, the simulations examined the reliability of the parameter estimates, the influence of the dataset size and the amount of excitation required for informative estimates. The simulations introduced two central questions: whether online least-squares can damage tracking performance when the model is already good, and whether online least-squares can improve tracking performance when the model is wrong.

The experimental validation aims to answer these questions under two separate system configurations: a nominal configuration (NC) and a configuration with enforced model mismatch (EMM). In the simulations, the nominal case represented the system where $G_0 = G_{ID}$ and $b_0 = b_{ID}$. In these experiments, the nominal configuration corresponds to the same sensor-rake location as was used for identifying the initial system. However, the system was reidentified with respect to Section 2-4 because the prior downstream location could not be precisely established. The corresponding identified model parameters are:

$$G_{ID} = \begin{bmatrix} 0.10874723 & 0.01051375 & 0.08157615 \\ 0.10871767 & 0.06459176 & -0.04993271 \\ 0.12169866 & -0.04846226 & -0.05842624 \end{bmatrix}, \quad b_{ID} = \begin{bmatrix} 1.32021694 \\ 1.63022557 \\ 1.58919714 \end{bmatrix}$$

The identified input-output map has a condition number of $G_{ID} = 2.4760$ and its smallest singular value is $\sigma_{min} = 0.0799$. An input-output delay of 1.5 s (2 samples) was selected empirically and included in the inverse control law. For both experimental configurations, the adaptive least-squares algorithms use G_{ID} and b_{ID} as their initial estimates. The model

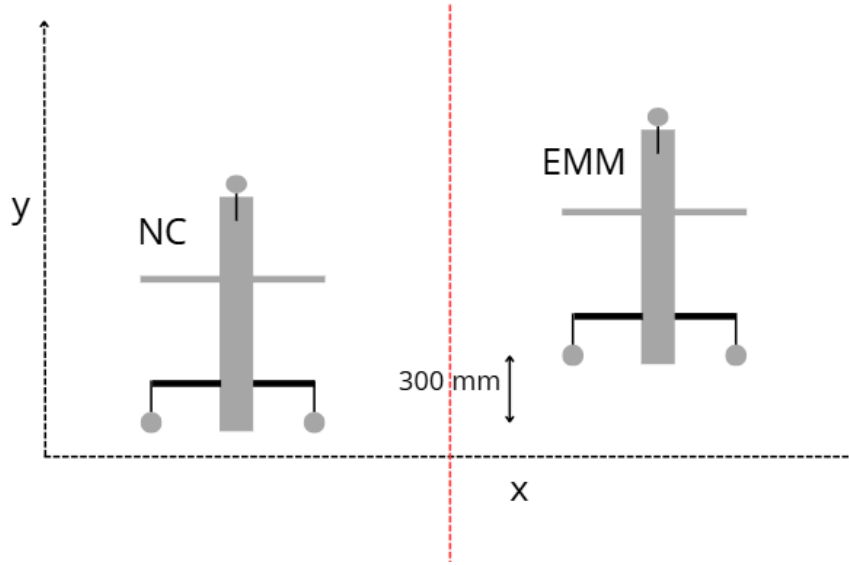


Figure 6-6: Sensor-rake configurations used for experimental validation. The Nominal Configuration (NC) uses the location at which the initial model was identified, while the Enforced Model Mismatch (EMM) configuration displaces the rake by 300 mm without reidentifying the model, thereby intentionally introducing plant-model mismatch.

estimation uses a regularisation term of $\rho_{\Phi} = 1E - 6$ for the cumulative, batch-wise and moving-window approach, while the inverse control law uses $\rho_G = 1E - 4$. For the RLS algorithm, the singular values of the weighted regressor are also computed to assess its excitation.

The two experimental configurations only differ in the sensor-rake location. The EMM configuration is translated upwards 300 mm with respect to the NC. This enforced model mismatch is visualised in Figure 6-6. By not reidentifying the model parameters at this new robot arm location, mismatch is forced onto the model.

Each adaptive least-squares algorithm was run five times per configuration using identical estimation and controller settings. Five experimental runs were chosen since that was achievable in the available time. The open-loop inverse controller was run three times per configuration under identical settings. For each algorithm and run, the same reference sequence was used. The same input constraints were applied as were initially done for the OFO algorithm in Section 5-1. If the inputs exceeded these constraints, the inputs were clipped accordingly.

The estimator settings for each approach were informed by the simulation results. For the batch-wise least-squares, the simulations showed that larger batch-sizes are required for accurate tracking and reliable updates. Therefore, a batch-size of 180 samples is selected. The moving-window approach showed that smaller window-sizes result in faster adaptation to changing model parameters. To balance adaptation speed with estimation reliability a window-size of 100 samples is selected for validation. For a full overview of the different adaptive least-squares settings, see table 6-5. Compared with the reference sequence used in the simulation, more variability is added throughout the run and initial input excitation is enforced by applying a multi-sine to the initial part of the reference.

Table 6-5: Algorithm-specific settings used in the experimental validation.

Approach	Dataset and update rule	Settings
Open-loop inverse control	The offline identified model ($\mathbf{G}_{\text{ID}}, \mathbf{b}_{\text{ID}}$) remains fixed throughout the run.	No online parameter updates.
Cumulative LS	All valid input–output pairs collected since the start of the run are retained. After initialization, the model is updated after every new valid sample.	Minimum initialization dataset: $N_{\min} = 180$ samples.
Batch-wise LS	The model is estimated from non-overlapping batches. An update is performed after each complete batch, after which the data are discarded.	Batch size: $N_B = 180$ samples (135 s).
Moving-window LS	Only the most recent input–output samples are retained. Once the window is full, the model is updated after every new valid sample.	Window size: $N_w = 100$ samples (75 s).
Recursive LS	After the initial $N_{\min}$ updates, the estimates are used in the inverse control law at every proceeding time step. Previous data are exponentially discounted using the forgetting factor.	$\lambda_{\text{RLS}} = 0.98$, $\mathbf{P}_0 = \mathbf{I}_4$, $N_{\min} = 60$ samples.

Experimental Validation

Chapter 6 introduced online model adaptation as a means of reducing the dependence of the inverse control law on the initially identified input-output model. This chapter will experimentally evaluate the different adaptive control architectures. First, Section 7-1 will analyse the tracking performance of the different adaptive least-squares algorithms for the *NC*. Section 7-2 will follow by analysing the same algorithms for the *EMM* configuration. Finally, Section 7-3 will assess run-to-run repeatability for the RLS approach for the *EMM* configuration.

7-1 Nominal Configuration

The nominal configuration evaluates whether the adaptive least-squares algorithms degrade tracking performance when the initial model already provides a reasonably accurate representation of the system. The open-loop inverse controller acts as a baseline, after which the different adaptive least-squares algorithms are compared with the obtained baseline for the same reference sequence. Figure 7-1 shows the *RMSE* of the open-loop controller and the different adaptive least-squares approaches, while Table 7-1 summarizes the overall RMSE and RMSE per constant and periodic parts of the reference. The results show that the inverse control law using the identified model is able to track the references at sensors y_1 and y_2 relatively accurately, whereas sensor y_3 has the most dominant contribution to the tracking error. Consequently, most of the improvements obtained through online model adaptation occur at sensor y_3 . Table 7-1 shows that all the adaptive least-squares algorithms are able to improve overall tracking. The batch-wise approach has the smallest improvement, reducing RMSE by 13.9%. Even though the cumulative and moving-window approach differ in operating conditions, they have a similar improvement, reducing RMSE by 19.3% and 19.0%, respectively. The cumulative approach has a lower RMSE for the constant part of the reference, whereas the moving-window approach has a lower RMSE for the periodic part of the reference.

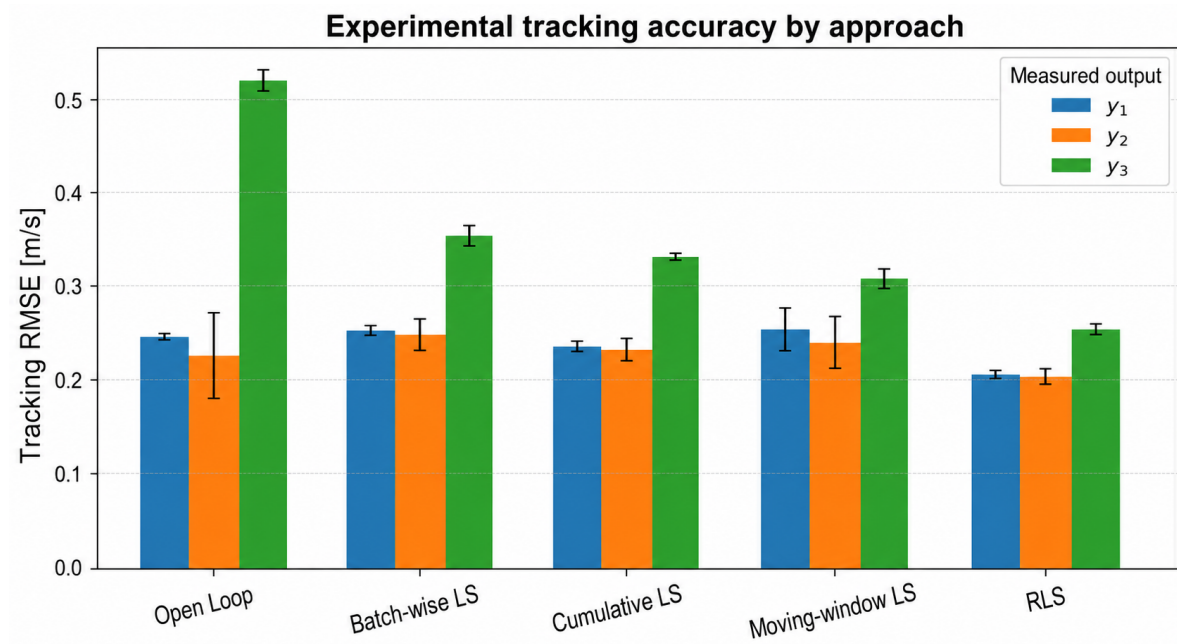


Figure 7-1: Sensor-specific RMSE comparison for the open-loop inverse controller and adaptive least-squares approaches under the nominal configuration. Bars show the mean across three open-loop runs and five runs for each adaptive approach; error bars denote one standard deviation across runs.

Table 7-1: Tracking performance of the inverse controller and adaptive least-squares approaches for the nominal case. The first 10 s of each run are excluded to remove the initial transient.

Approach	Overall RMSE [m s ⁻¹]	Constant RMSE [m s ⁻¹]	Periodic RMSE [m s ⁻¹]	Improvement [%]
Open-loop inverse	0.331	0.325	0.333	–
Cumulative LS	0.267	0.245	0.282	19.3
Batch-wise LS, $N_D = 180$	0.285	0.277	0.291	13.9
Moving-window LS, $N_D = 100$	0.268	0.284	0.248	19.0
RLS, $\lambda_{\text{RLS}} = 0.98$	0.221	0.217	0.224	33.2

Recursive least-squares achieves the lowest overall RMSE of 0.221 [m s⁻¹], with an overall improvement of 33.2% with respect to the open-loop inverse. Furthermore, Figure 7-1 shows that only the RLS algorithm is able to improve RMSE tracking at all three sensor locations.

Table 7-2 summarizes the conditioning of the regressor matrix and the estimated input-output map over all five runs of each adaptive least-squares approach. Although no rank deficient updates occurred across all approaches, the relatively high regressor condition number suggests that not all directions were equally excited. The cumulative approach had the best conditioned regressors, with a maximum condition value of 289. The moving-window approach had lesser conditioned regressors, with a maximum condition value of 9257. This shows that despite remaining full rank, some moving-window updates were poorly conditioned and therefore highly sensitive to estimation errors in the weakly excited directions.

Table 7-2: Regressor and estimated-gain conditioning metrics for the nominal case, evaluated over all five runs of each adaptive least-squares algorithm.

Approach	Rank-deficient updates	$\min \sigma_{\min}(\Phi)$	$\max \kappa(\Phi)$	$\min \sigma_{\min}(\hat{G})$	$\max \kappa(\hat{G})$
Batch	0	0.44	1039.0	0.0799	3.4325
Cumulative	0	1.54	289	0.0799	3.3895
Moving-window	0	0.03	9257.0	0.0047	53.9273
RLS	0	0.18	1276	0.0230	7.9182

The effect of regularisation can be assessed by comparing $\sigma_{\min}^2(\Phi)$ with $\rho_{\Phi} = 10^{-6}$. The moving-window approach had the lowest observed singular value at 0.03. In this case $\sigma_{\min}^2(\Phi) = 9 \times 10^{-4}$, which remains significantly higher than ρ_{Φ} . Therefore the regularisation term had a negligible effect on ensuring numerical stability for parameter estimates and acted primarily as a safeguard.

The estimated input-output matrices were more poorly conditioned than the originally identified model parameters, for which $\kappa(\mathbf{G}_{ID}) = 2.4760$. This is consistent with $\mathbf{G}_{ID}$ being fitted from a richer dataset than what is available within the online adaptation windows. The largest increase in condition number was observed for the moving window, with $\kappa(\hat{\mathbf{G}}) = 53.9273$. The effect of regularisation on the inverse control law is determined by comparing $\sigma_{\min}^2(\hat{\mathbf{G}})$ with $\rho_G = 10^{-4}$. Only for the moving-window approach was $\sigma_{\min}^2(\hat{\mathbf{G}}) < \rho_G$ observed. Without regularisation, the gain in the corresponding direction caused by $\sigma_{\min}$ would be

$$\frac{\sigma_{\min}}{\sigma_{\min}^2} = \frac{1}{\sigma_{\min}} = 212.77$$

With regularisation, the corresponding gain would be

$$\frac{\sigma_{\min}}{\sigma_{\min}^2 + \rho_G} = 38.50.$$

This shows that regularisation has a significant effect in suppressing large control actions in the direction of $\sigma_{\min}$. Figure 7-2 shows the ensemble mean and standard deviation of the RLS response across all five experimental runs. The low constant and periodic RMSE values indicate accurate absolute tracking, while the relatively narrow standard deviation across runs indicates the consistent behaviour between the five experiments. This is also in line with the low constant and periodic RMSE values found in Table 7-1.

Overall, the nominal experiments show that online model adaptation remains beneficial for improving tracking accuracy, even when the initial estimate represents the input-output behaviour relatively well. All adaptive modelling approaches reduced the overall RMSE, with the RLS algorithm achieving the largest improvement and improving tracking at every sensor location. Nevertheless, higher estimator adaptability does not automatically result in higher tracking performance. The moving-window approach had the worst conditioned estimate and did not outperform the cumulative approach.

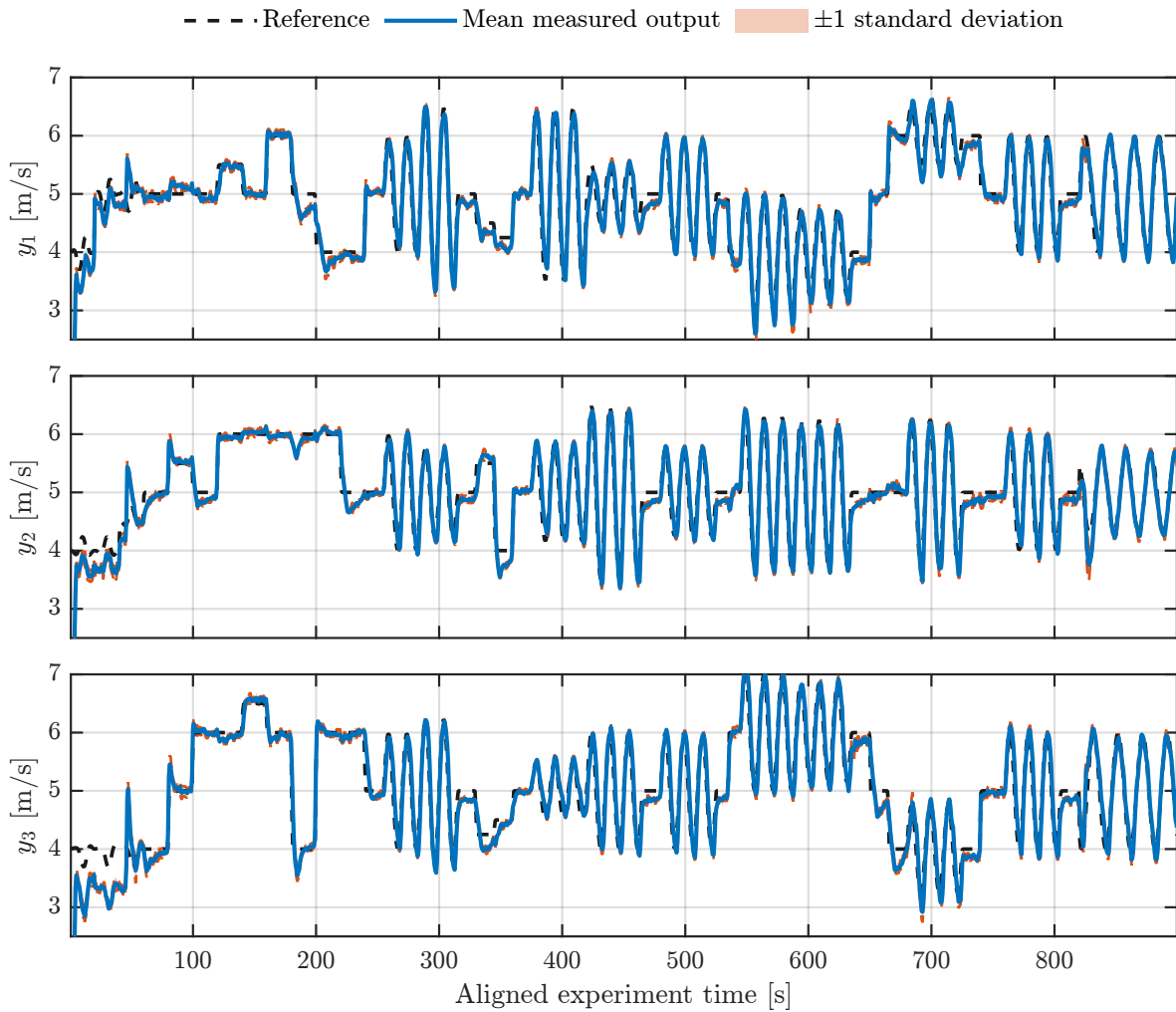


Figure 7-2: Ensemble-mean tracking trajectory of the RLS controller across five runs for the *NC* case. The shaded region denotes ± 1 standard deviation across runs.

7-2 Model Mismatch

The *EMM* configuration aims to answer the question whether online model adaptation can improve tracking performance when the initial model does not accurately represent the system. The same identified model, open-loop inverse controller, adaptive least-squares approaches and reference sequence are used as in the nominal configuration. Model mismatch is enforced by translating the robot arm 300 mm upward in the *y*-direction.

Figure 7-3 shows that the model mismatch has a large negative effect on the tracking performance for the open-loop controller. The overall RMSE is increased from 0.331 m/s in the *NC* to 0.510 m/s in the *EMM* configuration. The degradation is especially visible for sensor y_1 . This demonstrates that the initially identified model is not able to accurately capture the input-output behaviour for the displaced measurement case.

All the adaptive least-squares algorithms are able to improve overall tracking and even do so at every sensor location. Their across run performance is summarized in Table 7-3. The

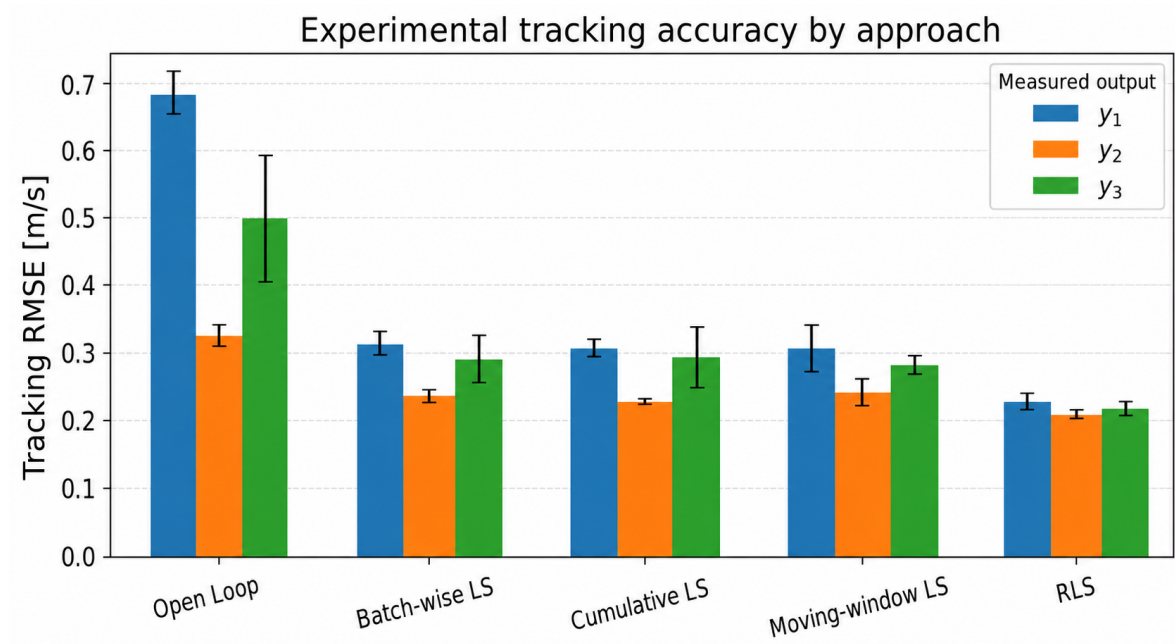


Figure 7-3: Sensor-specific RMSE comparison for the open-loop inverse controller and the adaptive least-squares approaches under enforced model mismatch. The bars show the mean over three open-loop runs and five runs for each adaptive approach. Error bars indicate one standard deviation across runs.

cumulative, batch-wise and moving window least-squares have approximately the same overall *RMSE* tracking value. Where the moving-window and batch-wise approach have more accurate tracking for the periodic part of the reference, the cumulative approach has more accurate tracking for the constant part of the reference. The recursive least-squares algorithm outperforms the other approaches and achieves an overall RMSE of 0.21 m/s. This corresponds to an overall improvement of 57.5% with respect to the open-loop controller.

Table 7-4 summarizes the conditioning of the regressor matrices and the estimated input-output matrices for the *EMM* configuration. Similar to the *NC*, no rank deficient updates were observed. The maximum regressor condition number was obtained for the moving-window approach, with 4316, and the lowest for the cumulative approach, with 274. The significance of the regularization term used for the model estimation is assessed by comparing

Table 7-3: Tracking performance of the inverse controller and adaptive least-squares approaches for the *EMM* configuration. The first 10 s of each run are excluded to remove the initial transient.

Approach	Overall RMSE [m s ⁻¹]	Constant RMSE [m s ⁻¹]	Periodic RMSE [m s ⁻¹]	Improvement [%]
Open-loop inverse	0.510	0.521	0.496	–
Cumulative LS	0.275	0.265	0.278	46.1
Batch-wise LS, $N_D = 180$	0.278	0.281	0.271	45.5
Moving-window LS, $N_D = 100$	0.275	0.278	0.268	46.1
RLS, $\lambda_{\text{RLS}} = 0.98$	0.217	0.210	0.223	57.5

Table 7-4: Regressor and estimated-gain conditioning metrics for the *EMM* configuration, evaluated over all five runs of each adaptive least-squares algorithm.

Approach	Rank-deficient updates	$\min \sigma_{\min}(\Phi)$	$\max \kappa(\Phi)$	$\min \sigma_{\min}(\hat{G})$	$\max \kappa(\hat{G})$
Batch-wise LS	0	0.51	941	0.0799	3.5572
Cumulative LS	0	1.54	274	0.0782	3.5044
Moving-window LS	0	0.08	4316	0.0009	246.5712
RLS	0	0.21	1107	0.0263	9.6914

$\sigma_{\min}^2(\Phi)$ with $\rho_{\Phi} = 10^{-6}$. The smallest singular value was 0.08 for the moving-window approach, giving $\sigma_{\min}^2(\Phi) = 6.4 \times 10^{-3}$. This remains larger than ρ_{Φ} , therefore, the regularisation has a negligible effect on the parameter estimation and acted only as a safeguard.

The obtained input-output estimates for the cumulative and batch-wise approaches remained well-conditioned with values close to the originally identified $\mathbf{G}_{ID}$. The moving-window approach obtained the highest condition number with 246.5712, and with 0.0009 the smallest singular value. This suggests the presence of excessive control inputs in this singular direction when no regularization is applied.

The effect of regularisation in the inverse control law is determined by comparing $\sigma_{\min}^2(\hat{G})$ with $\rho_G = 10^{-4}$. Only for the moving-window was $\sigma_{\min}^2(\hat{G})$ found to be smaller than $\rho_G = 10^{-4}$. Without regularisation, the gain in this smallest singular direction would be

$$\frac{\sigma_{\min}}{\sigma_{\min}^2} = \frac{1}{\sigma_{\min}} = 1.11 \times 10^3$$

With regularisation, the gain in this same direction would be

$$\frac{\sigma_{\min}}{\sigma_{\min}^2 + \rho_G} = 8.93.$$

This shows the effect of the regularisation term on attenuating large control actions in the direction corresponding to this singular value.

Although the enforced model mismatch increased the open-loop RMSE from 0.331 to 0.510 m/s, RLS achieved nearly identical absolute tracking performance in the *NC* and *EMM* configurations, with overall RMSE values of 0.221 and 0.217 m/s, respectively. This indicates that the recursive adaptation largely compensated for the imposed model mismatch.

Overall, the model mismatch configuration shows the real benefit of applying online model adaptation algorithms. Every adaptive least-squares algorithm is able to improve tracking performance for every sensor with respect to the open-loop controller. The RLS algorithm outperforms the other approaches and achieves approximately the same tracking error as was observed for the nominal configuration. For the average trajectory of the RLS algorithm for the *EMM* configuration, see Figure 7-4. This visualizes how the RLS algorithm can largely compensate for the enforced model mismatch.

These results reinforce the finding that faster adaptation does not automatically result in more accurate tracking. The moving-window approach, while being more adaptive than the batch wise and cumulative approach, has more unreliable estimates and does not outperform the

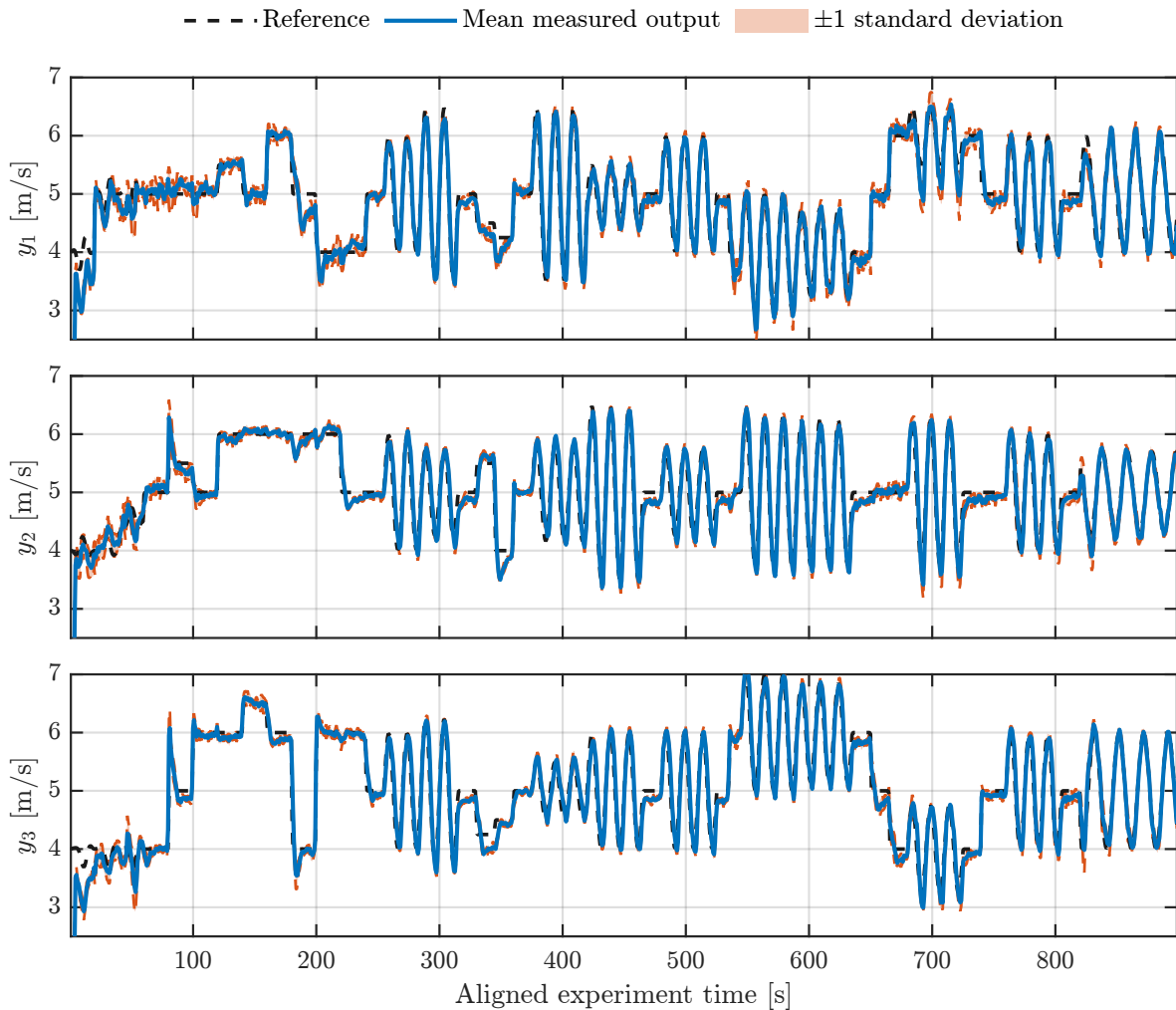


Figure 7-4: Ensemble-mean tracking trajectory of the RLS controller across five runs for the *EMM* configuration. The shaded region denotes ± 1 standard deviation across runs.

other approaches. The RLS approach has the highest tracking accuracy while having lesser regressor conditioning than the batch-wise and cumulative approach but better regressor conditioning than the moving-window approach. The RLS algorithm therefore provided the best combination of tracking performance and estimator conditioning of all the investigated approaches.

7-3 Closed-Loop Recursive Least-Squares Repeatability Assessment

Having determined that the RLS algorithm is the best performing adaptive modelling approach across the investigated operating conditions, this section will assess whether its closed-loop response can be reproduced across independent runs under the *EMM* configuration. For this configuration, five additional experiments were done to get a total of ten experimental runs for a run-to-run repeatability assessment of the closed-loop recursive least-squares algorithm.

Repeatability will be assessed in accordance with the repeatability limit defined by the American Society for Testing and Materials (ASTM) International [21]. ASTM E177 defines the repeatability limit r , such that the difference between runs is expected to be lower than r with a probability of approximately 95%. This limit is commonly approximated as $r \approx 2.8s_r$, where s_r is the repeatability standard deviation.

However, since they define the repeatability assessment based on the difference between runs, and the reference sequence is time-varying in this case, looking at the absolute difference between runs is not informative. Instead, the repeated experimental runs are aligned on a common time grid and repeatability is then evaluated point-wise for each sensor of the *EMM* configuration.

At each time instant k , the repeatability standard deviation is calculated across the ten runs independently for each sensor j as

$$s_{r,j}(k) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_{i,j}(k) - \bar{y}_j(k))^2}, \quad n = 10, \quad (7-1)$$

where $y_{i,j}(k)$ is the measured velocity at sensor j in run i , and $\bar{y}_j(k)$ is the corresponding mean velocity across the ten runs. The pointwise repeatability limit for sensor j is then

$$r_{\text{rep},j}(k) = 2.8s_{r,j}(k). \quad (7-2)$$

For ten repeated runs,

$$\binom{10}{2} = 45 \quad (7-3)$$

unique pairwise differences are evaluated at each time instant and for each sensor separately. A difference between runs p and q at sensor j satisfies the pointwise repeatability limit when

$$|y_{p,j}(k) - y_{q,j}(k)| \leq r_{\text{rep},j}(k). \quad (7-4)$$

For an assessment containing N time samples, this results in $45N$ pairwise comparisons per sensor. If at least 95% of these comparisons satisfy the repeatability limit, the measurements at that sensor are regarded as repeatable. The ten repeated runs and their ensemble mean are given in Figure 7-5.

To analyse repeatability across different operating conditions, three complementary assessments are performed. First, the repeatability limit is applied to the whole experimental run, including start-up and other transient parts of the experiment. Secondly, the first 60 samples of each experimental run are discarded, along with the first five seconds of each step. Lastly, the first 60 samples of each experimental run are discarded and the first whole period of each periodic reference is discarded. These last two assessments are to determine whether repeatability changes by excluding within-run transients.

Table 7-5 shows the repeatability coverage for these three assessments. It shows that even when transient effects are included, the flow satisfies the ASTM-style point-wise coverage criterion across all experimental runs. By excluding transient effects a slight increase in repeatability coverage can be observed. However, satisfaction of the point-wise coverage criterion is demonstrated regardless. The observed pairwise differences are therefore generally

consistent with the repeatability limit obtained from the ten repeated experiments. These coverage percentages alone do not tell whether the run-to-run spread is small or large in terms of the experiments. Apart from repeatable flow, the closed-loop controller must be able to reliably track the desired reference. To evaluate whether the pairwise variations are

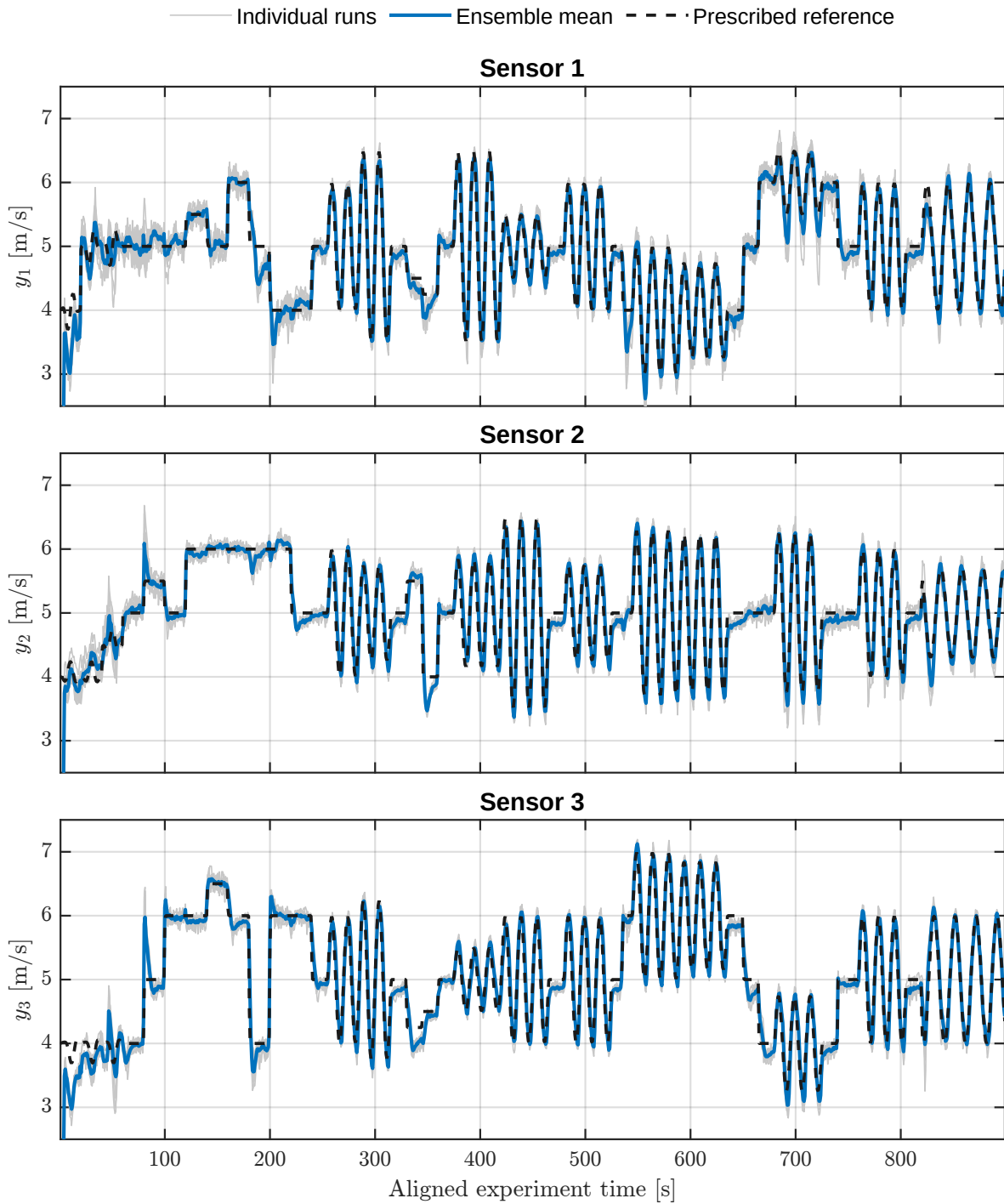


Figure 7-5: Ten repeated runs for the RLS algorithm in the *EMM* configuration.

Table 7-5: ASTM-style pairwise repeatability results for the three aggregate assessments. Coverage is the percentage of pair-time comparisons satisfying $|y_i(k) - y_j(k)| \leq 2.8s_r(k)$. Bold entries satisfy the required coverage of 95%.

Assessment	Sensor 1 [%]	Sensor 2 [%]	Sensor 3 [%]
Complete sequence	96.09	95.81	95.98
Established constant levels	96.12	96.20	96.37
Established periodic flow	96.15	95.83	95.84

sufficiently small, an additional reference-based tolerance is introduced:

$$T(k) = 0.05 |y_{\text{ref}}(k)|, \quad (7-5)$$

where $y_{\text{ref}}(k)$ is the common prescribed reference. This tolerance is used in two additional assessments. First, the observed pairwise difference is compared directly with the tolerance:

$$|y_{p,j}(k) - y_{q,j}(k)| \leq T_j(k). \quad (7-6)$$

The resulting coverage represents the percentage of observed pairwise differences that stay within 5% of the prescribed reference at that time instance.

The second assessment, is checking whether the estimated repeatability limit is smaller than the reference based tolerance. Resulting in:

$$r_{\text{rep},j}(k) \leq T_j(k) \quad (7-7)$$

The corresponding coverage represents the percentage of time instances for which the estimated repeatability limit remains within this reference-based tolerance. These criteria describe the magnitude of the run-to-run variability with respect to the prescribed reference and are not to be mistaken with the tracking error. Acceptable tracking should be determined by looking at the RMSE with respect to the reference as was done in Sections 7-1 and 7-2.

Table 7-6 shows that when the repeatability limit is compared against the reference-based tolerance the total accepted point-wise coverage reduces significantly. For the complete reference sequence, non of the sensors satisfy the required threshold. After excluding the warm-up

Table 7-6: Coverage relative to the application-specific tolerance $0.05|y_{\text{ref}}(k)|$. Pairwise coverage evaluates the observed differences $|y_i(k) - y_j(k)|$, whereas limit adequacy evaluates the pointwise repeatability limit $2.8s_r(k)$. Bold entries satisfy the required coverage of 95%.

Assessment	Criterion	Sensor 1 [%]	Sensor 2 [%]	Sensor 3 [%]
Complete sequence	Observed pairs within 5%	84.75	92.30	94.46
	Repeatability limit within 5%	32.19	65.64	74.42
Established constants	Observed pairs within 5%	87.01	96.11	95.00
	Repeatability limit within 5%	36.83	78.29	79.27
Established periodic flow	Observed pairs within 5%	85.29	93.37	96.56
	Repeatability limit within 5%	28.50	64.63	79.90

and constant reference transient effects, only for sensors 2 and 3 does the coverage surpass 95 % when the pairwise difference is smaller than the 5 % tolerance. When the warm up and periodic transient effects are excluded only sensor 3 surpasses the required threshold for when the pairwise difference is smaller than 5 %. When the repeatability limit is assessed against the 5 % tolerance, in none of the cases is the repeatability threshold satisfied. In order to determine whether or not the repeatability criteria are not met due to consistent variability or because some part of the reference has more outliers, the reference sequence is split into four equal segments. Table 7-7 shows the repeatability coverage based on the reference tolerance. Segments 1 and 4 consistently have the lowest coverage, with no sensors exceeding the 95 % threshold for either criteria. The low coverage for segment 1 can be partially explained by warm-up effects, which have not been excluded in these assessments. Although this is not likely the most dominant effect since segment 4 also consistently has low coverage. Segments 2 and 3 have the highest coverage for sensors 2 and 3. With these segments even surpassing the 95 % threshold for the pairwise differences being smaller than the reference-based tolerance. Despite none of the four segments having consistent acceptable repeatability coverage. These findings do indicate that repeatability coverage has a dependence on operating conditions.

Overall, the experiments satisfy the ASTM-repeatability criterion for all sensors based on the pointwise repeatability limit estimated from the data. This indicates that the observed differences between repeated experiments are generally in line with the repeatability limit estimated from the data. When the pairwise difference is assessed with respect to the reference-based tolerance $0.05|y_{ref}(k)|$ for settled constant flow, sensors 2 and 3 satisfy the repeatability threshold. For the settled periodic flow, only sensor 3 satisfies the repeatability threshold.

Table 7-7: Temporal variation in repeatability over four equal segments of the complete experimental sequence. Pairwise coverage is the percentage of observed pair-time differences satisfying $|y_i(k) - y_j(k)| \leq 0.05|y_{ref}(k)|$. Limit adequacy is the percentage of time samples satisfying $2.8s_r(k) \leq 0.05|y_{ref}(k)|$. No transient samples are excluded. Bold entries satisfy the required coverage of 95%.

Segment	Time interval [s]	Criterion		Sensor 1 [%]	Sensor 2 [%]	Sensor 3 [%]
1	0.8–224.7	Observed within 5%	pairs	79.86	86.93	89.66
		Repeatability within 5%	limit	19.73	55.52	57.53
2	225.5–449.5	Observed within 5%	pairs	92.50	97.73	98.84
		Repeatability within 5%	limit	54.85	86.29	91.97
3	450.2–674.2	Observed within 5%	pairs	82.58	97.96	97.93
		Repeatability within 5%	limit	30.44	82.61	89.97
4	675.0–898.9	Observed within 5%	pairs	84.05	86.59	91.41
		Repeatability within 5%	limit	23.75	38.13	58.19

The estimated repeatability limit does in no case remain within this reference-based tolerance for 95 % of the evaluated time-samples. Therefore, the experiments demonstrate statistical repeatability according to the pointwise repeatability criterion, but no universal repeatability with respect to a reference-based tolerance.

Chapter 8

Conclusion

The objective of this research was to assess to what extent the WindShaper can track prescribed stationary and non-stationary mean-velocity profiles under uncertainty in its input-output behaviour. This was assessed by first developing a reduced representation of the input-output relation and assessing its temporal stability. This was followed by investigating fixed-model feedback control and online model adaptation with inverse control for compensating model inaccuracies during reference tracking.

How can the relationship between WindShaper actuation and measured mean velocity be described in a suitable way for control? The first part of this research was aimed at finding a suitable way of modelling the input-output behaviour of the WindShaper. In order to significantly reduce the input-space, the input function was parametrized with relevant spatial basis functions. This allowed the input-output relation to be represented through an affine static model. The identified model was found to capture the dominant input-output sensitivities across the investigated operating range. However, the remaining modelling error with the validation data indicated that the static input-output model should be considered as an approximate representation of the plant rather than an exact predictor.

This is an important distinction in terms of control, since this makes the identified static model unsuitable for precise open-loop reference tracking. The experimental results showed that control through open-loop inversion resulted in persistent tracking errors. Despite that, it was observed that the static-model captured the input-output behaviour sufficiently well to determine appropriate input update directions in an OFO feedback architecture. The accurate tracking with the OFO architecture also supports the claim of the offline identified input-output map having steady-state characteristics. Therefore, the reduced static model provides useful information for control, but its prediction accuracy remains limited in a stand-alone open-loop architecture.

How does the measured WindShaper response behave over time under fixed stationary and non-stationary input sequences?

Long-term measured WindShaper response under fixed stationary and periodic input sequences was investigated by assessing temporal stability. For stationary flow, the largest fitted change was $\Delta y = -0.0946 \text{ m s}^{-1}$, measured at sensor location y_2 for the $\alpha_0 = 35$ uniform case. For periodic flow, temporal changes were assessed through the fitted mean velocity A , amplitude R and phase ϕ . The largest fitted changes were $\Delta A = 0.0477 \text{ m s}^{-1}$ for sensor y_3 at $T_s = 60 \text{ s}$, $\Delta R = -0.0326 \text{ m s}^{-1}$ for sensor y_2 at $T_s = 30 \text{ s}$ and $\Delta \phi = -0.0200 \text{ rad}$ for sensor y_2 at $T_s = 30 \text{ s}$.

Overall, both the stationary and non-stationary experiments support approximate within-run temporal stability over the investigated duration of 1800 s. For the stationary experiments, 13 out of 18 *CI*s excluded zero, indicating that several rates were statistically distinguishable from zero under the linear regression model. However, short-range residual autocorrelation was present, violating the assumption of residual independence. Therefore the ranges of the observed *CI*s must be interpreted with caution. For the periodic experiments, residual autocorrelation was generally limited for the amplitude and phase fits and for several mean velocity fits, indicating that the resulting *CI*s are more informative. The more persistent residual dependence was observed for larger excitation periods, so their corresponding *CI*s also need to be interpreted with more caution.

The fitted rates and absolute fitted changes should therefore be interpreted as indicators of the trends observed during the individual experimental runs rather than as precise estimates of drift. Despite the associated uncertainty, the magnitudes of the observed accumulated changes remained relatively small over the investigated duration. The WindShaper is therefore considered approximately stable under the investigated conditions. Therefore, within-run drift is not likely to be a dominant factor of the tracking errors that were observed in later reference tracking experiments. This motivated the control architecture compensating for model mismatch.

What capabilities and limitations does feedback control provide for tracking prescribed mean velocity reference profiles under model uncertainty?

The fixed-model feedback architecture showed that the identified input-output models contained useful information for correcting tracking errors, despite not being accurate enough for open-loop inversion. The OFO architecture using the static map resulted in accurate tracking of stationary references, which showed that the static map represented the input-output sensitivity directions well enough to update the control input. Therefore, feedback can compensate for steady-state model mismatch without the static model perfectly capturing the input-output behaviour.

Non-stationary reference tracking, however, showed the limitations for the OFO architecture. Incorporating a FIR model into an MPC architecture improved the tracking of non-stationary reference profiles. This was merely a proof of concept but showed that implementing a dynamic model can be useful for control of non-stationary profiles. This does not inherently imply that a dynamic model is necessary for non-stationary references but does show that the limitations of the static model become more relevant when temporal aspects are introduced. As a result, since both feedback architectures rely on a fixed model, their tracking performance relies on how well the model captures the input-output relation of the WindShaper. This motivated the online model adaptation approach.

To what extent can online adaptation of the input-output model improve tracking accuracy under model uncertainty? Online model adaptation significantly improved tracking for all the tested adaptive least-squares algorithms in both configurations. The RLS achieved the largest improvements in both configurations, reducing overall RMSE by 33.2% for the *NC* and 57.5% for *EMM* configuration. The absolute RMSE hardly differed between these two cases, with the RLS achieving 0.221 m s^{-1} in the *NC* and 0.217 m s^{-1} in the *EMM* configuration. Since the open-loop controller increased from 0.331 to 0.510 m s^{-1} under model mismatch, this shows that RLS largely compensated for the additional model mismatch.

For the facility to be relevant for future wind energy control experiments, run-to-run repeatability was assessed using ten repetitions of the closed-loop RLS algorithm under enforced model mismatch. Based on the pointwise statistical repeatability limit, all three sensors satisfy the required 95% pairwise coverage for the complete sequence, demonstrating statistical repeatability. Repeatability in terms of the stricter reference-based tolerance of $0.05|y_{\text{ref}}(k)|$ resulted in none of the sensors achieving 95% pairwise coverage over the whole sequence. However, the pairwise differences satisfied the criterion when transient effects were left out for certain sensors. The point-wise repeatability limit did not stay within the reference-based tolerance for 95% of the evaluated time samples in any assessment.

Therefore, the repeated experiments show statistical repeatability, but no universal repeatability with respect to the 5% reference-based tolerance. Run-to-run variability is found to depend on sensor location and operating condition, while their effects were not isolated.

To what extent can the WindShaper facility in the WindAI Lab track prescribed heterogeneous and non-stationary mean velocity profiles under uncertainty in its input-output behaviour? This research has shown that the WindShaper is able to generate and track heterogeneous and non-stationary mean velocity reference profiles under the investigated operating conditions. The achievable tracking accuracy is directly dependent on how model uncertainty is treated. The identified static model was found to capture the dominant input-output sensitivities well but is not accurate enough for precise open-loop tracking through model inversion. The measured output remained temporally stable under fixed inputs over the investigated duration of 1800 s. It was found that fixed-model feedback can compensate for tracking errors when the model contains sufficient input-output sensitivity or dynamic information. The online model adaptation was able to reduce the dependency on the initial model by updating the estimated input-output model from the measured data.

The RLS algorithm provided the strongest experimental improvement compared to the other adaptive least-squares approaches. For the *NC*, it reduced overall RMSE from 0.331 to 0.221 m s^{-1} and for the *EMM* configuration it reduced overall RMSE from 0.510 to 0.217 m s^{-1} , resulting in approximately equal absolute tracking in both configurations. These results show that the WindShaper is able to improve tracking significantly over the investigated heterogeneous and non-stationary reference profiles despite uncertainty in the initial input-output model. This is given that the provided measured output is available and the online input-output data provide sufficient excitation for adaptation. This capability is limited to the operating conditions, measurement locations and temporal range investigated in this research.

Future Work Several aspects of this research need to be addressed in future work. First of all, the sensor irregularities that were observed in Chapter 2. The observed inconsistencies

may originate from sensor calibration differences, local flow irregularities or a combination of both. Their individual contributions could not be isolated from the experiments. Although these inconsistencies could be compensated for through feedback and online model adaptation, future experiments require the measurement system to provide consistent and reliable velocity measurements. This can be investigated by comparing the sensor measurements at different locations and comparing them with an independent sensor.

With the fixed input sequences, the experiments demonstrated within-run temporal stability over the investigated duration, while the closed-loop RLS approach achieved statistically run-to-run repeatability. However, the operating range of these findings has not been systematically determined. This is particularly relevant for periodic flow, for which the output amplitude and phase response were found to depend on excitation frequency. For future work, the experiments that were done in the temporal stability assessment need to be expanded for a wider range of mean operating conditions, amplitudes and excitation frequencies to obtain a clearer picture of the dynamic bandwidth of the WindShaper facility.

For future wind energy control experiments, the flow in the region of interest should not be disturbed by external objects such as a measurement setup. A possible next step is to separate model learning from flow generation. Online model adaptation can first be used in real-time to learn the model parameters under specific operating conditions. These model estimates can then be used in an open-loop feedforward architecture, generating prescribed flow profiles without requiring a measurement setup for these experiments. Future work would therefore require to assess how accurate and how long, these adapted model estimates retain their tracking performance across repeated runs and changing operating conditions.

The scope of this research was limited to producing uniform, heterogeneous and non-stationary flow profiles controlled in terms of mean-velocity. However, to be particularly relevant for future wind energy control experiments the flow is desired to be controllable through turbulence intensity or spectral content. This could distinguish the WindShaper facility with respect to other wind energy control experimental facilities, since other facilities offer limited control over spatial and temporal flow characteristics.

Appendices

Appendix A

Alternative Flow Measurement Techniques

This appendix summarizes alternative flow-measurement techniques considered during the selection of the measurement system. The comparison focuses on characteristics relevant to the WindShaper facility and experiments, including temporal resolution, spatial resolution, suitability for real-time feedback, calibration requirements and practical implementation.

Hot-Wire Anemometer

A commonly used point measurement technique for measuring and characterizing flow is the Hot-Wire Anemometer (HWA) [22, 14]. A HWA generally consists of two probes with a thin metallic wire attached between them, usually in the order of microns, made from tungsten, platinum or platinum-iridium [23, 24]. This wire is heated by applying an electrical current. As wind flow passes by the wire, the temperature of the wire fluctuates accordingly. It is important to note that a hot-wire does not measure a velocity, but heat loss from a wire, which is done by measuring the voltage fluctuations. In order to obtain velocity (fluctuations), it is necessary to calibrate the measurements. HWA's such as in Figure A-1 contain a single wire, but triple wire setups also exist to obtain three dimensional measurements [22].

This method is able to determine variables such as mean velocity components, velocity fluctuations, mean temperature and temperature fluctuations [23, 24]. Furthermore, this method is known to have a high spatial and temporal resolution. This is very useful property as it is able to capture velocity fluctuations at the smallest scale [22].

When it comes to point-wise velocity measurements, hot-wire anemometry (HWA) is known to provide more accurate results than pressure-based probes for both stationary flows and flows with temporal velocity fluctuations, due to its small sensing volume and high temporal resolution [23]. Hot wire anemometer calibration requires a lot of attention. Owen and Owen [14] mention that it is necessary to use the exact same equipment and data acquisition techniques that were used during calibration as for experiments. Slight alterations in the

different steps of the measurement chain: anemometers, cables, signal conditioners, filter and analog-to-digital converters can immediately affect the voltage-velocity relation. Therefore additional frequent calibrations are required during tests.

A downside to this method is that in high turbulence conditions, flow velocity can be underestimated due to thermal transient effects along the wire [22]. Additionally, the thin wires are very fragile and can fail under high wind loading.

Particle Image Velocimetry

According to Raffel et al. [26], Particle Image Velocimetry (PIV) is a non-intrusive measuring method that captures flow by photographic recording of microparticles in a fluid flow. In their book 'Particle Image Velocimetry: a practical guide', they describe that the setup of a PIV system can generally be divided into several subsystems: seeding; the tracer particles that are to be tracked are released into the flow at the desired location, illumination; for the tracking of the particles they must be illuminated in a specified plane or volume they must be illuminated twice within a specified time interval, Recording; the light scattered by the particles must be recorded in either two separate frames or a sequence of frames, calibration; this is required in order to find the correct relationship between particle displacement in the image plane and the flow, evaluation; the displacement of the tracked particles must be determined by evaluating the recordings, post-processing; removing inaccurate measurements and isolating the desired flow characteristics requires extensive post-processing.

For wind tunnel and aerodynamic applications the usage of PIV is attractive because it can capture instantaneous flow fields over a specific test section, obtaining spatial characteristics such as mean flow, while not intruding the flow. When used at a very high temporal frequency, it can provide information about unsteady flow fields, something that is more difficult with other measurement techniques [26, 27].

PIV is very effective at measuring full flow fields and investigating causal effects. Van der Hoek et al. [28] used tomographic PIV for measuring the wake behind the upstream turbine to compare wake behaviour for different control strategies. Wang et al. [29] used PIV to characterize turbulent flow structures and the unsteady tip and root vortices evolution in wakes downstream of turbines. A major downside to this measurement method though, are the high costs that come with it [30].

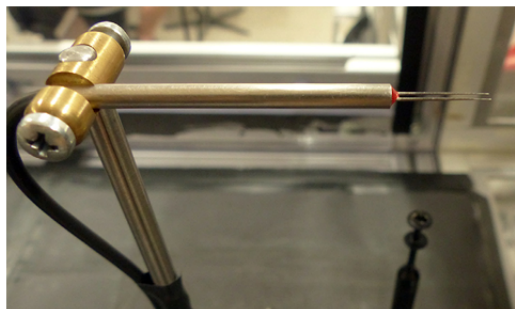


Figure A-1: Example of a single-wire hot-wire anemometer used for pointwise velocity measurements [25].

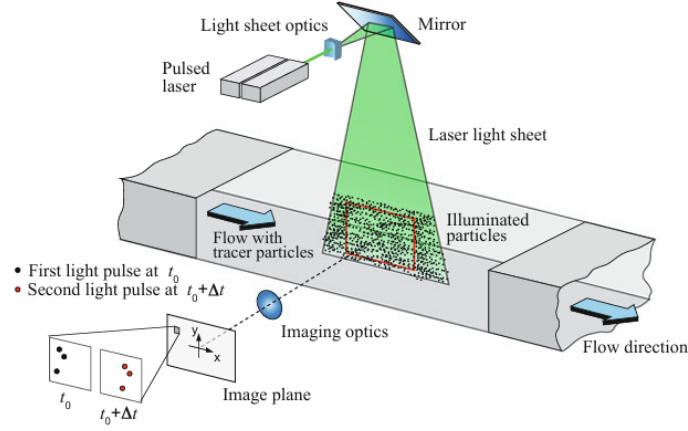


Figure A-2: Experimental setup of the standard (planar) 2C-2D PIV method [26].

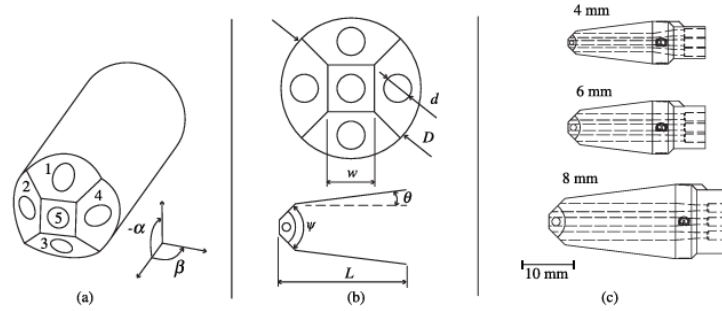


Figure A-3: Design of the Oxford probe with different probe diameters. [31]

Oxford Probe

Common tip shapes of five-hole probes are spherical, truncated canonical and pyramidal tips. Tips with sharper edges are generally less sensitive to Reynolds numbers because their separation points are better defined. Additionally, they are easier to manufacture than spherical tips. Therefore, five-hole probes with sharper edges such as the truncated canonical and pyramidal tips are most common. The tip angle (φ) typically ranges between 60° and 90° . Hall and Povey [31] found that the least Reynolds sensitivity was found for a pyramidal tip with a tip angle of 90° . This led them to a new probe design, the Oxford probe.

The characteristics of the Oxford probe are: a pyramidal tip head with a tip angle of $\varphi = 90^\circ$, which was chosen because of the reduced Reynolds sensitivity and appropriate for additive manufacturing processes. Furthermore, the tip-to-hole diameter ratio (D_{probe}/d_{hole})=5, where D_{probe} is the probe diameter and d_{hole} is the diameter of the holes, and the ratio of the tip diameter to length of the side of the frontal square (D_{probe}/w)=3.75. Additionally, the tip head is joined by an extended cone with an angle $\phi=14.2^\circ$ which has a length in accordance with $L > 2D_{probe}$. Figure A-3 gives a visualisation of the design of the Oxford probe and the different parameters that define it.

According to Hall [31], no physical calibration is needed if the flow is in the incompressible

Table A-1: Minimum inflow velocity associated with the recommended Reynolds-number range for different Oxford-probe diameters [31].

D [mm]	4	6	8	12	24
u_{min} [m/s]	53	35	26	18	9

Table A-2: Comparison of the flow-measurement techniques considered for the WindShaper experiments in terms of temporal and spatial resolution, real-time capability, practical implementation and calibration requirements.

Requirement	HWA	Oxford Probe	PIV	LDA	TriSonica
Temporal resolution	High (kHz)	Moderate (100 Hz)	Low (post-processed)	High (kHz)	Moderate (100 Hz)
Spatial resolution	Point [23]	Point [23]	Volumetric [26]	Point [32]	Point
Real-time feedback	Yes	Yes	No [26]	No	Yes
Cost/Ease of use	Moderate/Moderate	Low/ Moderate	High [30]/Complex	High [30] / Complex	High cost/Easy use
Calibration needs	Frequent [14]	Minimal [31]	Complex	Complex	Minimal
Suitability for Wind-Shaper	Limited by fragility	Ideal	Impractical	Overkill	Ideal

range and the Re range based on the tip size $Re_{D,critical} > 15 \times 10^3$. If this is the case the probes are able to measure with good accuracy. In this case, the calibration map provided by [31] can be used. Under ambient air conditions, for each probe diameter the minimum inflow velocity is given in table A-1. Furthermore, the probe is easily implementable in the windshaper and multiple can be inserted to obtain a more complete picture of the flow. Besides that, it only measures pressure and orientation, making it a non complex method. That being said, it has external pressure transducers potentially meaning a reduced temporal bandwidth.

Table A-2 gives an overview of all the pro's and cons of the considered measurement techniques.

Appendix B

Facility Characterization

The facility characterization performed in Chapter 2 was performed using continuous planar measurements. Preliminary measurements showed that step-wise measurement resulted in significant sensor vibrations which introduced additional measurement disturbances. Figure B-1 compares the resulting velocity field obtained from step-wise and continuous measurements and supports the use of continuous measurements for the facility characterization.

This appendix provides additional planar measurements supporting the control-oriented facility characterisation presented in Chapter 2.

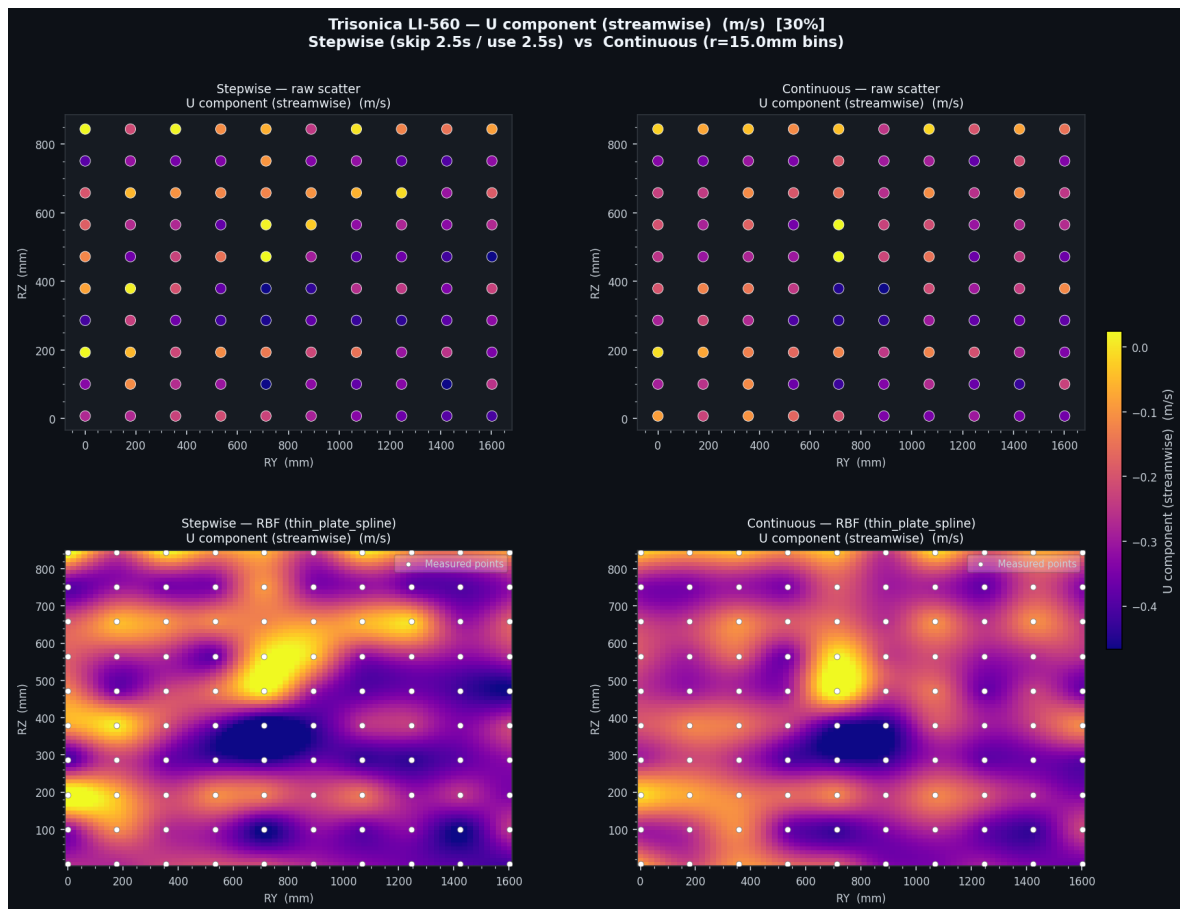


Figure B-1: Comparison of the mean velocity field obtained using stepwise and continuous TriSonica motion. The continuous measurement provides a smoother spatial representation of the flow field. The velocity deficit visible in the measurement plane is induced by the wind turbine.

B-1 Spatial Non-Uniformities

Figures B-2 and B-3 show the spatial variation with respect to the mean velocity. The spatial deviation of the flow is defined with the Root-Mean-Square (RMS) deviation of the local mean velocities, as

$$\sigma_{\bar{U},sp} = \sqrt{\frac{1}{N_b} \sum_{j=1}^{N_b} (\bar{U}_j - \bar{U}_{plane})^2} \quad (B-1)$$

where $\sigma_{\bar{U},sp}$ is the spatial RMS deviation of the binned mean velocity. N_b the amount of 20×20 mm bins $\bar{U}_j$ the mean velocity per bin and $\bar{U}_{plane}$ the mean velocity per plane.

Spatial mean-velocity deviation

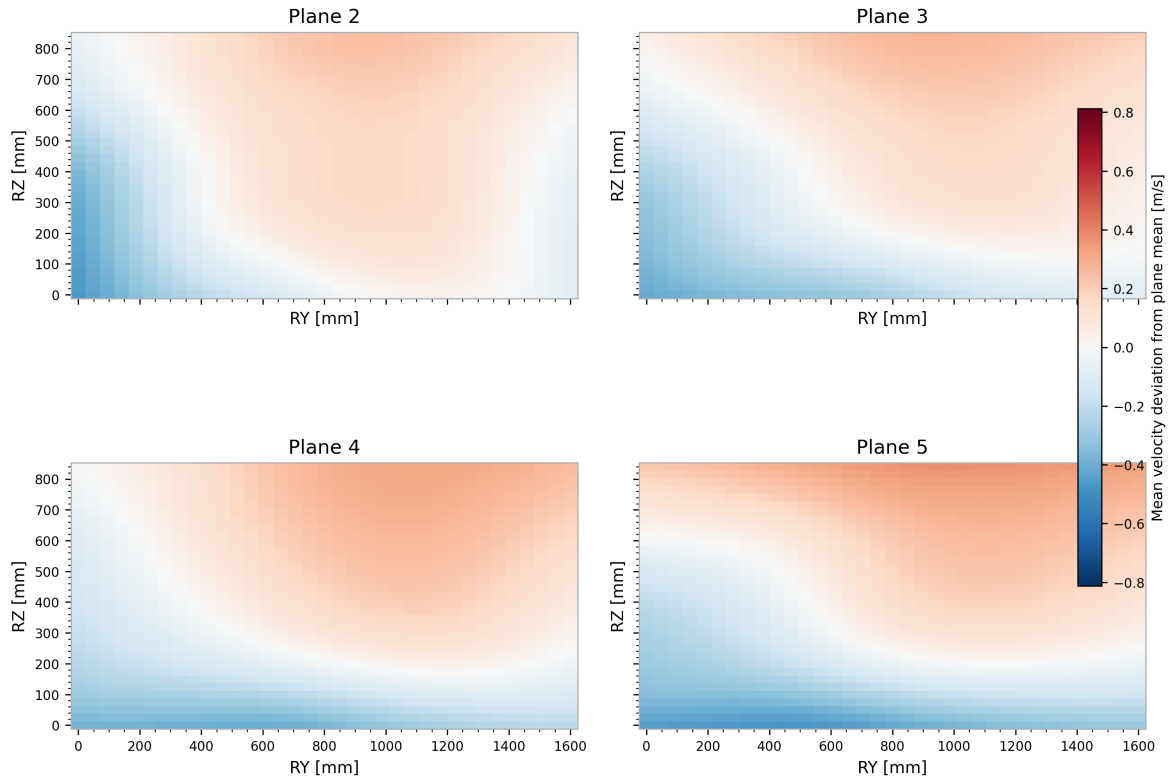


Figure B-2: Spatial mean-velocity deviation for a uniform 40% WindShaper command for the bottom half of the downstream planes 2–5. Colours indicate the deviation of the local mean velocity from the spatial mean velocity of the corresponding plane. Spatial non-uniformities remain present throughout the wind box and do not decrease for further downstream distances.

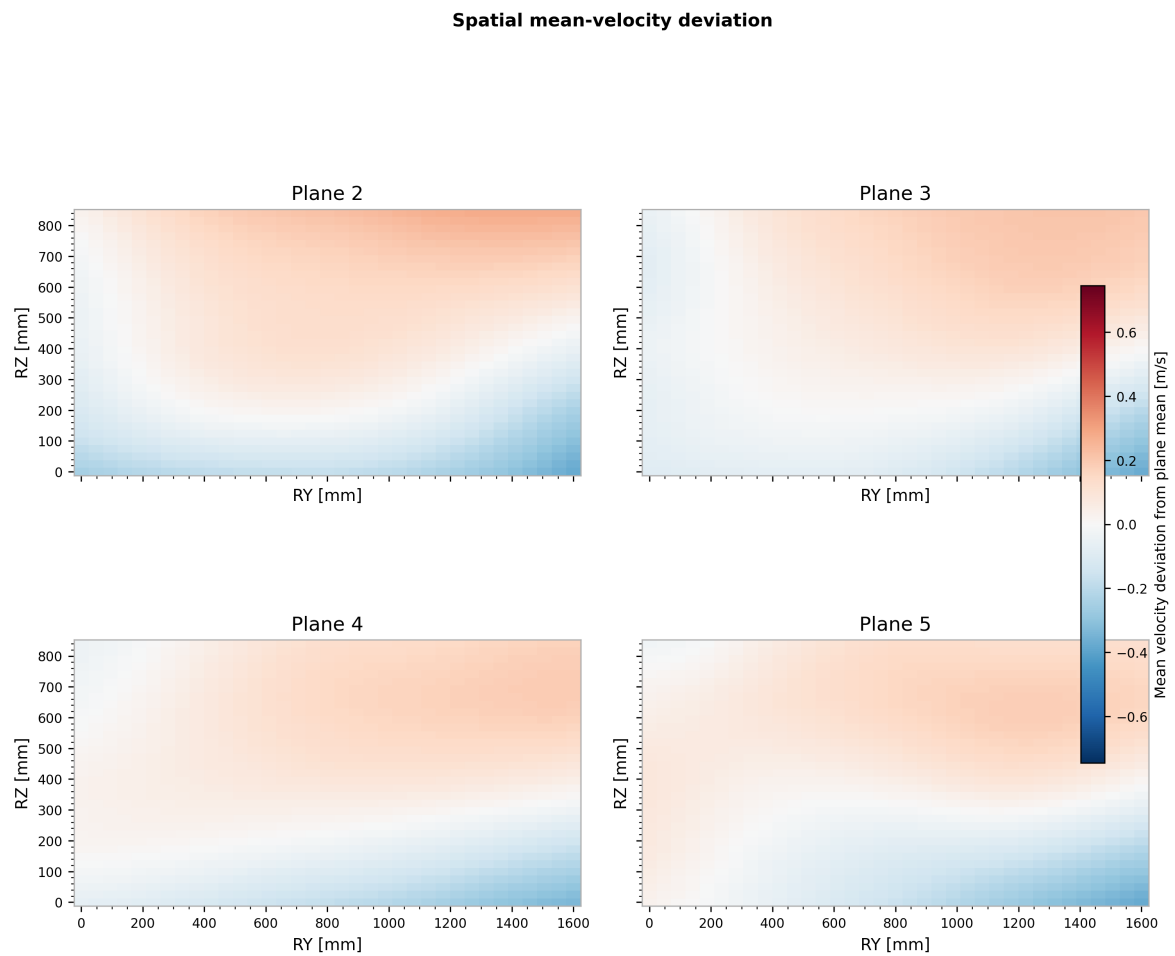


Figure B-3: Spatial mean-velocity deviation for a uniform 40% WindShaper command for the top half of the downstream planes 2–5. Colours indicate the deviation of the local mean velocity from the spatial mean velocity of the corresponding plane. Less non-uniformities seem present than for the bottom half but still remain present at all downstream distances.

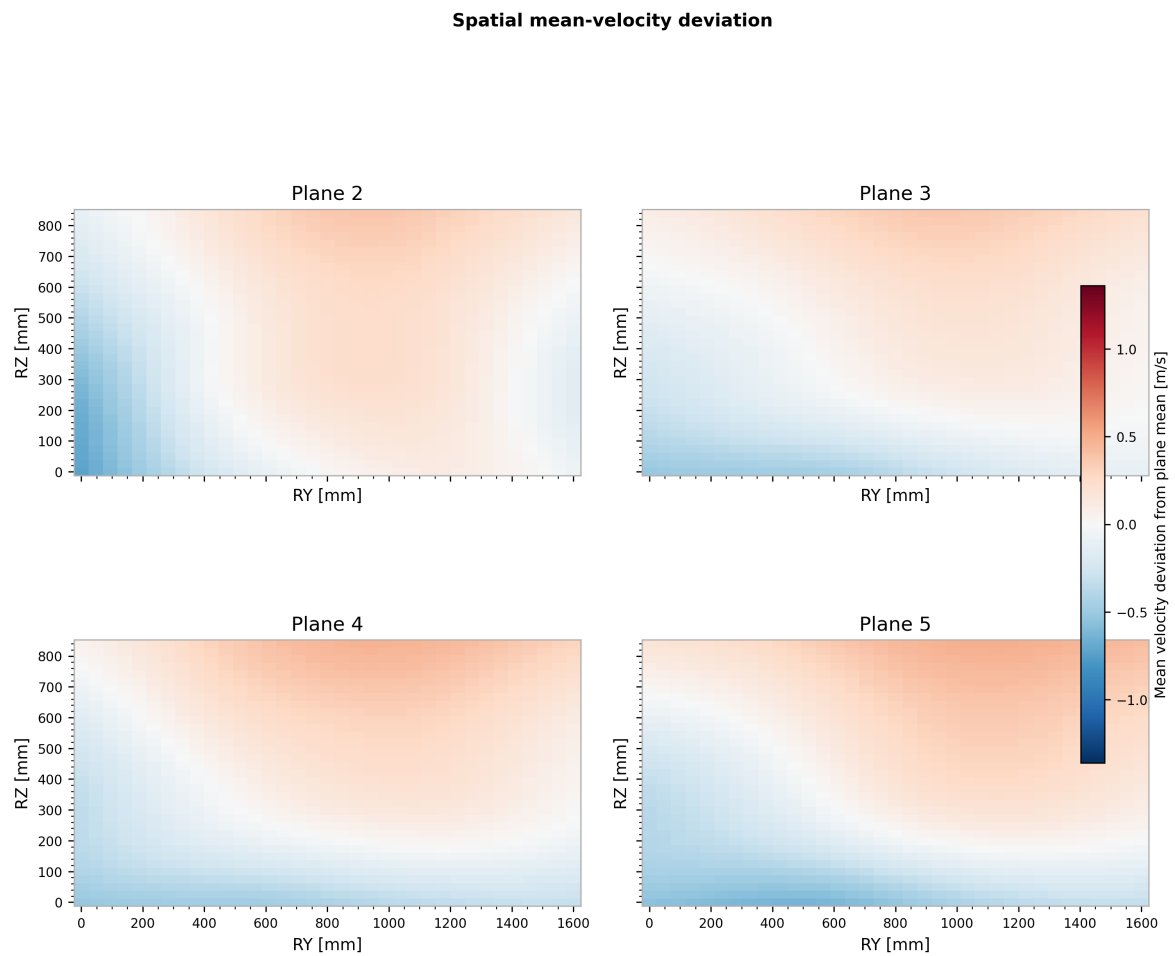


Figure B-4: Spatial mean-velocity deviation for a uniform 60% WindShaper command for the bottom half of the downstream planes 2–5. Colours indicate the deviation of the local mean velocity from the spatial mean velocity of the corresponding plane. Spatial non-uniformities remain present throughout the wind box and do not decrease for further downstream distances.

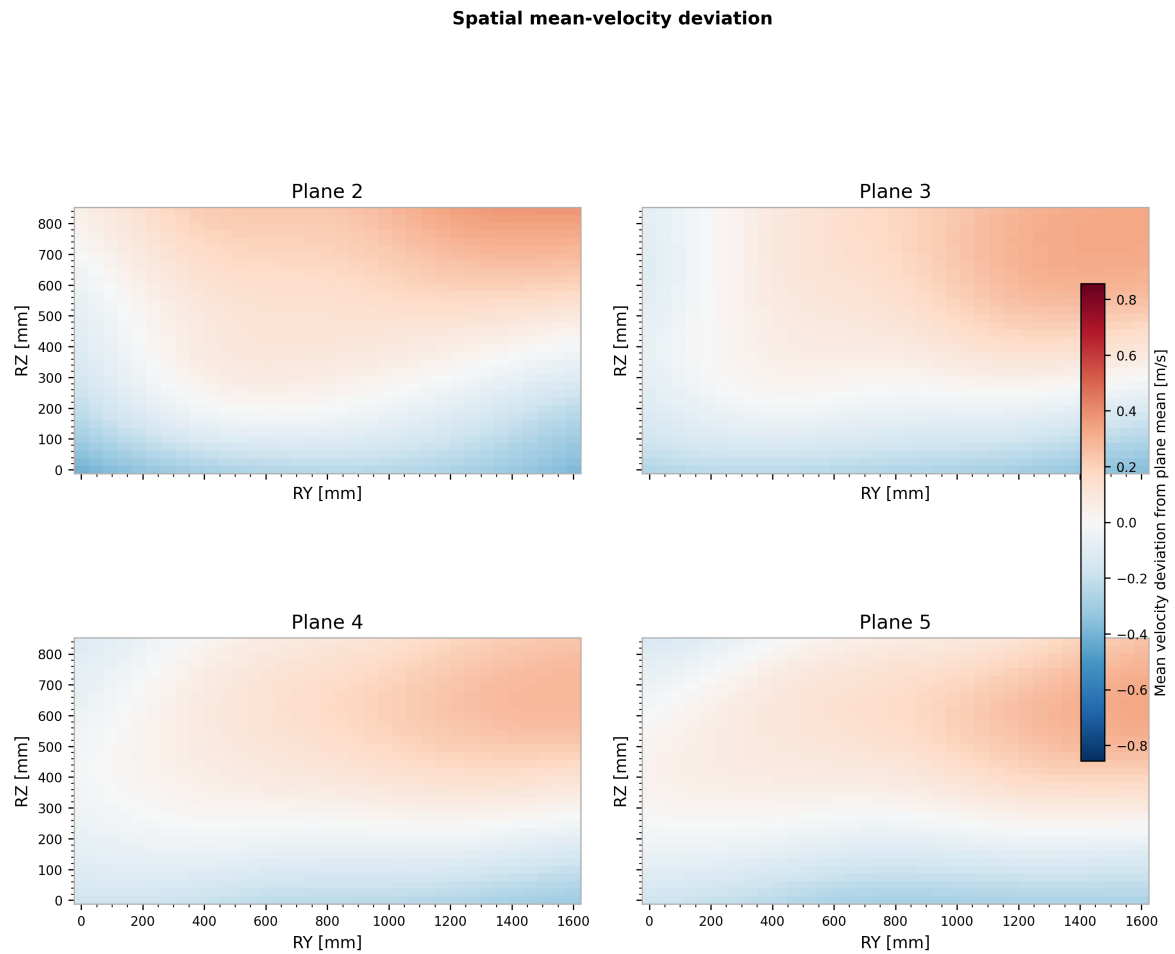


Figure B-5: Spatial mean-velocity deviation for a uniform 60% WindShaper command for the top half of the downstream planes 2–5. Colours indicate the deviation of the local mean velocity from the spatial mean velocity of the corresponding plane. Spatial non-uniformities remain present throughout the wind box and do not decrease for further downstream distances.

Appendix C

Static-Model Identification Inputs

The static model was identified and validated using 26 prescribed spatial input functions. These input functions were designed to sufficiently excite the uniform, horizontal gradient and vertical gradient basis directions individually and in combination. The operating points were distributed between the training and validation datasets across the experimental sequence to reduce sensitivity of the split to slow temporal changes in experimental conditions. The complete set of prescribed input functions is listed below.

$$u_1(x, y) = 45 + 15\zeta - 15\eta, \quad u_2(x, y) = 40, \quad (\text{C-1})$$

$$u_3(x, y) = 35 - 15\zeta + 15\eta, \quad u_4(x, y) = 45 - 15\zeta - 15\eta, \quad (\text{C-2})$$

$$u_5(x, y) = 40 + 20\eta, \quad u_6(x, y) = 35 + 15\zeta + 15\eta, \quad (\text{C-3})$$

$$u_7(x, y) = 45, \quad u_8(x, y) = 40 - 20\zeta, \quad (\text{C-4})$$

$$u_9(x, y) = 35 - 15\zeta - 15\eta, \quad u_{10}(x, y) = 45 + 15\zeta + 15\eta, \quad (\text{C-5})$$

$$u_{11}(x, y) = 35, \quad u_{12}(x, y) = 40 + 20\zeta, \quad (\text{C-6})$$

$$u_{13}(x, y) = 35 + 15\zeta - 15\eta, \quad u_{14}(x, y) = 45 - 15\zeta + 15\eta, \quad (\text{C-7})$$

$$u_{15}(x, y) = 40 - 20\eta, \quad u_{16}(x, y) = 40, \quad (\text{C-8})$$

$$u_{17}(x, y) = 32 + 12\zeta + 8\eta, \quad u_{18}(x, y) = 44, \quad (\text{C-9})$$

$$u_{19}(x, y) = 38 - 12\zeta + 8\eta, \quad u_{20}(x, y) = 36 + 14\zeta - 8\eta, \quad (\text{C-10})$$

$$u_{21}(x, y) = 42 - 18\zeta + 6\eta, \quad u_{22}(x, y) = 32 + 8\zeta + 12\eta, \quad (\text{C-11})$$

$$u_{23}(x, y) = 38 + 15\zeta - 15\eta, \quad u_{24}(x, y) = 36 - 8\zeta - 14\eta, \quad (\text{C-12})$$

$$u_{25}(x, y) = 42 + 6\zeta + 18\eta, \quad u_{26}(x, y) = 44. \quad (\text{C-13})$$

Appendix D

Within Run Temporal Stability

This appendix provides supporting information to the within run temporal stability assessment that was done in Chapter 3. It includes the measured temperature evolution across all experiments, the complete stationary drift results and the residual autocorrelation diagnostics to assess the temporal independence of the fitted residuals behind the ordinary least-squares *CI*.

D-1 Temperature Evolution

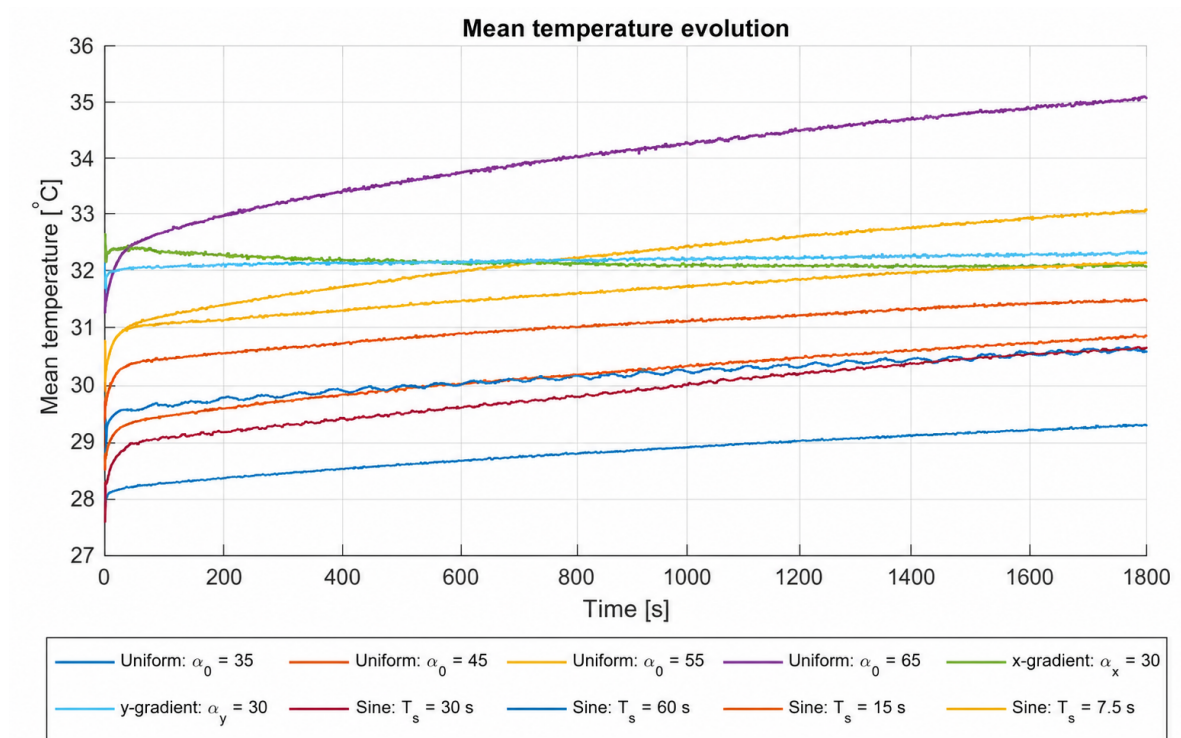


Figure D-1: Measured ambient-temperature evolution during the ten stationary and periodic experiments used for the within-run temporal-stability assessment. Temperature was monitored as supporting information and was not independently controlled.

D-2 Stationary Input Sequences

Table D-1: Estimated initial velocities and velocity drift rates for the stationary input experiments after discarding the first 60 s. The confidence intervals describe the uncertainty of the fitted rate c_t .

Input condition	Sensor	c_0 [m s ⁻¹]	c_t [m s ⁻¹ min ⁻¹]	95% confidence interval [m s ⁻¹ min ⁻¹]
Uniform: $\alpha_0 = 35$	y_1	4.9709	3.5783×10^{-4}	$[9.6120 \times 10^{-5}, 6.1955 \times 10^{-4}]$
	y_2	5.4796	-3.2757×10^{-3}	$[-3.5128 \times 10^{-3}, -3.0387 \times 10^{-3}]$
	y_3	5.5505	6.2460×10^{-4}	$[3.3115 \times 10^{-4}, 9.1804 \times 10^{-4}]$
Uniform: $\alpha_0 = 45$	y_1	6.2124	-1.6476×10^{-3}	$[-1.9690 \times 10^{-3}, -1.3263 \times 10^{-3}]$
	y_2	6.6218	-2.7375×10^{-3}	$[-3.0645 \times 10^{-3}, -2.4106 \times 10^{-3}]$
	y_3	6.9304	1.4449×10^{-3}	$[1.0964 \times 10^{-3}, 1.7934 \times 10^{-3}]$
Uniform: $\alpha_0 = 55$	y_1	7.3140	3.0738×10^{-3}	$[2.7150 \times 10^{-3}, 3.4326 \times 10^{-3}]$
	y_2	7.6959	-1.1110×10^{-3}	$[-1.6522 \times 10^{-3}, -5.6981 \times 10^{-4}]$
	y_3	8.3087	-1.0305×10^{-3}	$[-1.4180 \times 10^{-3}, -6.4305 \times 10^{-4}]$
Uniform: $\alpha_0 = 65$	y_1	8.7416	-9.9760×10^{-5}	$[-4.7959 \times 10^{-4}, 2.8007 \times 10^{-4}]$
	y_2	9.1482	1.4689×10^{-3}	$[7.2469 \times 10^{-4}, 2.2131 \times 10^{-3}]$
	y_3	9.7574	-2.4886×10^{-4}	$[-6.7974 \times 10^{-4}, 1.8203 \times 10^{-4}]$
x -gradient: $\alpha_x = 30$	y_1	4.5416	-7.4157×10^{-4}	$[-1.2566 \times 10^{-3}, -2.2652 \times 10^{-4}]$
	y_2	6.4729	1.0034×10^{-3}	$[7.7513 \times 10^{-4}, 1.2317 \times 10^{-3}]$
	y_3	2.6334	-4.5925×10^{-4}	$[-1.2324 \times 10^{-3}, 3.1390 \times 10^{-4}]$
y -gradient: $\alpha_y = 30$	y_1	7.1795	7.0116×10^{-4}	$[4.4559 \times 10^{-4}, 9.5673 \times 10^{-4}]$
	y_2	2.6327	-5.6764×10^{-4}	$[-1.6472 \times 10^{-3}, 5.1195 \times 10^{-4}]$
	y_3	2.6887	-3.7575×10^{-4}	$[-1.2233 \times 10^{-3}, 4.7178 \times 10^{-4}]$

The confidence intervals used in Chapter 3 assume temporally uncorrelated regression residuals. To assess this assumption, the autocorrelation function was computed for the residuals remaining after removal of the fitted linear drift. For the stationary experiments, one lag corresponds to one 0.75-s averaged measurement.

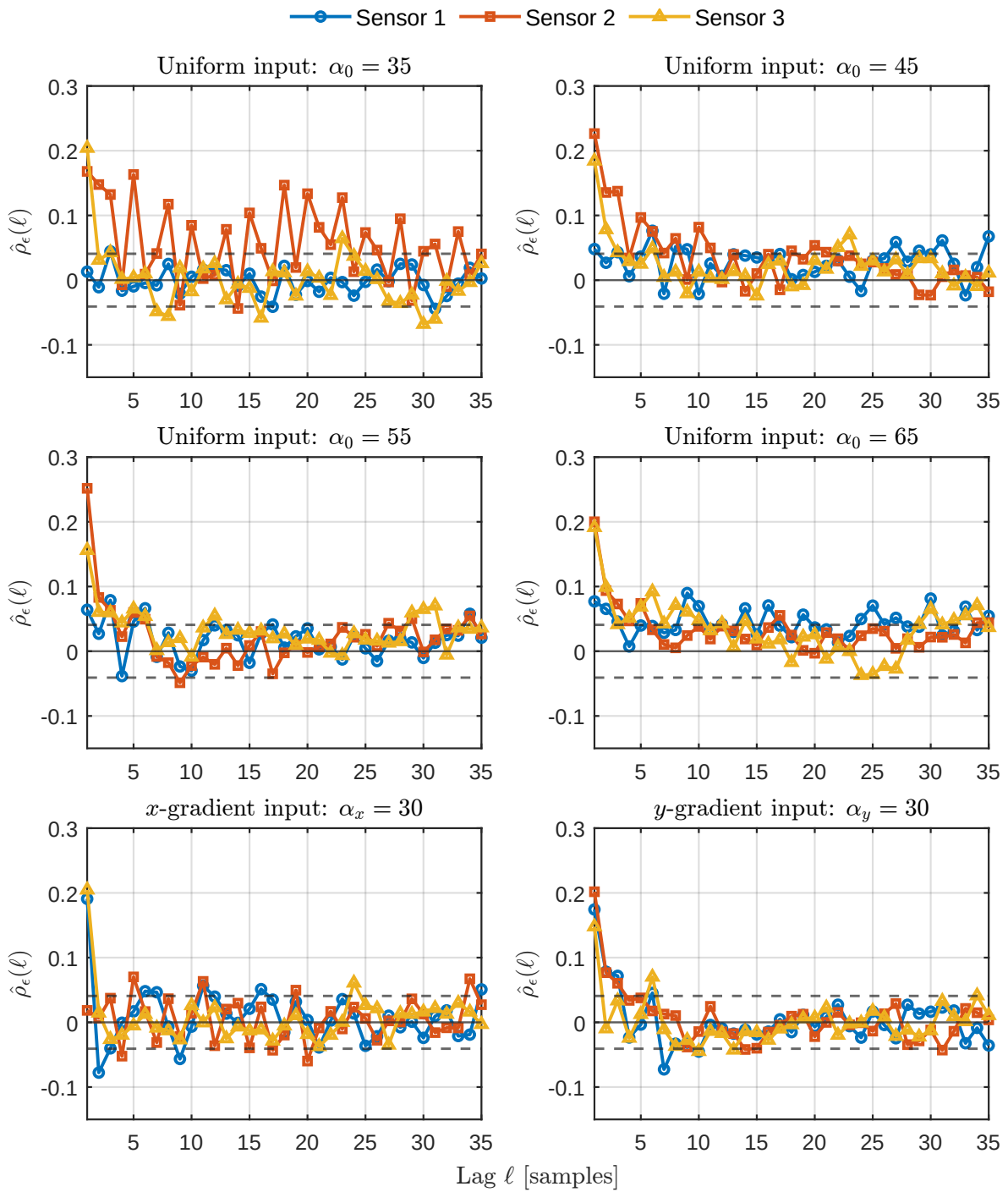


Figure D-2: Residual autocorrelation after removal of the fitted linear drift for the stationary fixed-input experiments. The three curves correspond to the three sensor outputs. One lag represents one 0.75-s averaged measurement, and the dashed lines indicate approximate 95% white-noise bounds.

D-3 Periodic Input Sequences

For the periodic experiments, residual autocorrelation was evaluated separately for the linear temporal trends fitted to the period-wise mean level A , amplitude R , and unwrapped phase ϕ . Since each observation corresponds to one fitted excitation period, a lag of one represents one complete period T_s . Consequently, the physical time represented by a given lag differs between excitation periods.

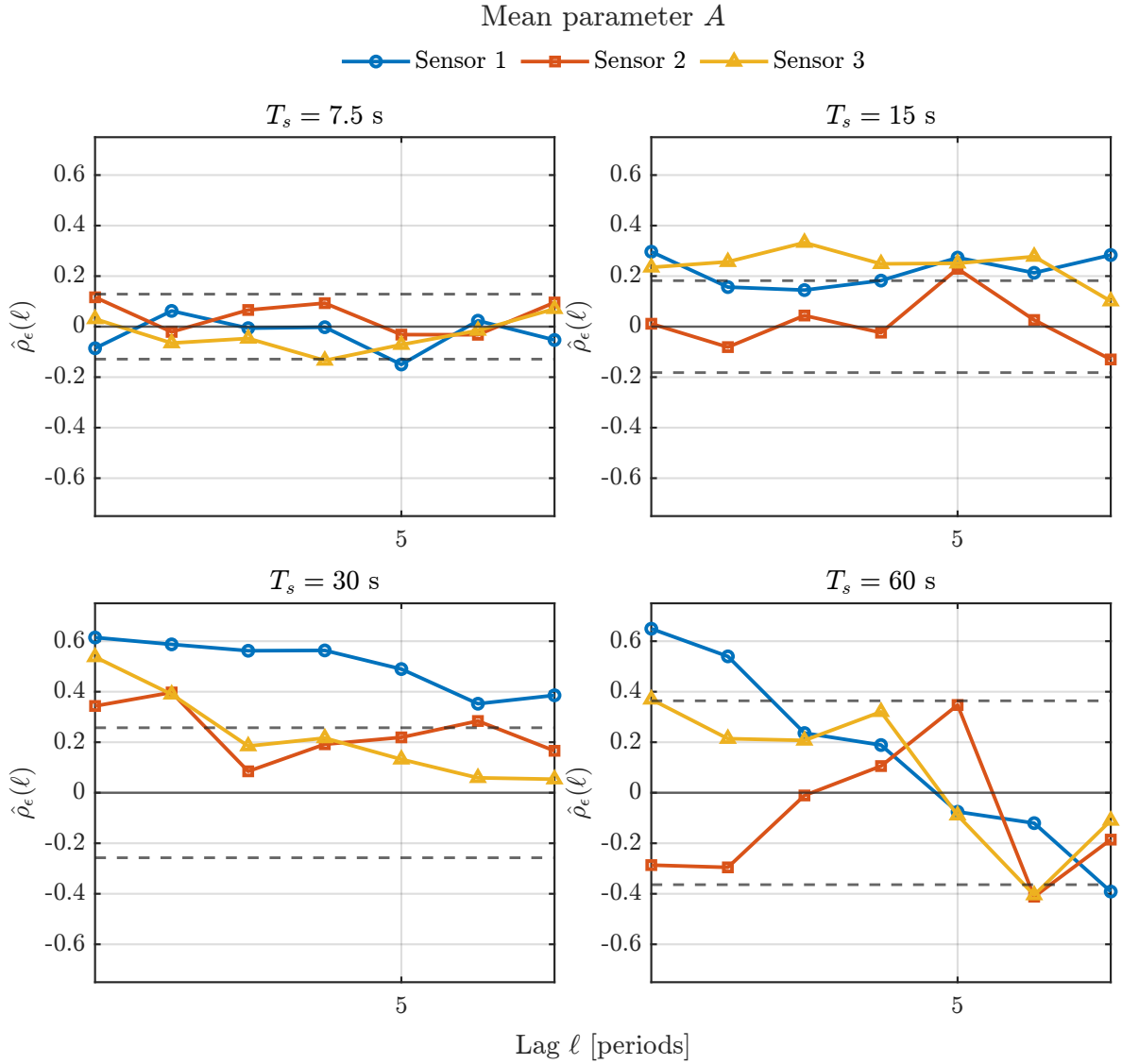


Figure D-3: Residual autocorrelation of the fitted mean-level A drift models for the periodic input experiments. Each panel corresponds to one excitation period T_s , and the three curves denote the sensor outputs. One lag represents one complete excitation period; dashed lines indicate approximate 95% white-noise bounds.

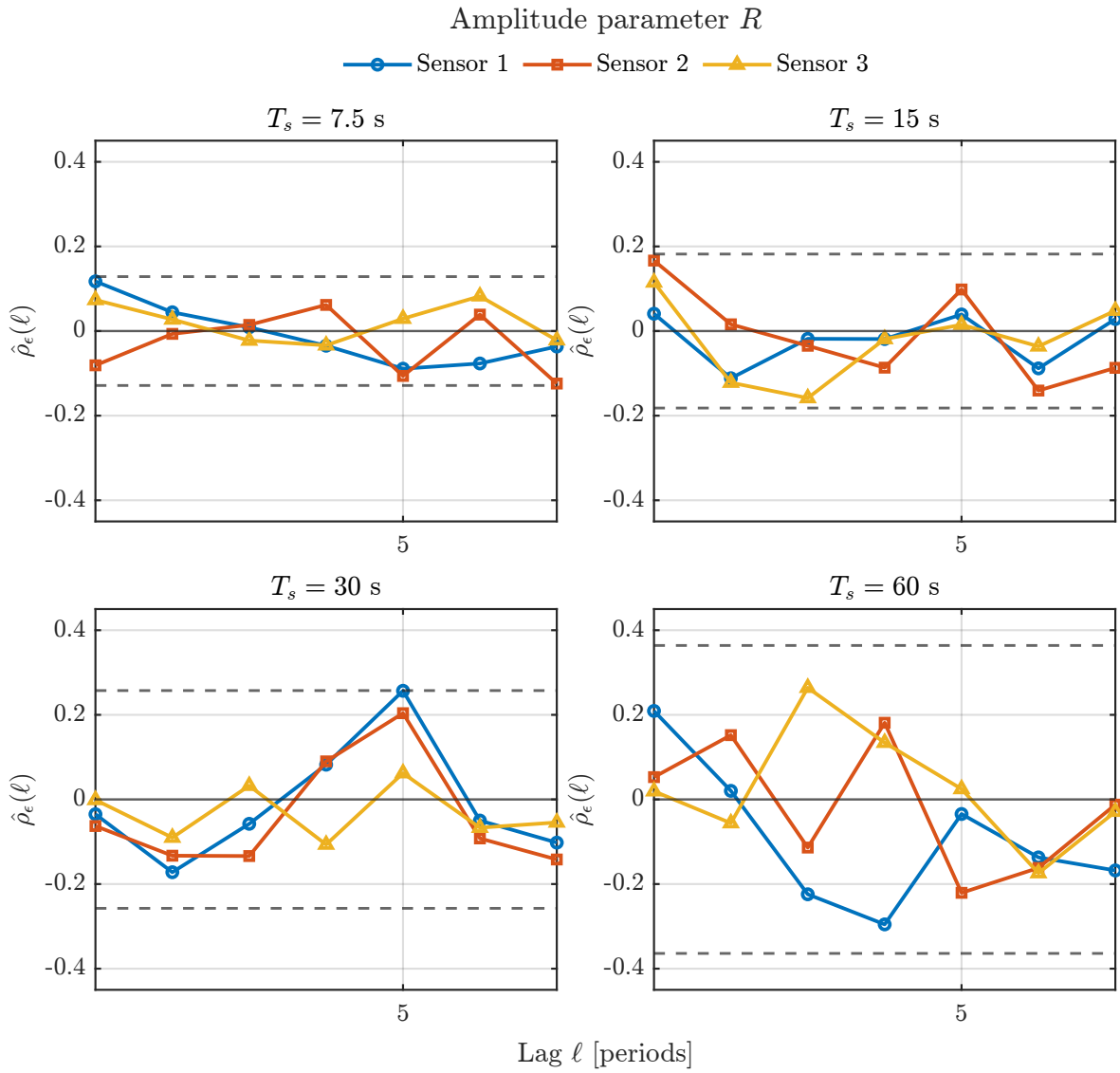


Figure D-4: Residual autocorrelation of the fitted amplitude R drift models for the periodic input experiments. One lag represents one complete excitation period; dashed lines indicate approximate 95% white-noise bounds.

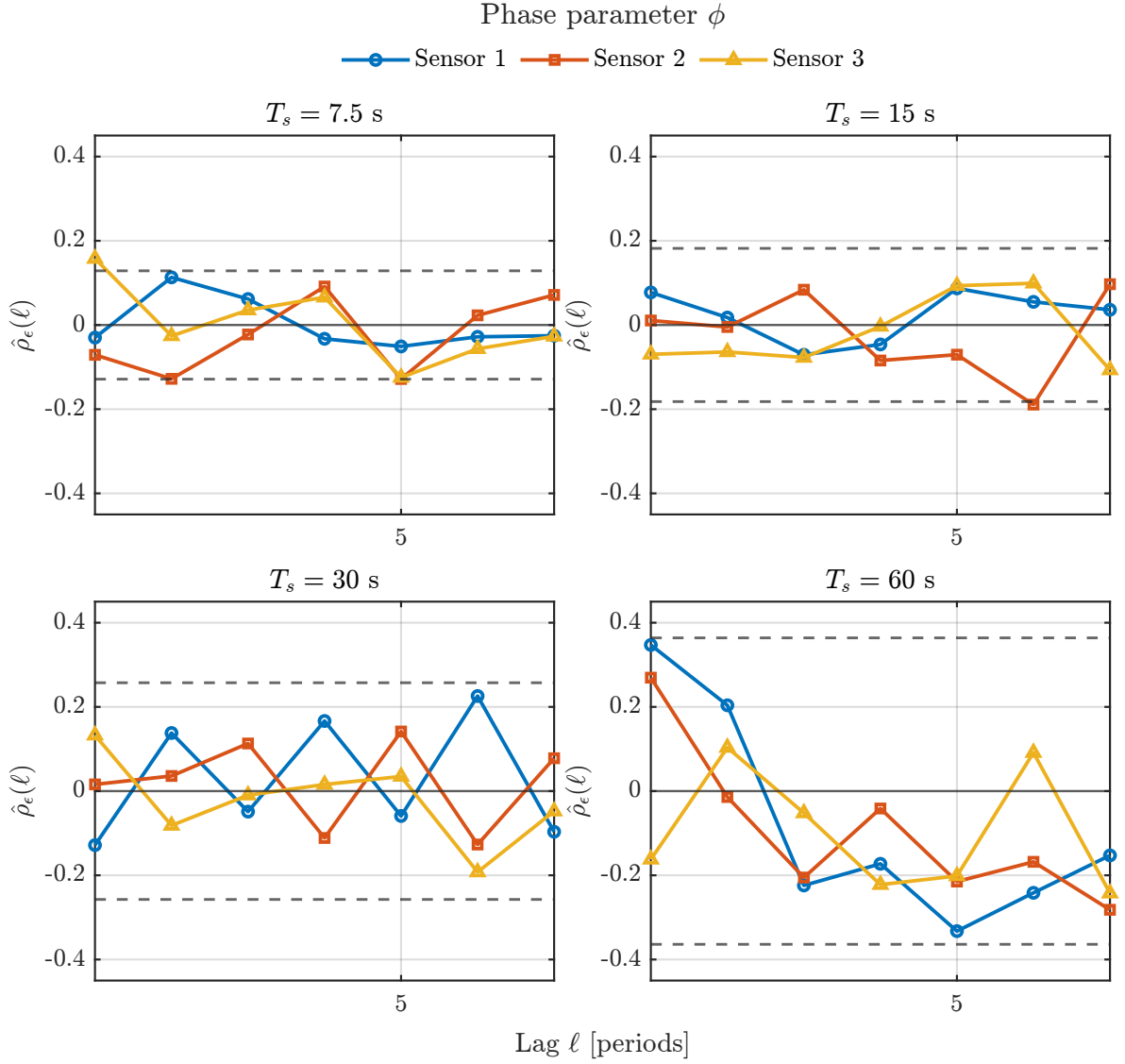


Figure D-5: Residual autocorrelation of the fitted unwrapped-phase ϕ drift models for the periodic input experiments. One lag represents one complete excitation period; dashed lines indicate approximate 95% white-noise bounds.

Least-Squares Simulation Details

E-1 Simulation Settings

This appendix summarizes the numerical settings used for the adaptive least-squares simulations in Chapter 6. The simulations were intended as controlled test cases for comparing estimator behaviour rather than as a high-fidelity model of the WindShaper dynamics.

Table E-1: Numerical settings and implementation choices used for the adaptive least-squares simulations presented in Chapter 6.

Setting	Implementation
Plant model	$\mathbf{y}_k = \mathbf{G}_{\text{true}}\boldsymbol{\alpha}_k + \mathbf{b}_{\text{true}}(k) + \boldsymbol{\epsilon}_k$. No additional input–output delay was simulated.
Initial model	$\hat{\mathbf{G}}_0 = \mathbf{G}_{\text{ID}}$ and $\hat{\mathbf{b}}_0 = \mathbf{b}_{\text{ID}}$, using the offline-identified static model from Section 2-4.
Plant cases	Nominal, bias mismatch, gain-and-bias mismatch, and slow drift, as defined in Table 6-1.
Sampling and duration	$\Delta t = 0.75$ s, $N = 902$ samples, and $t_f = 675.75$ s.
Reference sequence	Twenty-six constant-reference segments of approximately 20 s, followed by three periods of a $T = 20$ s sinusoid and three periods of a $T = 30$ s sinusoid.
Measurement noise	Independent zero-mean Gaussian noise, $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, 0.04^2 \mathbf{I}_3)$. The same noise realisation was used for all algorithms within each plant case.
Regularisation	Least-squares regularisation $\rho_{\Phi} = 10^{-6}$ and inverse-control regularisation $\rho_G = 10^{-4}$.
Open-loop inverse	$\mathbf{G}_{\text{ID}}$ and $\mathbf{b}_{\text{ID}}$ remained fixed throughout the simulation.
Cumulative LS	Adaptation started after $N_{\min} = 60$ samples. The model was subsequently updated at every sample using all available data.
Batch-wise LS	Non-overlapping datasets with $N_D \in \{60, 100, 140, 180, 220\}$. The model was updated after each complete batch.
Moving-window LS	Sliding datasets with $N_D \in \{60, 100, 140, 180, 220\}$. The model was updated at every sample after the window was filled.
Recursive LS	$\lambda_{\text{RLS}} \in \{0.98, 0.95\}$ and $\mathbf{P}_0 = \mathbf{I}_4$. RLS processed measurements from the start, while its estimate was used for inverse control after 60 samples.
Input constraints	No coefficient clipping or spatial PWM-feasibility constraints were imposed in the simulations.
Randomisation	MATLAB random-number seed <code>rng(1)</code> ; one noise realisation was evaluated per plant case.

Bibliography

- [1] J. A. Frederik. *Pitch control for wind turbine load mitigation and enhanced wake mixing: A simulation and experimental validation study*. PhD thesis, Delft University of Technology, 2021. Final published version.
- [2] Alessandro Bianchini, Carlos Simao Ferreira, Ion Paraschivoiu, and Pier Francesco Melani. Are vertical-axis wind turbines a scientific mirage or the future of offshore wind energy? a critical perspective a century after darrieus' patent. *Renewable and Sustainable Energy Reviews*, 234:116915, 2026.
- [3] F. Porte-Angel, M. Bastankhah, and S. Shamsoddin. Wind-turbine and wind-farm flows: A review. *Boundary-Layer Meteorology*, 2019.
- [4] S. C. Pryor, R. J. Barthelmie, and T. J. Shepherd. Wind power production from very large offshore wind farms. *Joule*, 2021.
- [5] R. Nash, R. Nouri, and A. Vassel-Be-Hagh. Wind turbine wake control strategies: A review and concept proposal. *Energy Conversion and Management*, 2021.
- [6] J. Meyers, C. Bottasso, K. Dykes, P. Fleming, P. Gebraad, G. Giebel, T. Göçmen, and J. W. van Wingerden. Wind farm flow control: prospects and challenges. *Wind Energy Science*, 2022.
- [7] D. Ossman, P. Seiler, C. Milliren, and A. Danker. Field testing of multi-variable individual pitch control on a utility-scale wind turbine. *Renewable Energy*, 2021.
- [8] M. A. Z. Inestroza, P. Hulsman, V. Petrović, and M. Kühn. Dynamic induction control for mitigation of wake-induced power losses: a wind tunnel study under different inflow conditions. *Wind Energy Science*, 2025.
- [9] A. A. W. van Vondelen, D. C. van der Hoek, S. T. Navalkar, and J. W. van Wingerden. Experimental validation of synchronized helix wake mixing control. *Renewable Energy*, 2026.

- [10] F. V. Mühle, F. M. Heckmeier, F. Campagnolo, and Breitsamter C. Wind tunnel investigations of an individual pitch control strategy for wind farm power optimization. *Wind Energy Science*, 2024.
- [11] M. Placidi, P. E. Hancock, and P. Hayden. Windturbine wakes: experimental investigation of two-point correlations and the effect of stable thermal stability. *Journal of Fluid Mechanics*, 2023.
- [12] A. Stefan-Zavala, I. Scherl, I. Mandralis, S. L. Brunton, and M. Gharib. Data-driven modelling for on-demand flow prescription in fan-array wind tunnels. *Cambridge University Press*, 2025.
- [13] LI-COR, Inc. Mounting the li-560 anemometer.
- [14] F. K. Owen and A. K. Owen. Measurement and assessment of wind tunnel flow quality. *Progress in Aerospace Sciences*, 44(5):315–348, 2008.
- [15] K. B. Rajasekara Babu, G. Hu, and B. R. Noack. From active grids to fan-array wind generators: A review of turbulence generation, control, and artificial intelligence integration in wind tunnels. *Physics of Fluids*, 2025.
- [16] S. Brunton and B. Noack. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 2015.
- [17] Pennsylvania State University. Stat 501: Regression methods – lesson 1: Simple linear regression. <https://online.stat.psu.edu/stat501/Lesson01>. Accessed: 27 August 2026.
- [18] Pennsylvania State University. Topic 2: Time series & autocorrelation. <https://online.stat.psu.edu/stat501/Topic2TSandAutocorr>. STAT 501: Regression Methods, accessed 26 August 2026.
- [19] G. Prasath and J. Bagterp Jørgensen. Model predictive control based on finite impulse response models. In *American Control Conference*, 2008.
- [20] Wittenmark B. J. and Åström K. J. *Computer Controlled Systems - Theory and Design*. Prentice Hall, 1997.
- [21] ASTM International. Standard practice for use of the terms precision and bias in ASTM test methods, 2020.
- [22] M. William and M McQuilling. Turbulent jet statistics: Five-hole probe vs hot-wire anemometry. In *AIAA SciTech 2024 Forum*, Orlando, FL, USA, January 2024. AIAA.
- [23] S. Ristić, J. Isaković, B. Ilić, and G. Ocokoljić. Review of methods for flow velocity measurement in wind tunnels. *Scientific Technical Review*, 54(3):60–75, 2004.
- [24] D. Barratt, M.A. Nguenpang, and M.D. Atkins. Chapter 4 - fast-response velocity and shear stress measurements. In Tongbeum Kim, Tian Jian Lu, and Seung Jin Song, editors, *Application of Thermo-Fluidic Measurement Techniques*, pages 101–123. Butterworth-Heinemann, 2016.

-
- [25] J. Sobczyk. Experimental study of the flow field disturbance in the vicinity of single sensor hot-wire anemometer. *Experimental Fluid Mechanics*, 2018.
 - [26] M. Raffel, C. E. Willert, F. Scarano, C. J. Kähler, S. T. Wereley, and J. Kompenhans. *Particle Image Velocimetry*. Springer, 2018.
 - [27] C. Tropea, A. L. Yarin, and J. F. Foss, editors. *Springer Handbook of Experimental Fluid Mechanics*. Springer, Berlin, Heidelberg, 2 edition, 2016.
 - [28] D. van der Hoek, C. Ferreira, and J. W. van Wingerden. Experimental comparison of induction control methods for wind farm power maximization on a scaled two-turbine setup. *Journal of Physics*, 2024.
 - [29] Z. Wang, A. Ozbay, W. Tian, and H. Hu. An experimental study on the aerodynamic performances and wake characteristics of an innovative dual-rotor wind turbine. *Energy*, 2018.
 - [30] C. McKenna Neuman, P. Ashmore, and S.J. Bennett. 1.17 laboratory and experimental geomorphology: Examples from fluvial and aeolian systems. In J. F. Shroder, editor, *Treatise on Geomorphology*, pages 325–348. Academic Press, San Diego, 2013.
 - [31] B. F. Hall and T. Povey. The oxford probe: an open access five-hole probe for aerodynamic measurements. *Measurement Science and Technology*, 28(3):035004, 2017.
 - [32] G. Rossi. *Mechanical and Thermal Measurements by Images and Waves. Smart Sensors, Measurement and Instrumentation*, chapter Laser Doppler Anemometry, Vibrometry and Particle Image Velocimetry. Springer, 2025.