# Online Black-box Shape Optimization for a Seamless Active Morphing Wing

Master thesis

O.L. Ruland

TU Delft

*cover image: the SmartX-Alpha active morphing wing demonstrator in the open jet facility wind tunnel (source: Tigran Mkhoyan & Dreux productions)*

# Online Black-box Shape Optimization for a Seamless Active Morphing Wing

## Master thesis

by

O.L. Ruland

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday October 27, 2021 at 14:00.

| | | |
|---|---|---|
| Student number: | 4308220 | |
| Project duration: | September 28, 2020 – October 27, 2021 | |
| Thesis committee: | Dr. ir. R. De Breuker | chair |
| | Dr. ir. C. de Visser | examiner |
| | Dr. X. Wang | supervisor |
| | ir. T. Mkhoyan | supervisor |

*This thesis is confidential and cannot be made public until April 27, 2023.*



TU Delft

# Preface

This report was written towards the fulfillment of the final requirements for obtaining a Master of Science degree in aerospace engineering, with a specialization in aerospace control and simulation at Delft University of Technology. The contents of this document constitute the fruits of a year's worth of hard work. It has been a long and challenging, yet fulfilling journey, the end of which marks the end of my time as an engineering student. During the past year, I have been blessed with the opportunity to work with incredible people and technology. I have been given the chance to explore and peer over the boundaries of the state-of-the-art in a very interesting field and have been encouraged to push them.

However, if I have seen further, it has been by standing on the shoulders of Giants. First of all, I would like to thank Xuerui "Sherry" Wang, my daily supervisor, for her endless positivity, support, and drive, and for investing so much time in me. Thank you for pushing me to write stories worth publishing. Without it, no papers would have ever been submitted. I would also like to thank Tigran Mkhoyan, my second supervisor, for sharing so much of this time and expertise. Tigran has developed the vision system used in the wind tunnel experiment. He also is the proud father of SmartX-Alpha, which has served as the testbed for my work, in every sense of the word – creator, caretaker, protector. I will forever remember: "the wing is alive". Finally, I would like also like to thank Bo Sun for his work on the backlash controller, which was a critical part of the experimental set-up, and Iren Mkhoyan for her assistance in setting up the wind tunnel experiment.

Last but not least, I would like to extend my gratitude to my family: Thijs, Joke, and Fabian for their unconditional support and to my girlfriend Sanne for her limitless support, compassion, and ever graciously postponing our holiday.

*O.L. Ruland*
*Rhodes, October 2021*

# Summary

Today, the need for sustainable aviation is greater than ever before. The transition towards commercially viable low or zero-emission flight may be the greatest challenge facing the aviation industry to date. Promising alternatives to fossil-fuel sources are biofuels and electrical propulsion technologies. Moreover, drag-reducing improvements in wing design can both lower the emissions of fossil-fuel-burning aircraft as well as boost the viability of sustainable alternatives by reducing the amount of energy required for sustained flight. However, in recent years, the traditional aircraft and wing designs have already become highly optimized and the efficiency returns increasingly diminished. Hence, more radical departures from conventional designs are needed to bridge the gap towards sustainable aviation.

Morphing is a bio-inspired technology that enables the shape of the wing to be smoothly altered in flight. Whereas traditional wing designs are optimized for a specific flight condition, usually $50$ %-fuel load cruise, active morphing wings are able to tailor their shapes for maximum aerodynamic efficiency at any flight condition. While the wing shapes of traditional aircraft are determined long before a flight, active morphing technologies allow us to find the optimal wing shape in flight akin to how birds feel the air flow around them and shape their wings accordingly. Compared to the alternative of scheduling predetermined wing shapes based on analytical models, this online form of wing shape optimization could make active morphing wings truly adaptive and further increase their performance.

Strategies for online wing shape optimization have been proposed in the literature. These existing strategies employ local models around a trimmed condition with either direct search methods or gradient-based optimization procedures. In any case, these methods are only capable of finding local minima in the optimization landscapes and require renewed model identification before they can be used at other flight conditions. The model identification procedures typically require sweeps of all morphing actuators which result in increased fuel consumption and undesirable bumpiness in flight, which negatively affects their viability for application on commercial flights. These limitations may be overcome by continuously learning the complex relation between the morphed wing shape and its aerodynamic performance online instead of repeatedly optimizing locally from scratch. The development of such a framework has been the subject of this thesis work.

More formally, the aim of this thesis has been to realize the most aerodynamically efficient shape on a seamless active trailing-edge camber-morphing wing by developing a black-box online-learning shape optimization method and testing the method in a wind tunnel experiment. Although the methodology proposed in this work can be applied to any camber morphing wing platform, its effectiveness has been demonstrated for a particular active morphing wing named SmartX-Alpha[1], in the Open Jet Facility (OJF) of Delft University of Technology.

Before the wind tunnel validation experiment, the new online wing shape optimization framework was developed and tested on a simulation model. This model used the Vortex Lattice Method (VLM) to simulate the aerodynamics of a distributed-trailing-edge-flap adaptation of the SmartX-Alpha wing geometry. Because the VLM method neglects viscous effects, this model was augmented using corrections based on wind tunnel data from previous wind tunnel campaigns with SmartX-Alpha. Within the proposed optimization framework, the shape of the wing is described in terms of virtual inputs that scale the contributions of basis shapes to describe a single curve that dictates the amount of camber morphing across the span of the wing. The goal of the developed framework is to continuously learn the relations between the angle of attack and these virtual inputs, and the resulting lift and drag coefficients and to use this knowledge to determine the most efficient wing shape which satisfies the target lift requirement and does not violate the morphing limits.

The online learning of these mappings was done with Radial Basis Function neural Networks (RBFNs), which are locally sensitive global function approximators that are robust to measurement noise. Two networks were trained in a mini-batch training routine with the adaptive gradient descent algorithm

[1]The project video can be found via `https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s`

Adagrad. The resulting meta-model, also referred to as the on-board model, was queried with populations of candidate inputs by the Covariance Matrix Adaptation – Evolutionary Strategy (CMA–ES) evolutionary optimization algorithm. The desirability of the lift and drag coefficients predicted by the on-board model was subsequently scored using a cost function. This cost function was designed to score predicted outputs depending on their lift-to-drag ratio and the deviation from the target lift coefficient. Finally, the scores of the candidate solutions were used by the CMA–ES optimizer to sample a new and improved generation of candidate solutions. This loop was continued until convergence within each iteration of the optimization framework. The best candidate angle of attack and wing shape combination after convergence was actuated on the "real" system. The measurements resulting from these inputs were then used to further train and improve the on-board model.

Because of the complex relationships between the inputs and the wings' aerodynamic efficiency, and because of the large high-dimensional input space, a large amount of data is required to learn an accurate black-box model. In addition, the measurement times for drag estimation are typically relatively long, i.e., around one minute. Therefore, the on-board model is trained using a combination of recent and historical data. In the simulation and wind tunnel experiment, an initial phase of search space exploration and data gathering dubbed the wandering phase was performed. The pseudo-random inputs evaluated during this phase were generated with a Sobol sequence to guarantee an even density of data throughout the domain. In the real-world application of the framework, this phase would be performed only once during a test flight. In the true operation phase, referred to as the optimization phase, only the most promising inputs are actuated on the morphing wing without any further exploration maneuvers and the on-board model is locally improved as new measurements become available.

Simulations using the augmented aerodynamic model have shown that after initially being trained on a different wing model, the on-board model was able to adapt to the changed system. After a wandering phase of 100 iterations, the proposed methodology was able to realize improved lift-to-drag ratios for a wide range of target lift coefficients without any further model identification maneuvers. Compared to the unmorphed wing jig shape, the aerodynamic efficiency of the wing model was increased by $2.5\%$ to $14.6\%$ for target lift coefficients between $0.25$ and $1.00$.

Subsequently, the proposed optimization framework was demonstrated on the SmartX-Alpha morphing wing platform in an open jet wind tunnel. After 150 iterations of wandering, the framework was operated in the optimization phase mode for 15 iterations with a target lift coefficient of 0.65. The angles of attack and wing shapes actuated during this phase converged onto the target lift coefficient and resulted in higher lift-to-drag ratios than were measured for the unmorphed jig shape. Relative to the jig shape, the aerodynamic efficiency of the wing was improved by $8.4\%$. This corresponds to a $7.8\%$ reduction in aerodynamic drag relative to the wing jig shape at the same lift coefficient. Later attempts with other target lift coefficients were unfortunately unsuccessful after an approximately 1.6-degree bias in the angle of attack actuation was unintentionally introduced. This bias affected both the actuated angle of attack and the estimated lift and drag forces through an angle-of-attack-dependent transformation of the measured forces. However, these biases were corrected in post-processing. The drag reductions predicted by the on-board model after post-processing of the data are between $6.5\%$ and $19.8\%$ for target lift coefficients between $0.35$ and $0.80$. However, the estimations are likely to be optimistic for the lower target lift coefficients as the prediction error for these low total drag points is high in the relative sense. More importantly, post-processing has provided insight into the trends in the optimal wing shape and angle of attack combinations for varying target lift coefficients.

This thesis report details the development and testing of a novel online black-box performance optimization framework for active morphing wings. The proposed method extends the scope of current state-of-the-art online shape optimization methods for morphing wings by learning and optimizing a global meta-model. This eliminates the need for repeated model excitation maneuvers for operation at different flight conditions, which hinder the real-world application of current online shape optimization methods. The effectiveness of the framework has been demonstrated on the SmartX-Alpha active trailing-edge camber morphing wing demonstrator in a wind tunnel experiment. Moreover, due to its black-box nature, the proposed methodology can be applied to any active morphing wing platform without a deep understanding of the system mechanics.

A number of recommendations for future work are made. Firstly, it is recommended that a higher-fidelity aerodynamic model of a trailing-edge camber morphing wing is developed such that the method can be tested and improved for high-speed flight. Secondly, the on-board model should be extended to include the Reynolds and Mach numbers as inputs such that the scope of the method can encompass

the full operational range of airspeeds of the intended platforms. Furthermore, it is recommended that in future work, the optimization bounds are informed and adaptable. In other words, they are linked to a trust region of the on-board model. This would make the method even more robust in real-world applications. Fourth, it is recommended that in future work, the hyperparameters of the on-board model are re-optimized using the newly gathered wind tunnel data. The optimal hyperparameter settings are dependent on the data set used, and this newly gathered data set is more representative of the distribution from which new measurements are 'sampled' than the old simulation-based data set on which the initial hyperparameter optimization was based. In addition, a simplified and more elegant formulation of the cost function is proposed to be used in future work. Lastly, some recommendations are made regarding the changes in sensing equipment that would be necessary to apply the proposed methodology to real-world free-flying aircraft.

# Nomenclature

| | |
|---|---|
| $\alpha$ | angle of attack [deg] |
| $\rho$ | air density [kg · m$^{-3}$] |
| $R_e$ | Reynolds number [-] |
| $\alpha_{\text{act}}$ | actuated angle of attack [deg] |
| $\alpha_{\text{c}}$ | commanded angle of attack [deg] |
| $\Gamma$ | vortex strength [m · s$^{-1}$] |
| $\lambda$ | population size [-] |
| $\mathbf{S}_{x_n}$ | sampled Sobol sequence vector [-] |
| $\mathbf{x}_0$ | initial solution vector |
| $\mathbf{x}_{\max}$ | search space upper bound vector [-] |
| $\mathbf{x}_{\min}$ | search space lower bound vector [-] |
| $\mu$ | dynamic viscosity [ kg · (m · s)$^{-1}$] |
| $\sigma$ | standard deviation [-] |
| $\theta$ | actuator angle [deg] |
| $\varepsilon$ | turntable misalignment angle [deg] |
| $b$ | wing span [m] |
| $c$ | chord length [m] |
| $C_{\text{L}}$ | lift coefficient [-] |
| $C_{\text{L}_\text{t}}$ | target lift coefficient [-] |
| $C_{\text{m}}$ | pitching moment coefficient [-] |
| $D$ | drag force [N] |
| $f$ | flap angle [deg] |
| $F_x$ | force in x-axis [N] |
| $F_y$ | force in y-axis [N] |
| $L'$ | lift force per unit span [N · m$^{-1}$] |
| $L$ | lift force [N] |
| $l_c$ | characteristic length [m] |
| $N$ | number of samples [-] |
| $n_x$ | number of panels in the x-axis [-] |
| $n_y$ | number of panels in the y-axis [-] |

| | |
|---|---|
| $S$ | wing surface area [$m^2$] |
| $t$ | time [s] |
| $T_n$ | Chebyshev polynomial of degree n [-] |
| $t_f$ | objective function variation threshold [-] |
| $u$ | virtual input [-] |
| $V$ | air speed [$m \cdot s^{-1}$] |
| $x$ | spanwise coordinate [m] |
| $x_{\text{hinge}}$ | fractional chordwise hinge location [-] |
| $y$ | spanwise coordinate [m] |
| $z$ | vertical coordinate [m] |
| $z_{\max}$ | vertical trailing edge displacement upper limit [m] |
| $z_{\min}$ | vertical trailing edge displacement lower limit [m] |
| $z_{\text{te}}$ | vertical trailing-edge displacement [m] |
| $\mathbf{U}$ | input vector [-] |

# Contents

# 1 Introduction

In this introductory chapter, the research is introduced and defined. First, the research background and problem are introduced. Secondly, the research questions and the research objective will be stated. Then, the structure of the research project is set out and finally the outline of this thesis report is described.

## 1.1. Introduction to morphing and shape optimization

With the drive towards more sustainable aviation in recent years, the need for more efficient aircraft is ever increasing. And while the first electrically propelled aircraft are making their debut, the future in which electrical flying will be the standard in aviation is still far from being realized. Presently, the range and endurance of electrically-propelled aircraft are typically limited due to the lower energy density of batteries compared to jet fuel. Improvements in the aerodynamic efficiency of aircraft wing design could both reduce the emissions caused by fossil-fuel-burning aircraft as well as increase the viability of electrical alternatives.

The most aerodynamically efficient shape for an aircraft wing is dependent on the aircraft's gross weight, airspeed, and flight altitude. During the flight, all four of these factors tend to vary. This means that at various moments throughout the flight, different wing shapes are optimal in terms of aerodynamic efficiency. Since conventional wings lack the ability to adapt their shape in flight, traditional wing design has always been a compromise between efficiency and effectiveness at various flight conditions. Because of these conflicting wing requirements, designing a wing to perform well at one flight condition means sacrificing performance at some other flight conditions.

Active morphing wings seek to overcome these limitations by emulating nature. Birds can efficiently fly at a wide range of speeds by adapting their geometry to particular conditions in-flight [1]. They can sense the airflow around them and can bend, twist, and fold their wings to achieve good performance, stability, and maneuverability in many different flight conditions. Derived from the Greek word 'morphos', which means shape, morphing is the ability to smoothly change the shape of a structure. Active morphing wings are wings that are able to smoothly alter their shape in flight by some means of actuation. Many examples of different morphing technologies have been published, including but not limited to camber, thickness, sweep and even span morphing. A comprehensive overview of morphing technologies can be found in [2].

However, the freedom to choose between many possible wing shapes to use in-flight gives rise to the question of what wing shape should be used at any point during a flight. The current, and most straightforward way to determine the optimal wings shape is with the optimization of a validated analytical model. The wing shapes are optimized for various flight conditions, and the best shape to use at any point during the flight is selected from a look-up table based on the current flight conditions. The major drawback of scheduling predetermined wing shapes is that this method is not adaptable. In addition, the accuracy of the results is limited by the quality of the model used.

An interesting and promising alternative to the look-up table method is online shape optimization. With online shape optimization, instead of predetermining what shapes are to be used, the best wing shapes are found out during flight. In our comparison to the ways birds operate, their muscles, tendons,

and feathers correspond to the servos, linkages, and flexible skin segments found on many morphing wings. The online shape optimization method would then be analogous to some of the conscious and unconscious processes taking place in bird brains when they are feeling the wind and are figuring out how to best shape their wings. With online shape optimization capability aircraft too could sense and react to their environments. This could improve the performance of morphing wing aircraft and make them truly mission-adaptive.

### 1.1.1. Problem statement

The goal of online shape optimization is to use measurements from the aerodynamic forces acting on an aircraft with morphing wings to determine the optimal wing shape in-flight. As illustrated in Fig. 1.1, the online shape optimization algorithm receives data from aircraft sensors directly and produces an estimate of the best wing shape to actuate on the morphing wings. To date, the domain of reliable and accurate measurement and estimation of lift and drag forces in flight remains a challenging area. However, within the scope of this thesis, it is assumed that noisy, but reasonably accurate measurements of the lift and drag forces are available. Nevertheless, the aim of the optimization algorithm will be limited to the maximization of the steady-state lift-to-drag ratio in trimmed flight, rather than the instantaneous lift-to-drag ratio. This makes the previous assumption more reasonable as the steady-state lift and drag forces can be estimated from the aircraft gross weight, constant flight path angle, and fuel flow over longer periods of time. Furthermore, it allows the aerodynamic efficiency optimization to be combined with other objectives such as (gust) load alleviation on the basis of time scale separation [3]. While the optimal shape for drag reduction serves as a baseline shape that is only slowly adapted, much faster adaptations of this baseline shape can be overlayed by the gust load alleviation optimization. This way multiple morphing objectives can be achieved independently.

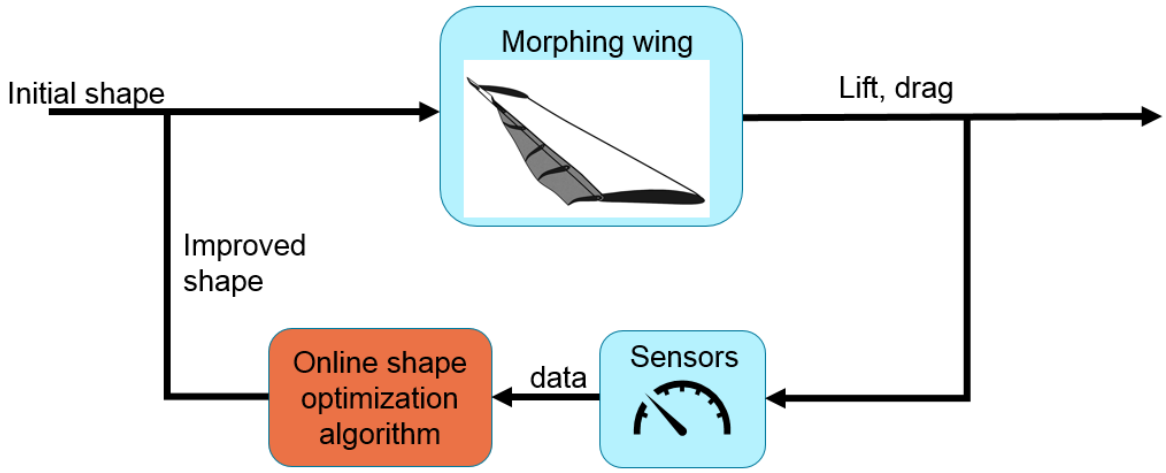



Figure 1.1: Overview of the data-flow between an online shape optimization (orange) and a morphing wing.

Although the methodology proposed in this work can be applied to any camber morphing wing platform, its effectiveness will be demonstrated for a particular active morphing wing named SmartX-Alpha[1], in the Open Jet Facility (OJF) of Delft University of Technology. SmartX-Alpha is a rectangular-planform seamless active trailing-edge camber morphing wing demonstrator that comprises six distributed translation induced camber (TRIC) morphing modules [4].

## 1.2. Research questions and objectives

In this section, the research questions and objectives are presented.

[1]The project video can be found via `https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s`

### 1.2.1. Research questions

**Main research question**

How can the aerodynamic performance of active camber morphing wings be improved with an adaptable black-box online data-driven optimization strategy?

In order to further define the research and answer the main research question, the following sub-questions are formulated:

1. What are suitable modeling methods for trailing-edge camber morphing wings with low computational cost and sufficient accuracy for the purpose of online black-box performance optimization?

2. What are suitable methodologies for an adaptable online wing shape optimization framework?

3. Compared to non-morphing wings of similar geometry, what increase in lift-to-drag ratio can be achieved on active trailing-edge camber morphing wings with online black-box wing shape optimization?

4. What improvements to the methodology are needed to transfer the shape optimization framework to in-flight operation, and what are the challenges?

In order to enable the development and preliminary testing of the online wing shape optimization strategy, a simulation model of a trailing-edge camber-morphing wing is required. To this end, a modeling method needs to be identified that has low evaluation cost such that many evaluations can be afforded, while also being able to simulate the drag mechanics that define the optimization landscape sufficiently. Furthermore, a review of optimization methodologies and their suitability for the intended use case of online shape optimization will have to be performed. After a suitable methodology has been integrated, the effectiveness of the method must be experimentally validated. In the hardware-in-the-loop wind tunnel experiment the method should be demonstrated and the performance benefit should be quantified. Lastly, with regards to future work on the topic, this thesis work aims to investigate the major challenges and possible improvements from which the framework could benefit.

### 1.2.2. Research objective

The main research objective of this thesis is:

**Research objective**

to realize the most aerodynamically efficient shape on a seamless active trailing-edge camber morphing wing by developing a black-box online-learning shape optimization method and testing the method in a wind tunnel experiment.

## 1.3. Research approach

The thesis project consists of four phases as shown in Fig. 1.2. The first two phases correspond to the development of an aerodynamic model of a morphing wing and an online shape optimization method. In the third and fourth phases, the focus is on the evaluation of the online optimization framework on the aerodynamic model and in a wind tunnel respectively.

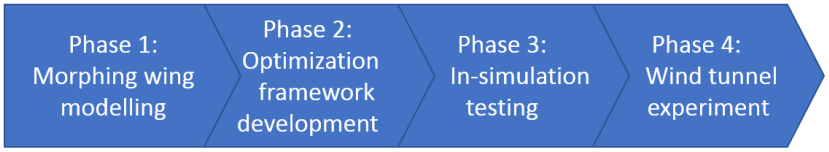



Figure 1.2: Overview of the four main phases of the thesis work.

The major milestones towards the completion of these phases are as follows:

1. Develop an aerodynamic model of a trailing-edge camber morphing wing.
   (a) Aerodynamic solver selection.
   (b) Wing shape parameterization selection.
   (c) Configuration of the aerodynamic model with SmartX-Alpha geometry.
   (d) Aerodynamic model augmentation.
   (e) develop a surrogate model.
2. Implementation of optimization framework.
   (a) Select black-box optimization method.
   (b) Optimize optimization configurations.
   (c) Select model type for on-board model.
   (d) Create offline training data set.
   (e) Optimize on-board model hyperparameters.
3. Testing of optimization framework on an aerodynamic model.
   (a) Identify best shapes and corresponding performances on a nominal aerodynamic model.
   (b) Simulate online shape optimization framework operation on a surrogate aerodynamic model.
4. Testing of optimization framework in wind tunnel campaign.
   (a) Measure the performance of SmartX-Alpha with wing jig shape.
   (b) Measure the performance of SmartX-Alpha with optimal shapes from the aerodynamic model.
   (c) Measure the performance of SmartX-Alpha with black-box online shape optimization.

## 1.4. Thesis outline

The structure of this thesis is as follows. In chapter 2 a literature review of online wing shape optimization research is presented. The design and verification of the aerodynamic model and the online shape optimization framework are discussed in chapter 3. Then, in chapter 4 a conference paper that has been accepted by the AIAA 2022 SciTech Guidance, Control, and Dynamics conference on the framework and simulation results is presented. The procedures followed during the subsequent wind tunnel experiments, supportive material, and a summary of the findings are discussed in chapter 6. Then, in chapter 5 the paper on the wind tunnel experiments that will be submitted to the IEEE journal of transactions on control systems technology is shown. The results from the simulation and wind tunnel experiments will be compared in chapter 7. Lastly, in chapter 8, the research questions will be answered and recommendations are given for future work.

# 2

# Literature review of online wing shape optimization research

In this chapter, a brief history of the advancements in online wing shape optimization research is presented. First, the research of NASA Dryden on the online optimization of redundant control surfaces on conventional wing platforms is reviewed. secondly, the online optimization of full-span distributed leading and trailing-edge control surfaces using a generating search set method is described. Thirdly, a strategy for wing shape optimization in terms of induced drag through lift distribution matching is discussed. Next, the current state-of-the-art methods for online wing shape optimization of active morphing wings proposed by NASA Ames are reviewed. Finally, some conclusions regarding the implications for the current thesis research are drawn.

## 2.1. Online optimization of redundant control surfaces

Before recently renewed interests in morphing technologies, research has been conducted on real-time adaptive configuration optimization for conventional wing platforms. Researchers from the NASA Dryden Flight Research Center have proposed and demonstrated adaptive model-free methodologies for the measurement-based in-flight optimization of redundant control effectors for conventional wing aircraft.

In 1999, a practical application of an in-flight, real-time, adaptive configuration optimization algorithm was demonstrated on an L-1011 aircraft that was modified with variable-camber capability in the form of symmetric deflection of the outboard ailerons [5]. The optimization process consisted of a smooth low-frequency forced excitation maneuver and regression analysis to identify drag equation coefficients until convergence. The forced excitation maneuver was required to be slow such that quasi-equilibrium was maintained as the method assumes steady-state flight. In the drag model used, the optimal symmetric aileron position was one of the regression-estimated coefficients [6]. The outboard ailerons were then commanded to this optimally determined position. During the flight tests, a drag reduction of $1\%$ was achieved, even though the outboard ailerons only make up approximately $23\%$ of the L-1011's wingspan and approximately $3\%$ percent of the total wing area. Wing platforms with a greater range of shape adaptation are better able to harness the potential drag reductions of morphing. But with increasing numbers of morphing actuators and fewer constraints on the attainable wing shapes, the corresponding optimization problems become increasingly more difficult to solve. Because the system under consideration only numbered a single degree of freedom, the optimal input could be directly estimated through the regression analysis. However, for complex nonlinear higher-degree of freedom systems, such as morphing wings, this is no longer possible. Instead, a more sophisticated optimization strategy is required.

In 1995, a periodic perturbation extrema-seeking technique was proposed for direct adaptive performance optimization of subsonic transport aircraft [7]. The proposed method superimposed sinusoidal perturbation signals with unique frequencies on each of the decision variables (control effectors). The components of the local gradient of a performance index with respect to the decision variables were then approximated with an online estimation of the correlation between the sinusoidal probing signals

and this performance index. Subsequently, the estimated gradient was used as the search direction to improve the performance index, and by extension the aircraft performance. While this optimization method is gradient-based, it is based on averaging and signal integration rather than signal differentiation. This makes the method robust to measurement noise and biases. Velocity maximization and fuel-flow minimization modes of the algorithm were demonstrated on the NASA Dryden B-720 nonlinear flight simulator for single- and multi-effector optimization cases in [8]. The peak-seeking control algorithm was designed as an add-on to an existing nonlinear dynamic inversion inner-loop controller and research autopilot. The algorithm directly added biases to the deflections of the control effectors, which were smoothly transitioned to minimize transients. During the transient response, the inner-loop controller and auto pilot adjusted the stabilators and throttle setting to maintain steady-sate flight. The resulting noisy fuel-flow measurements were time-averaged. The effectiveness of this steepest descent method is heavily influenced by the tuning of the negative gain which is multiplied with the estimated gradients to generate the next control surface commands.

In 2013, a peak-seeking control approach was proposed for real-time trim configuration optimization for reduction of aircraft fuel-consumption was proposed [9]. The methodology is based on a steepest-descent algorithm that uses a time-varying Kalman filter to estimate the gradients of the fuel flow with respect to the control surface positions. In essence, this approach is similar to the method proposed in [7], however, the measurement noise- and bias-robust estimation of the performance gradients are made with a Kalman filter rather than with signal correlation. The method has been demonstrated to achieve $2.5\%$ reduction in fuel flow to the relative trim of the aircraft by manipulating the deflections of symmetric ailerons, trailing-edge flaps, and leading-edge flaps of a nonlinear simulation model of an F/A-18 airplane. In flight tests with an F/A-18 airplane, the proposed method was demonstrated to achieve approximately $3\%$ of fuel-flow reduction relative to the baseline trim at the same flight condition [10]. In these flight tests, the algorithm consistently converged onto the same solution from different initial conditions. However, the high degrees of freedom of over-actuated active morphing wing platforms pose high-dimensional search spaces with possibly many local optima. In these non-convex optimization landscapes, gradient based-methods are prone to converging on local optimal solutions [11]. While restarting the optimization procedure from different initial conditions is a commonly practiced solution for work on simulation models, such restart strategies are unsuitable for in-flight applications to aircraft due to the bumpiness and increased fuel consumption they cause.

## 2.2. Optimization with the generating set search method

In 2006, Jacobsen demonstrated real-time drag reduction on a wing model with eight leading-edge and eight trailing-edge control surfaces in a wind tunnel [12]. Because of the unreliability of derivatives computed from the measured signals, a derivative-free generating set search (GSS) method was used. GSS is a non-linear iterative solver that needs to be initiated within the feasible region of the search space and adjusts the search direction in the input space without crossing the boundaries posed by the optimization constraints. The method generates a set of search directions around the current iterate which are adapted to the geometry of the nearby constraints. After function evaluation in the search directions, the current iterate is replaced when better performing points are identified. The proposed method was generalized by taking both linear equality as well as linear inequality constraints into account. Before optimization, a sweep of the control surfaces was performed to identify the parameters of a linear lift function which poses the linear fixed-lift constraints. The GSS optimizer was then able to reduce the drag on the wing model, whilst maintaining lift for a fixed angle of attack by iteratively adjusting the deflections of the leading- and trailing-edge control surfaces. The angle of attack was not included as a free variable because of the long time required for actuation. It is noteworthy that the same lift could potentially be achieved with an even lower drag at another angle of attack. However, at other angles of attack, the coefficients of the linear model need to be identified again with additional sweeps of the control surfaces. Lastly, while improved deflections were found which yielded lower drag, these solutions possibly only represent local optima in the optimization landscape, as GSS is local in nature [13].

## 2.3. Optimization through lift distribution matching

In that same year, Kolonay and Eastep demonstrated an inverse optimization technique to manipulate the spanwise lift distribution of a combined structural and aerodynamic model of a flexible wing to mini-

mize lift-induced drag [14]. The objective function for the optimization was the sum of the square of the errors between the desired (elliptical), and actual lift distributions at 40 spanwise locations. Iterative optimization of an approximate problem was used to eliminate the need for the trimmed angle of attack and the elevator deflection to participate in the design optimization as free variables. The wing models used were comprised of 20 discrete active conformal trailing edge control surfaces distributed along the entire wingspan. This shape adaptation capability of this kind of seamless distributed camber morphing model is similar to that SmartX-Alpha. The optimization strategy has some important limitations regarding practical implementations. Most importantly, the elliptical load distribution matching is only effective for minimizing lift-induced drag. This means, that in actuality other drag components could be increased to attenuate or even nullify the gains from the reduced lift-induced drag. An example could be increased form drag from the fuselage because of an increased angle of attack. Furthermore, as the strategy involves scheduling the model-based optimal deflections, the accuracy of the methods is limited to the quality of the model used and not adaptable. Since accurate estimation of the spanwise load distribution in flight is very difficult, the load distribution matching strategy is not very well suited for online performance optimization.

## 2.4. Online grey-box shape optimization for morphing wings

Researchers at NASA Ames Research Center have proposed online performance optimization methodologies for active morphing wings. In 2016, a real-time adaptive least-squares drag minimization approach was proposed by Ferrier, Nguyen, and Ting [15] for the Variable Camber Continuous Trailing Edge Flap (VCCTEF) described by Nguyen et al. [16, 17]. The VCCTEF system, shown in Fig. 2.1, comprises 19 spanwise flap sections. The discrete flap sections are connected through flexible elastomer covering. This allows the VCCTEF to adapt its spanwise camber distribution. However, variation of local camber is only possible between different spanwise sections, but not within a section. In contrast, the TRIC morphing mechanism employed by SmartX-Alpha allows for both inter- and intra-modular variation of the wing camber allowing for smoother and more continuous control of the spanwise wing twist.

The proposed optimization strategy used a recursive least squares algorithm to estimate the derivatives of the aerodynamic coefficients with respect to the system inputs, which are the parameters of the lift, drag, and moment coefficient models. The lift and moment coefficients are modeled as linear functions, whereas the drag coefficient is modeled as a quadratic function. These models are only valid in the local region around the trim angle of attack. The optimal wing shape and elevator deflection were then calculated from a constrained optimization problem using the Newton-Raphson method. Constraints were posed on the lift and moment coefficients and the wing tip twist and bending.

Improvements to the model excitation method, on-board model, and optimization methods were demonstrated in wind tunnel experiments to achieve up to $9.4\%$ drag reduction on the Common Research Model (CRM) with the VCCTEF at off-design conditions at low subsonic speeds in 2019 [18]. In 2021, simulations have also indicated that a $3.37\%$ drag reduction is achievable using the online optimization strategy on the CRM with a distributed mini-plain flap system at Mach 0.85 [19].

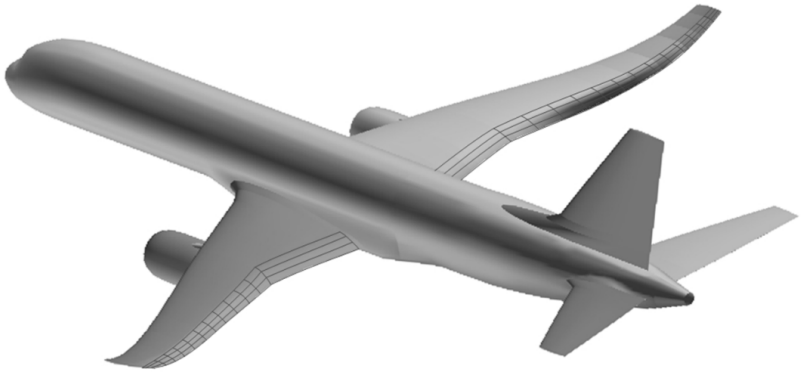


Figure 2.1: An aircraft with Variable Camber Continuous Trailing Edge Flap [15].

Whilst the coefficients of the linear-in-the-parameters multivariate polynomial model adopted in [18, 19] can be estimated with relatively low computational cost, the model is only valid in the local region around the trim condition. This means that in order to perform real-time drag minimization across the entire flight envelope, the model parameters need to be re-estimated at every operational point. Another drawback of the employed model is that the model structure is fixed. However, the optimal model order of the polynomial model in the various inputs dimensions may be different for different flight regimes.

Moreover, the required model excitation maneuvers that comprise both angle of attack and flap deflection inputs would induce undesirable bumpiness and structural loads. Last but not least, the use of a local model together with a gradient-based optimization method makes the solution prone to converge onto a local optimum.

## 2.5. Proposed research direction

By contrast, a global on-board model, while more difficult to identify online, could allow for continuous drag minimization throughout the flight envelope without any additional model excitation maneuvers. Moreover, when paired with a global optimization method, global optima with even better performance could potentially be found. For this reason, it is proposed that a novel online optimization strategy is investigated which attempts to maintain global knowledge of the input space. Furthermore, it is proposed that this method should update this global knowledge online through an adaptable black-box model. The black-box model should not assume a model structure based on possibly lacking a priori knowledge of the complex nonlinear relationships it will attempt to learn, but rather be able to learn any mapping accurately given enough data.

# 3

# Simulation validations

In this chapter, the development and component-wise validation of the online optimization wing shape optimization framework is decribed. An description of the aim, architecture and simulation results is presented in the paper which has been accepted for the AIAA 2022 SciTech Guidance, Control, and Dynamics conference (Chapter 4). This chapter does not seek to repeat the information in that chapter, but rather to expand on it. The structure of this chapter is as follows. First, the development and configuration of the developed aerodynamic model is described in sec. 3.1. Then the selection of the optimization strategy and the tuning of the relevant parameters is described, and insight is provided into the results of offline optimization on the aerodynamic model in sec. 3.2. In sec. 3.3, the selection and configuration of the on-board model are described, and the prediction quality of offline- and online-trained models are compared through parameter sweeps. Lastly, a description of the algorithm and the phases used in the experiments is provided in sec. 3.4.

## 3.1. Aerodynamic model

In this section, the development of the aerodynamic model is presented. First, the aerodynamic modeling method and solver selection are described, followed by a description of the model geometry and the solver configuration. Subsequently, the parameterization of the wing shape and an analysis of the experimental data from previous campaigns used for the correction of the aerodynamic model are presented.

### 3.1.1. Method and solver selection

The aerodynamic model was designed to satisfy the following objectives:

- demonstrate the functionality of shape optimization;
- predict the trends in optimal shapes;
- predict the attainable lift-to-drag ratio increases (or drag reductions);
- serve as a platform for initial training of the on-board model.

From these goals, the following requirements are formulated:

1. The method should have low computational cost.
2. The method should have sufficient accuracy to serve as a starting point for the initial training of the on-board model.

The behavior of the model should be close enough to the real morphing wing such that the trends in the optimal shapes predicted on the model are representative of the trends in the optimal shapes of the real morphing wing system. However, as the optimization method is designed to learn and adapt online, the information from the aerodynamic model will only serve as a starting point. Hence, limited

model accuracy is tolerable and this needs to be traded off against the associated computational time. The computational time per evaluation is an especially important aspect of the aerodynamic model as black-box optimization methods, in general, require many function evaluations to converge. For this reason it is also of paramount importance that the aerodynamic solver in which the method will be implemented can be integrated with either Python or Matlab such that function evaluations can be performed and processed automatically by the optimizer without any manual steps.

Based on these requirements the Vortex Lattice Method (VLM) was selected as the aerodynamic modeling method for the aerodynamic model. VLM is an inviscid 3D potential-flow method that divides the wing geometry into a number of panels. A discrete horseshoe vortex of unknown strength $\Gamma_{ij}$ is located at the quarter chord of each panel. The lift and induced drag of the given geometry in a given flow condition is then estimated through the calculation of these vortex strengths. A prime driver for the accuracy of the VLM is the number of panels used in the discretization of the wing geometry. The VLM aerodynamic model includes the following assumptions [20]:

1. The airflow is assumed to be incompressible. This assumption limits the model validity to air speeds below Mach numbers less than 0.3.

2. The airflow is inviscid. This assumption limits the model validity to Reynolds numbers much greater than 1.

3. It is assumed that no flow separation effects occur. This assumption limits the validity of the model to angles of attack for which no flow separation occurs.

4. The effects of the thickness of the airfoil or wing on the lift and induced drag are negligible.

The primary reason for the selection of this method is the low computational cost. Even though VLM is considered a low-order method, the methods accuracy is sufficient for the purpose of initial training of the on-board model. Moreover, through the addition of data-driven corrections, a low order estimation of viscous effects was included such that the trade-off between the wing camber and the angle of attack as mechanisms of lift induction could be simulated as well.

**Solver selection** For the implementation of the VLM aerodynamic model, it was decided to use an existing aerodynamic solver. An overview of the considered solvers is shown in Tab. 3.1. In this context the word scriptability is used to refer to the degree in which a particular solver supports automated configuration of the geometry and evaluation of the aerodynamic solution through its own or other programming environments. At this stage in the thesis project, the programming language to be used for development and simulation of the online wing shape optimization framework was not yet fixed. Therefore, both VLM solver implementations available in Matlab and Python were considered. The widely used solver XFLR5 has no scriptability and thus was not suitable for integration in an automated framework. However, due to its widespread use and proven implementation of the VLM, XFLR5 was used to verify the results of the finally selected aerodynamic solver, which had only recently been released. Therefore no published code verification examples from trusted sources were available.

The Athena Vortex Lattice (AVL) solver implementation was also not chosen because it does not support direct scriptability in either python or Matlab. Instead, those programs can only be used configure runfiles for AVL. While not impossible, this option is more difficult to implement than direct scriptable solvers with no inherent benefit. Futhermore, Flow5, a the scriptable successor to XFLR5, had to be dropped due to the cost of it's license. Both Aerosandbox and Tornado were viable options for the implementation of the VLM aerodynamic model. Since, good options were available for both Matlab- and Python-based implementation, the final decision was made to use Python for all simulation work based on personal preference and experience. Hence, the VLM method was implemented using the Aerosandbox package [21] in python.

### 3.1.2. Geometry definition

An overview of the SmartX-Alpha wing geometry and the axes-system used in the following descriptions are shown in Fig. 3.1. The $0.5$ m $\times$ $1.8$ m rectangular half-wing comprises six morphing modules which are seamlessly joined with a flexible elastomer skin. The body-fixed reference frame referred to in the following discussions originates at the leading edge of the wing root end. The positive x-axis is aligned

Table 3.1: Overview of aerodynamic solvers.

| Solver | method | computational cost | scriptablility | cost |
|---|---|---|---|---|
| XFLR5 | VLM | low | none | free |
| Flow5 | VLM | low | Python | paid |
| Aerosandbox | VLM | low | Python | free |
| AVL | VLM | low | indirect* | free |
| Tornado | VLM | low | Matlab | free |

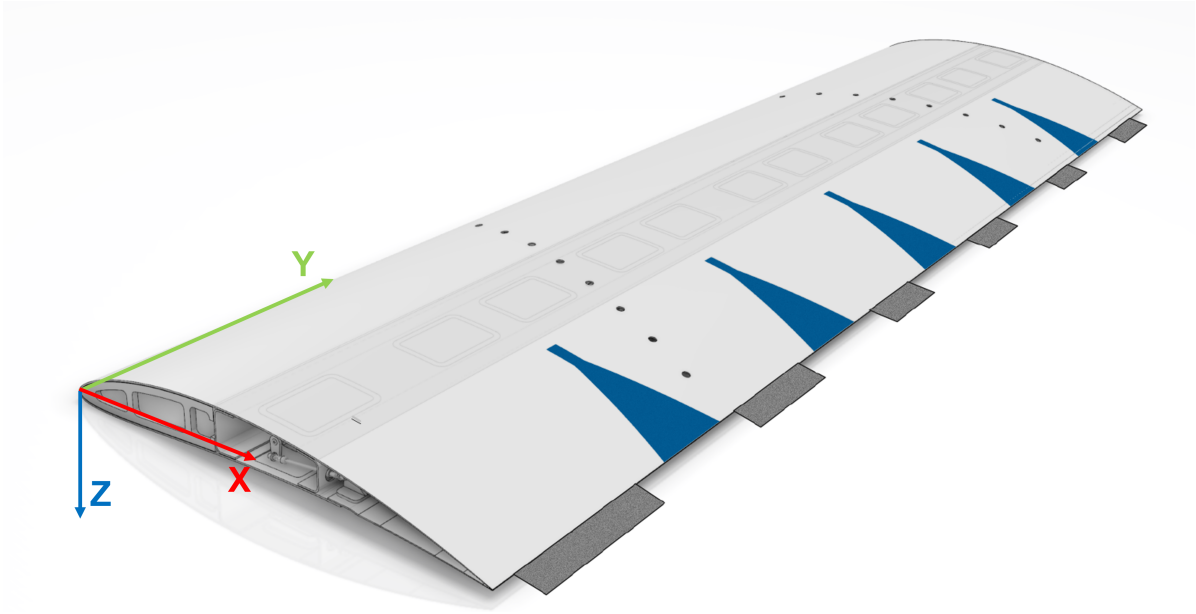



Figure 3.1: Primary axes of the body fixed-reference frame.

with the wing chord, the y-axis is defined along the spanwise direction, and the positive direction of the z-axis is defined perpendicular to the xy-plane and in the direction of camber increase (downwards).

The Translation Induced Camber (TRIC) morphing mechanism allows for both camber and twist morphing of the modules [22]. The exact 3D deformations of the morphing modules and the elastomer skin between them under loads are complex and would require a high fidelity coupled aero-structural model. Such models are computationally more expensive than the lower order VLM model selected, and their improved accuracy is not required for the research in this thesis as explained in sec. 3.1.1. Instead, the camber and twist morphing wing is represented by a full-span distributed-flap wing model. In this model, the airfoil sections comprise a NACA6510 base airfoil with a single-hinged flap. The horizontal location of the flap hinge was selected as $77\%$ of the chord length measured from the leading edge, and the vertical location was selected as $50\%$ of the local airfoil thickness. The selection of the flap hinge point was based on Digital Image Correlation (DIC) measurements of symmetric deflections of SmartX-Alpha. Fig. 3.2 shows a comparison between the baseline NACA6510 airfoil (black), the DIC measured shape resulting from maximum upwards deflection of the trailing edge (green) and the flapped NACA6510 airfoil (red).

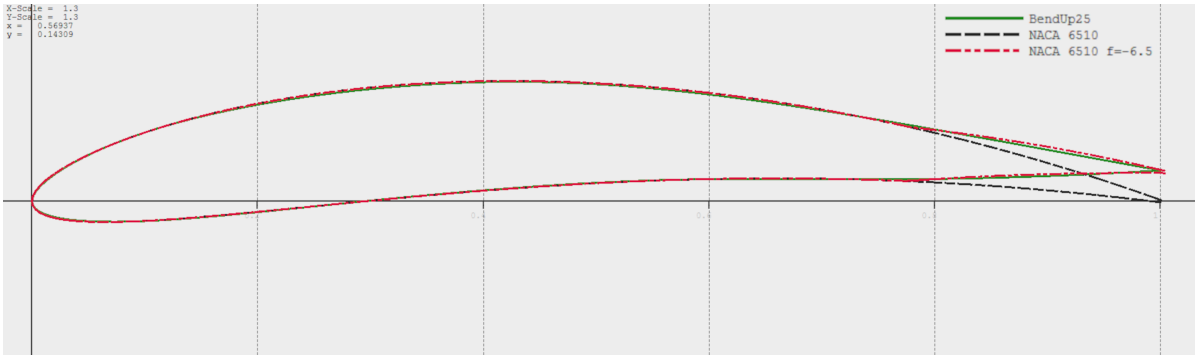



Figure 3.2: Comparison of a flapped NACA6510 airfoil and morphed airfoil shape of SmartX-Alpha.

The geometry of this distributed-flap wing is defined by flapped airfoil sections at 14 spanwise locations $y_0, \ldots, y_{13}$. The flap angles at these locations are designated $f_0, \ldots, f_{13}$.These 14 spanwise wing stations comprise the wing root, wing tip and the 12 morphing actuator locations as shown in Tab. 3.2, where the designation m$i$s$j$ corresponds to the location of servo unit $i$ in module $j$.

Table 3.2: Spanwise locations of station locations on the aerodynamic model of the morphing wing.

| station | y [m] | station (con'd) | y [m] (con'd) |
|---|---|---|---|
| root | 0.00000 | m4s1 | 0.91440 |
| m1s1 | 0.01448 | m4s2 | 1.18553 |
| m1s2 | 0.28553 | m5s1 | 1.21448 |
| m2s1 | 0.31448 | m5s2 | 1.48553 |
| m2s2 | 0.58553 | m6s1 | 1.51448 |
| m3s1 | 0.61448 | m6s2 | 1.78553 |
| m3s2 | 0.88553 | tip | 1.80000 |

The morphing of the airfoil section at the servo locations can be directly controlled by the respective servos. However, no actuators are located at the wing root and tip ends of the wing, and the shape of the wing at these locations can only be indirectly controlled and is determined by the deflections of the adjacent actuators. Therefore, it is assumed that the spanwise variation of camber within a twist morphing module is linear. Under this assumption, the flap angles at the root and tip location $f_0, f_{13}$ can be calculated by extrapolation of the slope of the flap angle in the first and last modules respectively as shown in Eq. (3.1)

$$\begin{aligned} f_0 &= f_1 - (y_1 - y_0) \cdot \frac{f_2 - f_1}{y_2 - y_1} \\ f_{13} &= f_{12} + (y_{13} - y_{12}) \cdot \frac{f_{12} - f_{11}}{y_{12} - y_{11}} \end{aligned} \tag{3.1}$$

**Model inputs** As described, the shape of the distributed flap model is governed by the flap deflection vector $f \in \mathbb{R}^{14}$. However, on the real morphing wing, these 'equivalent flap angles' do not exist, and cannot be set. Instead, in reality, the shape of the morphing wing is controlled through the actuator deflections $\theta \in \mathbb{R}^{12}$. Hence, a conversion of the actuator angles to equivalent flap angles is needed. This conversion is done in two steps. First, the actuator angles are used to approximate the displacements of the wings' trailing edge. This is done with Eq. (3.2). The slope of the trailing-edge displacement $z$ with respect to the actuator angle $\theta$ was estimated using DIC-measurements at symmetric actuator deflections from $-25° \leq \theta \leq 25°$. The contributions of the adjacent servos in the case of asymmetric actuation are neglected such that the trailing-edge displacement $z_i$ of each servo station is only determined by the deflection $\theta_i$ of the servo unit at this location.

$$z_i = \theta_i \cdot 5.6 \times 10^{-4} \tag{3.2}$$

The second step is the conversion of the trailing-edge displacements to their equivalent flap angles. The geometric relation between vertical trailing-edge displacement and the equivalent flap angle is given by Eq. (3.3). Here, $c$ is the local airfoil chord length and $x_{\text{hinge}}$ is the horizontal location of the flap hinge measured from the leading edge as a fraction of the chord length.

$$f_i = \arcsin\left(\frac{z_i}{c(1 - x_{\text{hinge}})}\right) \tag{3.3}$$

### 3.1.3. Solver setting configuration

One of the most critical settings of the aerodynamic solver is the selection of the geometry discretization parameters. The wing surface is discretized into a finite amount of rectangular panels. For each panel, the vortex strength $\Gamma$ of a horseshoe vortex is calculated. The discretization of the wing geometry represents a trade-off between model accuracy and computational cost. With increased panel density, the model accuracy is increased. However, with increasingly finer meshes the accuracy return diminishes while the computational costs increasingly increase. Therefore, the panel discretization parameters need to be carefully selected such that the model yields accurate predictions against an affordable computational cost. To this end, the model predictions and computational times of the model with different discretization are compared. The discretization parameters are $n_x, n_y$ which represent the number of panels in the x and y directions respectively. In the spanwise y-dimension, the shape of the wing is described at the 14 locations shown in Tab. 3.2. That is why the parameter $n_y$ corresponds to the number of panels per spanwise section. Hence, the minimum number of panels in the y direction is 13, which corresponds to $n_y = 1$. The wings shape in the following test case comprised a linear spanwise variation of the local flap angle from the minimum to the maximum angle and an angle of attack of zero degrees. The distribution of the panel lengths was uniform in the spanwise direction and in the chordwise direction a cosine distribution was used to refine the mesh near the wings' leading and trailing edges. The resulting variations of the model outputs and computational time with these discretization parameters are shown in Fig. 3.3.

From the predicted lift and drag coefficients, it is clear that increased values for $n_y$ have an insignificant effect on the model outputs compared to the effect of $n_x$. At the same time, the number of panels, and hence the computational time grows rapidly for increased values of $n_y$. The slopes of the model output with increased values for $n_x$ is initially large but diminishes rapidly to an almost indiscernible slope near $n_x = 10$. Therefore, in an effort to limit the computational time required, the democratization parameters are selected as $n_x = 10, n_y$. This resulting mesh comprises 10 panels in the chordwise direction and 13 panels in the spanwise direction, which amounts to 130 panels in total across the 1.8m half wingspan.

An overview of the configurable aerodynamic solver parameters and their selected values is presented in Tab. 3.3. The air density was selected as the sea-level air density of the International Standard Atmosphere. An airspeed of 10 m/s was used, which was consistent with the expected wind tunnel speed to be used in the later experiments. However, the wind tunnel experiments were performed with a wind speed of 15 m/s. While this difference may seem significant, this is only the case for the actual lift and drag forces. The effect on the dimensionless lift and drag coefficients is negligible in this case.

Table 3.3: Overview of the configured parameters of the aerodynamic solver.

| parameter | value | unit |
|---|---|---|
| airspeed | 10 | m/s |
| air density | 1.225 | kg/m^3 |
| x panels | 10 | - |
| y panels | 13 | - |

Finally, the geometry described thus far is mirrored in the wing root x-z plane to create a full wing. This is done so that only the tip of the wing constitutes a 'free' end where the pressure difference between the top and bottom wing surfaces is allowed to equalize, giving rise to tip vortices. In other words, the interface between the wing root and aircraft fuselage is assumed to be ideal. While the SmartX-Alpha demonstrator is only a half wing, a full wing would be integrated into aircraft in a real-world application. During the wind tunnel experiment involving SmartX-Alpha, a flow table will be used

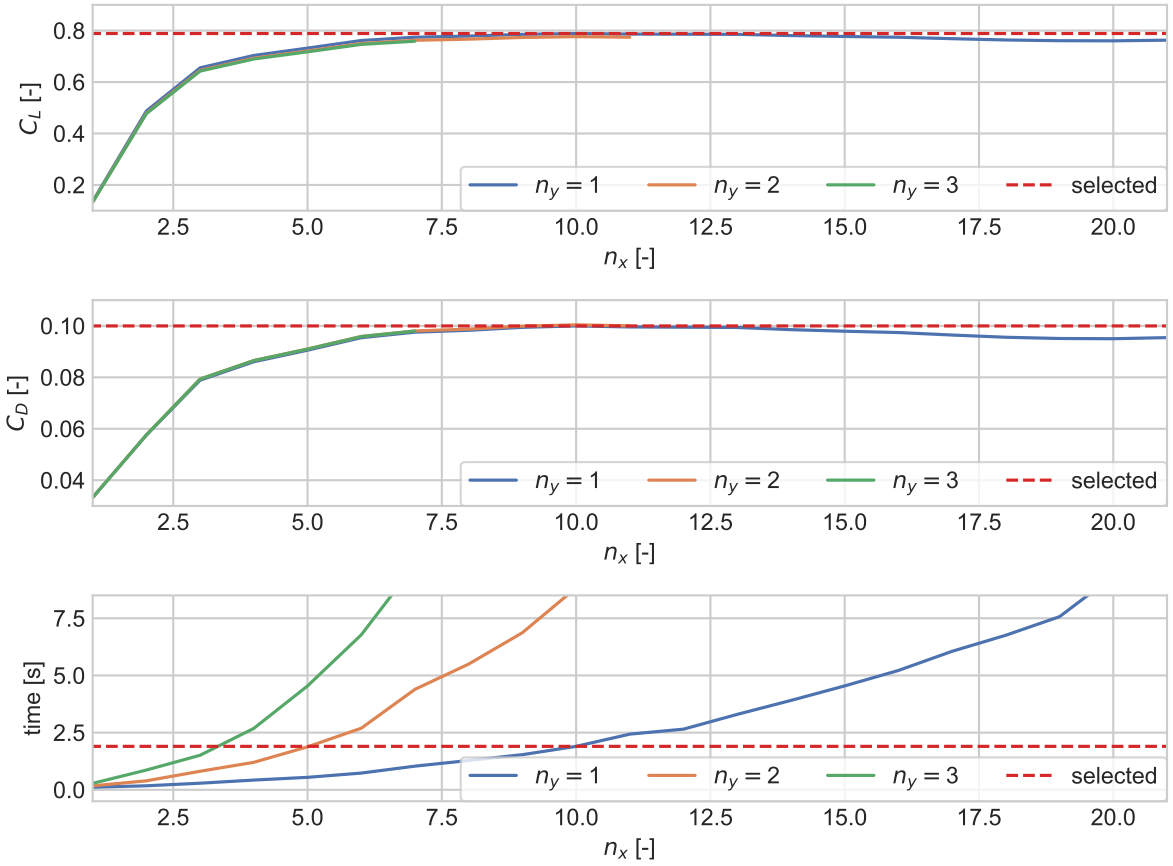



Figure 3.3: Variation of computational time and aerodynamic model outputs with panel mesh parameters.

to simulate the aircraft fuselage such that the pressure field will be similar to the real-world situation. The final geometry discretization and the streamlines in an example flow condition are illustrated in Fig. 3.4. Note that vortices characterized by the spiraling shapes of the streamlines only occur at the two wingtip ends.

### 3.1.4. Virtual shape functions

In the case of over-actuated active morphing wings, like SmartX-Alpha, the wing shape is controlled with many morphing actuators. And while in high-dimensional input space of such a system makes the number of theoretically possible inputs virtually limitless, in practice the achievable and desirable deflections of the various actuators are not independent. In other words, there exists a sub-region of the input space which is of interest in practice. Hence, by limiting the problem to such a sub-region of the input space, the dimensionality of the optimization problem may be reduced at the price of little to no reduction in the quality of the solutions found.

In the application under consideration, a certain degree of smoothness is required of desired wing shapes if they are to be actuated on the system. This requirement constitutes a coupling between the morphing actuators that can be exploited. And by describing a parameterized wing shape, rather than the actuator inputs, the dimensionality of the optimization problem becomes independent from the dimensionality of the input space of the morphing wing platform on which the framework is applied.

To this end, the wing shape is parameterized with a fifth order Chebyshev approximation. Since these wing shape parameters are not inputs to the physical system, they are dubbed “virtual inputs”. The five virtual inputs $u_1, \dots u_5$ are scaling factors that scale the first five Chebyshev polynomials of the first kind which are shown in Eq. (3.4). The chebyshev polynomials are bounded to $-1 \leq T_n(x) \leq 1$ on the $[-1, 1]$ domain. The wing shape is then given by the linear combination of these basis functions as shown in Eq. (4.1).

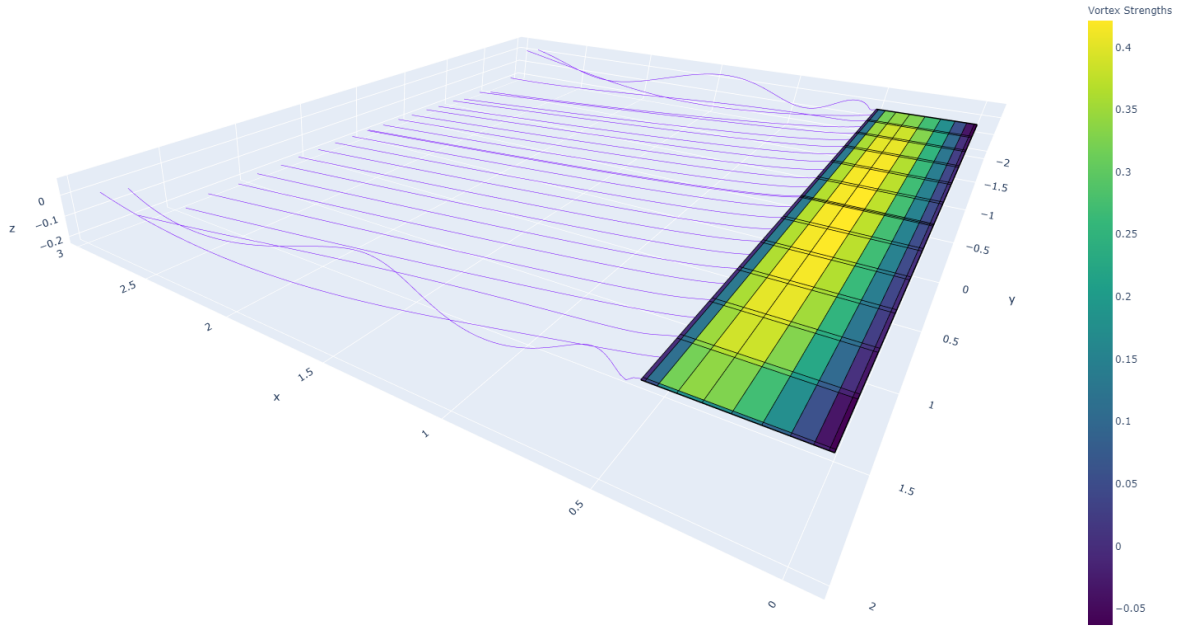



Figure 3.4: SmartX-Alpha wing geometry and panel mesh in Aerosandbox.

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1
\end{aligned} \tag{3.4}
$$

The five base shapes defined by the virtual inputs, re-scaled onto the [0,1.8m] domain, are depicted in Fig. 3.5. SmartX-Alpha's 12 actuator locations are indicated with the black vertical lines. The contributions of the unit virtual inputs at each of the actuator locations are marked with the triangles in their corresponding colors. The complexity of the basis shapes increases with their number. The selection of the number of virtual inputs to use represents a trade-off between the order reduction of the optimization problem and the complexity of the describable wing shapes.

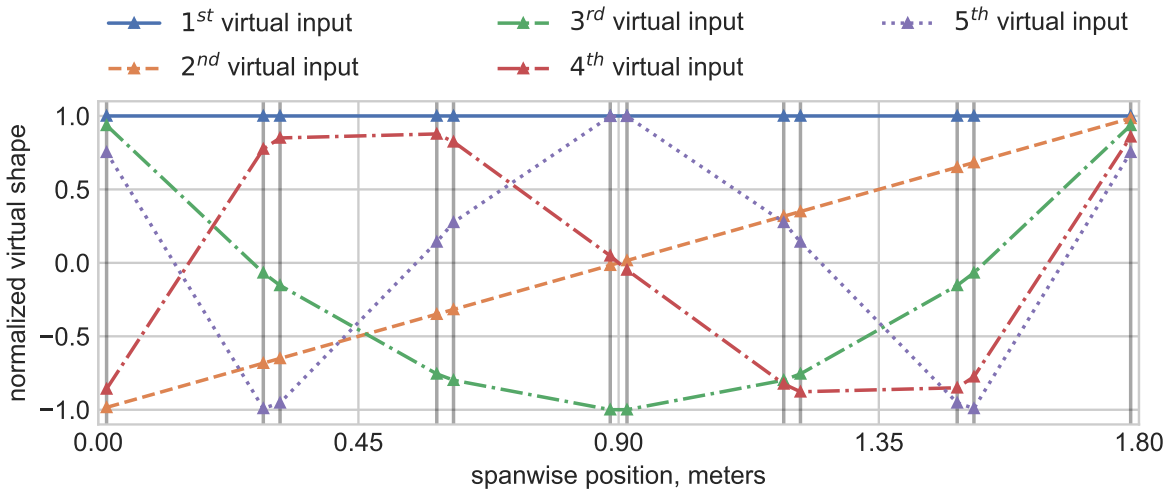



Figure 3.5: Virtual input basis functions and their contributions at the actuator position locations (marked by triangles).

$$z(y) = \sum_{i=1}^{5} u_i T_i(y) \tag{3.5}$$

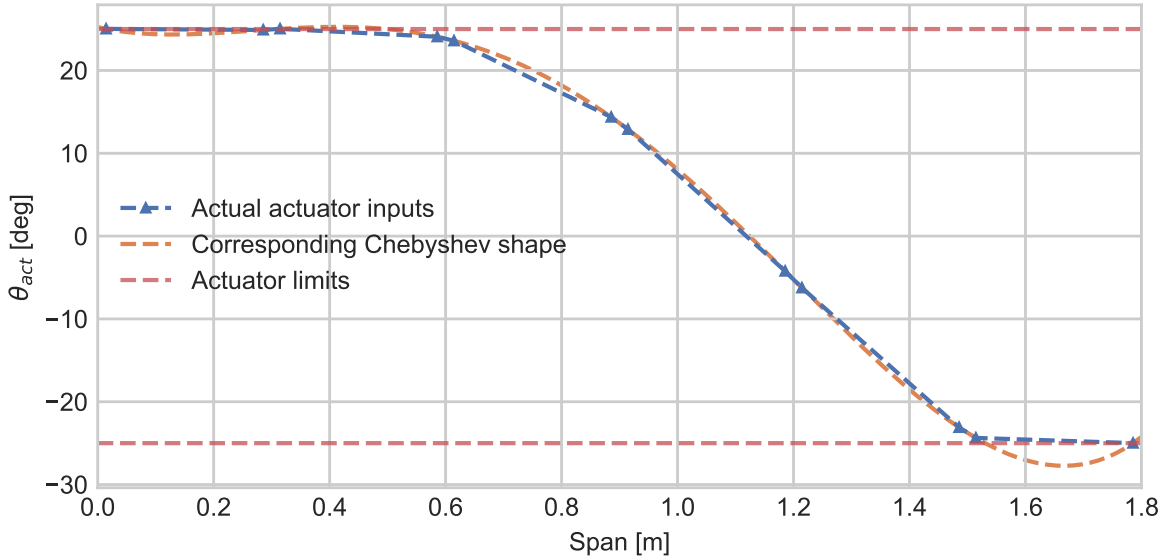



Figure 3.6: Virtual input shape (orange) and corresponding actuator deflections (blue).

Note that the virtual shape function can reside outside the actuator limit bounds outside of the actuator locations. In Fig. 3.6, this is the case between the $11^{\text{th}}$ and $12^{\text{th}}$ actuator positions. However, while the virtual shape function locally 'breaches' the limits, none of the actuators are assigned a deflection outside their allowable range.

### 3.1.5. Model augmentation data set analysis

In this section, insight is provided into the data set on which the aerodynamic model corrections were based. This data set was collected in the Open Jet Facility (OJF) wind tunnel at Delft University of Technology in August 2020. The data set comprised lift and drag coefficient measurements for angle-of-attack sweeps with various degrees of constant spanwise camber morphing actuated on SmartX-Alpha. While this data set does not provide insight into the optimal spanwise camber distribution, it does hold information on the optimal trade-off between the overall amount of camber and the angle of attack used for lift generation in this flow regime.

From the lift-to-drag ration plot in Fig. 3.7, it is evident that the highest degree of negative camber morphing (cyan) offers the highest lift-to-drag ratio for angles of attack greater than four degrees. However, the aerodynamic efficiency envelope of the morphing wing is determined by the drag costs associated with an entire range of amounts of lift production. Hence, a more compelling overview is obtained from looking at the lift-to-drag ratios against the range of operation lift coefficients.

A clear and intuitive plot is obtained by converting the lift-to-drag ratio to a relative increase compared to a certain baseline. An example of this is shown in Fig. 3.8 for the trailing-edge flap deflections of an L-1011 aircraft with a baseline of zero flap deflection. From this figure, the optimal flap settings are clearly visible, and it can be observed that for increasing lift coefficients, increasing flap deflections are optimal.

The same type of plot based on interpolation of the aforementioned SmartX-Alpha data set is shown in Fig. 3.9. The lift-to-drag ratios are shown relative to a baseline of a constant zero actuator angle across the span. Although the same trend in increasing lift-to-drag ratio for higher-camber settings with increased lift coefficients can be observed, a large part of the operational lift coefficient range is dominated by a single setting. For lift coefficients below $0.85$, the maximum negative camber morphing setting yields the highest aerodynamic performance. This means that in terms of the trade-off between the angle-of-attack and camber morphing as means for lift production, a spanwise reduction in camber

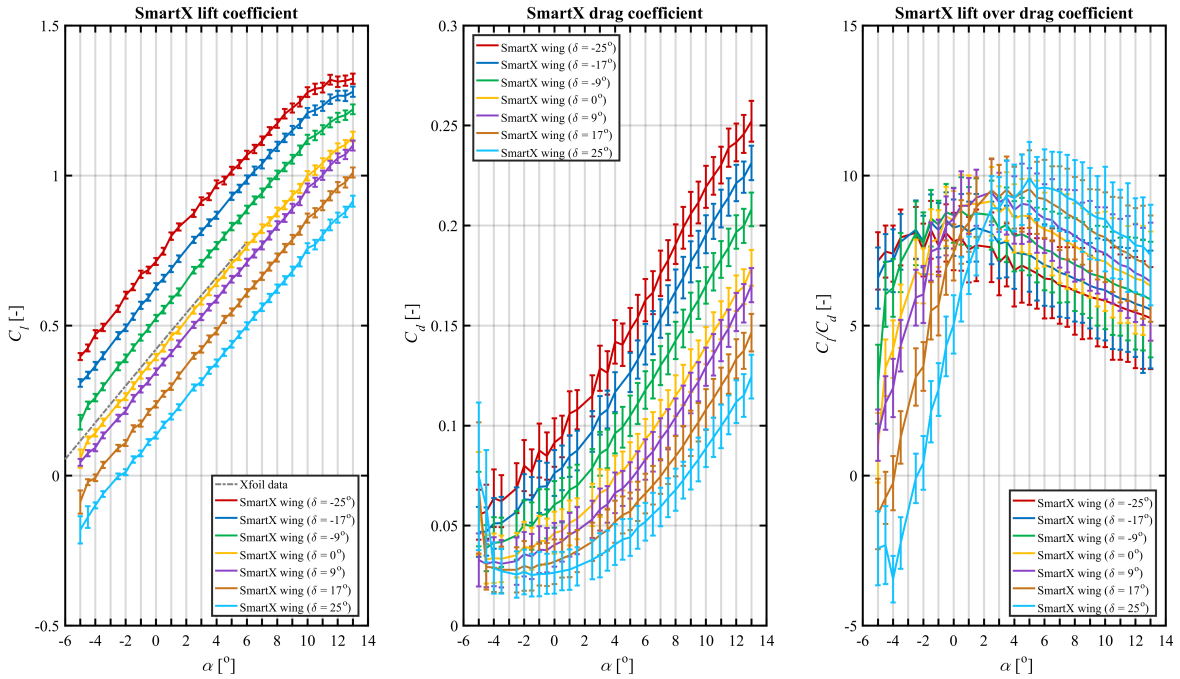



Figure 3.7: Measured lift and drag coefficients for angle-of-attack sweeps with various constant spanwise camber morphing settings from a previous wind tunnel campaign (August 2020) with SmartX-Alpha.

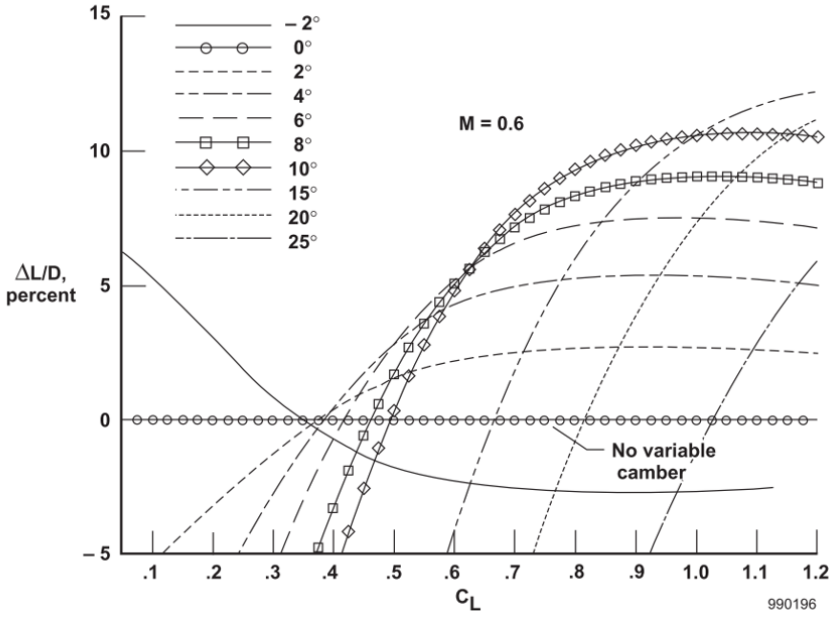



Figure 3.8: Percentage of performance benefit variation with trailing-edge deflections for an L-1011 aircraft [23].

and a corresponding increase in the angle of attack compared to the baseline is more efficient across the operational lift coefficient range. This can be explained by the fact that the NACA6510 airfoil already is highly cambered. However, if the maximum amount of negative camber morphing is used across the wing, no room is left for further spanwise camber reduction towards the wing tip, which is known to reduce the induced drag of rectangular wing platforms without a wash-out. Therefore, the online wing shape optimization procedure will have to trade-off the benefits of a close to ideal spanwise lift distribution and an overall amount of camber versus angle-of-attack balance which is efficient for the

target lift coefficient.

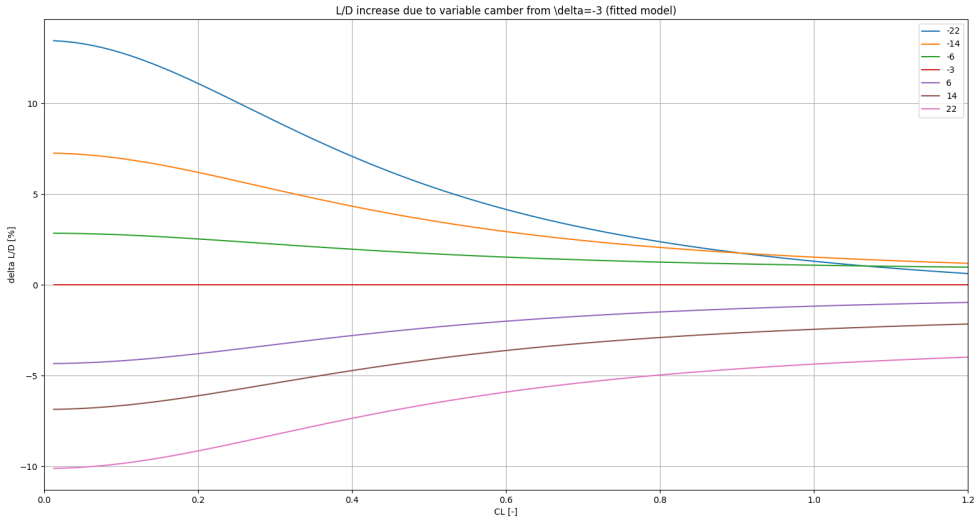



Figure 3.9: Percentage of performance benefit for various constant spanwise trailing-edge deflections for SmartX-Alpha, relative to the unmorphed base shape.

## 3.2. Optimizer

In this section the selection and configuration of the optimization procedure is described. Subsequently, the results from direct offline optimization of the previously described aerodynamic model with the selected optimization strategy are presented and discussed.

### 3.2.1. Optimizer requirements

The requirements for the optimization methodology, are derived from the main research question and are closely related to the requirements of the on-board model as the optimizer will need to operate on the on-board model. The following requirements were identified for the optimization method:

1. The optimization method should be a black-box optimization method.
2. The optimization method should be capable of global optimization.
3. The computational cost of the optimization should be realistic for real-time hardware implementation.
4. The optimization method should be able to handle constraints on the input and intermediate variables.

Black box optimization is defined as the “study of design and analysis of algorithms that assume the objective and/or constraint functions are given by blackboxes” [24]. Since the goal of the framework developed in this thesis is to perform online black-box aerodynamic performance optimization, the modeling method selected for the on-board model will be a black-box model. Hence, only input, output information will be available from the on-board model. Thus, the optimizer will have to maximize the objective function score using only function evaluations and their returned scores. Secondly, because the on-board model will be a global model, it is important that the optimization method is also able to overcome local minima such that the global optimum can be found on the on-board model. The third requirement is that the optimization should have low computational cost. This requirement is somewhat conflicting with the previous two and is therefore challenging. Finding the optimal inputs that correspond to the global optimum of a completely unknown function using only evaluations of that

function is challenging and will require many function evaluations and thus a significant computational time. However, in order for the application of the optimization framework in an online setting, it is of paramount importance that the computational time is required is as low as possible. Herein lies the most challenging aspect of online black-box shape optimization. However, because the goal is to optimize the steady-state lift-to-drag ratio, rather than the instantaneous lift-to-drag ratio during maneuvering, a relatively long computational time can be afforded. Since the dissipation of the transient aerodynamic responses to shape changes and subsequent averaging of noisy lift and drag force measurements may take upwards of one minute, an optimization time of 10 to 15 seconds is deemed to be acceptable. The fourth requirement is that the optimization method should be able to handle both constraints on the input and on the intermediate variables. Although the intermediate states are not of direct concern in the black-box strategy, some cannot be completely disregarded. Specifically, the actuator inputs to the morphing wing are to be limited to the achievable range of $\pm 25$ degrees. However, the system inputs that describe these are the virtual inputs. The actuator angles are determined by the combined effects of the virtual inputs, and thus their constraints can not be handled with rectangular bounds on the virtual input variables.

### 3.2.2. Optimizer selection

A categorized overview of black-box optimization (BBO) methods is shown in Fig. 3.10. Black-box optimization methods can be classified into the following four main categories: direct search methods, model-based methods, heuristic methods, and hybrid methods.

Direct search methods evaluate points around a current point to find the location with the lowest objective functions. Based on the objective function values, a new current point is selected for each iteration and a new set of surrounding points is computed. The set of points can be described by a multi-dimensional simplex, as is the case for the Nelder-Mead algorithm [25], or by a discretization in the primary axes such as in Coordinate Search (CS). The Generalized Pattern Search (GPS) is in operation similar to CS, however, the axes that define the mesh around the current point, also referred to as polling directions are flexible and it includes an additional search step that can help to overcome local minima [26]. The Mesh Adaptive Direct Search (MADS) [27] algorithm is a generalized extension of the GPS algorithm which allows an infinite set of polling directions. These direct search methods are local in nature [13]. Therefore, they do not satisfy the requirements set for the BBO method.

Model-based BBO methods use function evaluations to iteratively identify and optimize a black-box model of the objective function. Trust region BBO algorithms fit a usually smooth model around a current point, which is presumed to be accurate in a trusted sub-region around the current iterate. Derivative information is then used to find the next candidate point that lies within the trust region and improves the objective function score. Trust region methods too are local in nature [13]. Response Surface Methods (RSM) identify and iteratively refine a global model, such as s radial basis function [28], to approximate the black-box objective function. Bayesian Optimization (BO) similarly also retains a global model of the objective function. This method relies on Bayes' theorem to determine the best point to sample based on which points are either most likely to yield an improved value or for which points the model is most uncertain. This way BO is able to sample-efficiently trade-off between exploring and exploiting [29]. Projection-based methods identify a reduced-order model of the governing equations of a system. For models with known governing equations, many projection-based optimization techniques exist. However, for the optimization of black-box objective functions, this approach is challenging. One published example of projection-based BBO exists [30]. In this method, called UNIPOPT, the objective function values were projected on a subspace that was designed using a sum of the decision variables. However, no open-source python implementations of this algorithm exist as of yet.

Analytical methods are inherently infeasible for BBO as the objective functions are treated as black boxes. Unsurprisingly, many BBO algorithms are heuristic methods. Among the heuristic BBO methods are the sub-classes of stochastic and discrete methods, i.e., methods that do or do not involve random search steps. Hit-and-run, simulated annealing, particle swarm, and genetic algorithms all belong to the stochastic sub-class of heuristic BBO methods. Hit-and-run algorithms were independently proposed by [31] and [32]. In hit-and-run methods, randomly generations candidate solutions are compared to the current iterate, which is replaced if the candidate point represents an improvement over the current iterate. Candidate solutions are generated with a step into a direction sampled from a uniform distribution. In a generalized version by [33] the step size was also sampled from a uniform distribution and the algorithm was proven to converge onto a global optimum under mild conditions. The simulated

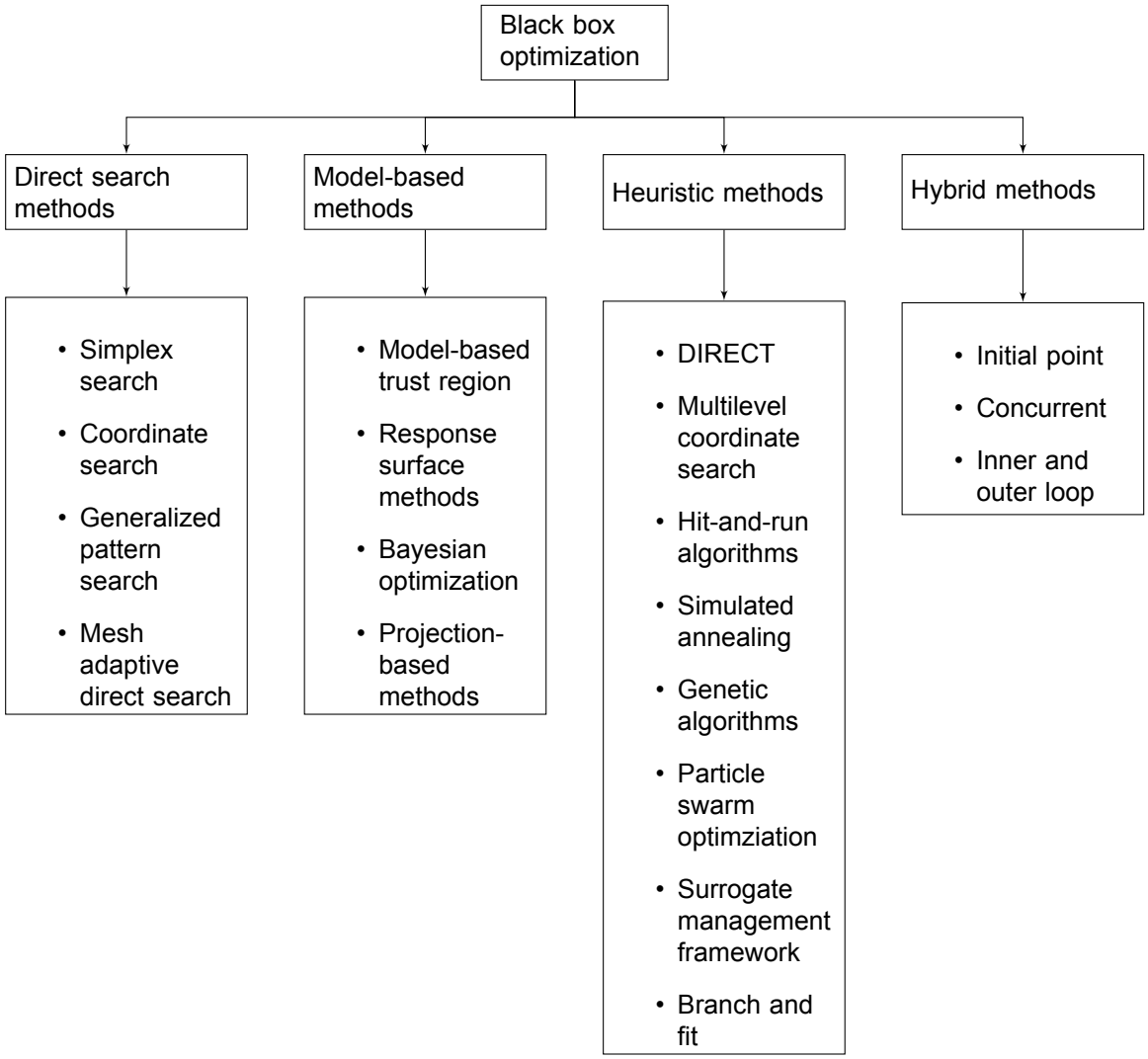



Figure 3.10: Overview of black-box optimization categories.

annealing algorithm was first proposed by [34] for combinatorial optimization problems and was later extended to continuous problems. The idea of simulated annealing is based on the analogy between optimization and metallurgic annealing process which involves the controlled heating and cooling of materials to alter their physical properties. In contrast to many other optimization methods, simulated annealing will with some probability accept a new incumbent solution whose objective function value is lower than the current incumbents. This allows the algorithm to make uphill steps and potentially escape local minima. The probability of acceptance of a candidate solution is dependent on a "temperature" parameter which is slowly decreased or "cooled" to zero. Despite the ability to overcome some local minima, it is not guaranteed that the global optimum will be obtained within a finite amount of iterations [35].

Genetic Algorithms (GA), also known as evolutionary algorithms, were first introduced by [36] and mimic natural evolution. Populations of candidate solutions are ranked and the best candidates are mutated and crossed to generate a new generation of candidate solutions. Analogous to the survival of the fittest principle in evolutionary biology, the highest-quality solutions and their offspring are selected for further consideration. GA optimization methods are not affected by discontinuities in the objective function, are good for multi-modal and high-dimensional problems. Developments in GA have introduced strategies that adapt the covariance matrix that is used to sample the new candidate solutions.

The Covariance Matrix Adaptation – Evolutionary Strategy (CMA–ES) proposed by [37], adapts the search distribution to the contours of the objective function using deterministic update rules and objective function value information from evaluated points. With a sufficiently large population size, CMA–ES has been shown to have good global convergence properties[38].

Particle Swarm Optimization (PSO) proposed in [39] is a population-based global global search algorithm, which is inspired by the cooperative behavior of large groups of animals like bees. During each iteration of the algorithm the objective function is evaluated on the locations of the particles. Apart from their spatial location property, the particles also have velocity attributes that determine how they will move through the search space. The particle velocities are calculated based on their previously velocities, the best solution position for the individual particles and the entire swarm, and acceleration coefficients.

The DIvide a hyper-RECTangle (DIRECT) proposed in [40], is a Lipschitzian-based BBO algorithm that uses estimates of the Lipschitz constant to identify potentially optimal hyper-rectangles and divide them until the algorithm converges. The objective function is evaluated at the center of each hyper-rectangle. These boxes are subdivided into thirds, once of which inherits the center of the previous hyper-rectangle for which the objective function is already known. Hyper-rectangles with low objective function values or steep objective function gradients are selected for dividing. DIRECT is guaranteed to converge onto the global optimum if the objective function is continuous in the vicinity of the global optimum [41]. This means that constraints for e.g. actuator saturation cannot be handled with objective function penalties that introduce discontinuities in the objective function. Furthermore, implementations of DIRECT are only available in Matlab and FORTRAN. Therefore, DIRECT can only be integrated with python using a FORTRAN compiler and a python-based wrapper.

The Multilevel Coordinate Search (MCS) algorithm proposed in [42] also recursively divides the optimization domain into hyper-rectangles with an evaluated base point like DIRECT. However, MCS allows this base point to be anywhere within these multi-dimensional boxes instead of only at the center. Each is assigned a level $s$, which is increased each time it is split. During every iteration, the boxes with the lowest objective function value are split. A global search is performed by splitting low-level boxes. The space within these hyperrectangles has not been thoroughly searched. The local search is conducted by determining promising search directions with a quadratic surrogate model which allows for fast convergence. The hyperrectangles are divided until their levels equal the maximum limit $s_{\max}$. MCS uses the $s_{\max}$ value to regulate the depth of the local search and thus the trade-off between searching locally and globally. As $s_{\max}$ approaches infinity, the evaluated points form a dense and resprepresentative subset of the optimization domain, and the algorithm converges onto a global minimum [42]. However, As $s_{\max}$ approaches infinity, so does the number of function evaluations required.

The surrogate Management Framework (SMF) method proposed in [26] integrates a pattern search method with a surrogate model. The relatively low-function-evaluation-cost surrogate model is used to accelerate and guide the pattern search method. The search step alternates between evaluating promising solutions and calibrating the surrogate model and balances local and global search.

The Stable Noisy Optimization by Branch and FIT (SNOBFIT) proposed in [43] is a heuristic method

which combines randomness and surrogate models. SNOBFIT fits quadratic models around the incumbent solutions, and fits linear models around the other evaluated points. The optimization of those models produces new candidate points for evaluation. When not enough evaluated points are available to fit the surrogate models, random points are generated. In addition points from relatively unexplored sub-regions of the domain are selected as candidates for evaluation. Soft constraints are implemented through a penalty-based approach. No theoretical guarantees exist for reaching the global optimum with a finite number of function evaluations.

Hybrid methods are methods that combine traditional BBO methods to overcome the limitations of the individual methods in terms of computational cost, convergence probability, or the number of function evaluations required. In general, there are three ways in which BBo methods are are combined [24]. The result from the first (global) algorithm is the initial point for the second (local) BBO procedure. Another way of hybrid BBO is the simultaneous operation of two algorithms that share incumbent solutions or function evaluation data. The third way in which BBO methods can be combined is where one algorithm controls the outer loop, and a second BBO algorithm is sequentially called in an inner loop.

The consensus in BBO literature is that no single ultimate BBO method exists [44, 45, 13]. Each of the available BBO methods have their restrictions, and which is the best method depends on the nature of the problem. The restrictions of individual methods may be countered with the integration of two complementary methods in a hybrid optimization framework.

For non-convex problems, CMA-ES was found to outperform state-of-the-art hit-and-run, simulated annealing, particle swarm, and other genetic algorithms [13]. CMA–ES is a very promising optimization method due to its proven effectiveness in finding global solutions on high-dimensional noisy and non-convex problems. It's largest drawback may be the high number of function evaluations required for optimization. Therefore, it may be useful to integrate this method with the use of a meta-model in order to reduce number of evaluations made on the real system. Based on the review of BBO literature it was decided to perform a practical implementation of both CMA–ES and BO to more precisely analyze their performance on the problem at hand. BO was chosen because of its sample efficiency, owed to the Gaussian process models which are used to trade-off search space exploration and exploitation.

Both CMA–ES and BO were tested on the aerodynamic model to investigate their effectiveness for wing shape optimization for active camber morphing wings as both these methods are capable of global black-box optimization and off-the-shelf python implementations for these methods are readily available. Bayesian optimization was performed using the implementation from the Scikit-learn package [46]. The CMA–ES optimization was performed using the pycma package [47]. It was found that both optimizers required many function evaluations. In combination with the 1 to 2 second evaluation time of the aerodynamic model, the computational time required for direct optimization was found to be unacceptably long for online optimization. CMA–ES was able to consistently converge onto well-performing wing shape and angle of attack combinations with an average of 2200 function evaluations. The average computational time was approximately 45 minutes. BO does not run until convergence but rather stops after a set number of iterations. With 1 hour of computational time, the solutions found by the BO algorithm were inferior to the inputs identified by CMA–ES. The wing shapes did not seem to make sense and their lift-to-drag ratios were lower than those of the CMA–ES solution for the same lift coefficients.

Therefore, it was decided to drop BO as an option for online optimization and continue with the CMA–ES optimizer. However, in order to overcome the long computational time required, it was decided to combine CMA–ES with a low evaluation cost global surrogate model in an inner-outer-loop hybrid approach. CMA-ES will evaluate populations of candidate solutions on the global model in the inner loop, and the global model will be improved using the latest system measurements in the outer loop. The online identified global model is referred to as the on-board model. The incorporation of a low evaluation cost global model greatly accelerates the speed of the CMA–ES optimization procedure, thereby making the framework viable for online use.

The solutions of the direct offline optimization on the aerodynamic model using CMA–ES are presented and analyzed in section 3.2.4. The selection and implementation of the on-board model is described in section 3.3. A more comprehensive description of the hybrid optimization framework and the fundamentals of the CMA–ES algorithm are presented in chapter 4 and 5 respectively.

### 3.2.3. Optimizer configuration

Describe selection of:

1. population size $\lambda$
2. initial standard deviation $\sigma_0$
3. initial population mean $\mathbf{x}_0$
4. scaling of the input variables $\mathbf{c}_\sigma$
5. objective function variation threshold $t_\mathrm{f}$
6. search space bounds $\mathbf{x}_\mathrm{min}, \mathbf{x}_\mathrm{max}$

The selection of the search space bounds for the optimization procedure is important because if the optimizer is allowed to go outside the region where the model can be trusted, it will exploit the inaccuracies of the model to find better, but unrepresentative solutions. In direct optimization of the aerodynamic model, search space bounds were only set for the angle of attack input because the aerodynamic model is only valid within $[-2.5^\circ \leq \alpha \leq 10.0^\circ]$. The virtual inputs were not as their magnitudes do not affect the validity of the aerodynamic model directly as long as the resulting wing shape is within the actuator limits. These morphing actuator limits however are handled within the cost function as the actuator angles are not model inputs, but intermediate states. In the online optimization using the on-board model, the search space bounds were selected as the bounds of the training data used because the accuracy of the neural networks outside the domain of their training dataset cannot be guaranteed. The specific bounds of the inputs are shown in the min and max window columns in Tab. 3.6.

The starting point $\mathbf{x}_0$ of was selected as the center of the input space, specifically $\mathbf{x}_0 = \frac{\mathbf{x}_\mathrm{min}+\mathbf{x}_\mathrm{max}}{2}$ such that optima throughout the domain can be reached with the fewest steps required.

It is recommended in the CMA-ES source code manual that when a global optimum is sought after, the initial step size is defined such that the expected location of the optimum is not far outside $\mathbf{x}_0 \pm 2\mathbf{c}_\sigma\sigma_0$ [48]. Since the global optimum is expected to be anywhere within the domain spanned by the training data set, the initial standard deviation and the scaling vector $\mathbf{c}_\sigma \in \mathbb{R}^6$ are selected such that $\mathbf{x}_0 \pm 2\mathbf{c}_\sigma\sigma_0$ spans the width of the domain in each input axis using Eq. (3.6).

$$\mathbf{c}_\sigma = \frac{1}{2\sigma_0}\left[\frac{\mathbf{x}_\mathrm{min}+\mathbf{x}_\mathrm{max}}{2} - \mathbf{x}_\mathrm{min}\right] \tag{3.6}$$

In the simulations using the nominal and surrogate models, the global step size was selected as $\sigma_0 = 3.125$ and the corresponding input scaling was selected as shown in Tab. 3.4.

Table 3.4: CMA-ES scaling of the input variables where the virtual inputs describe the morphing actuator deflections.

| input | scaling | $\sigma$ | unit |
|---|---|---|---|
| $\alpha$ | 1.000 | 3.125 | deg |
| $u_1$ | 2.679 | 8.371 | deg |
| $u_2$ | 2.793 | 8.728 | deg |
| $u_3$ | 0.773 | 2.415 | deg |
| $u_4$ | 0.536 | 1.676 | deg |
| $u_5$ | 0.415 | 1.298 | deg |

The objective function variation threshold $t_f$ is used as the convergence criterion by the optimizer. When the variation of the objective function, which is the cost function part of the framework, over all the candidate solutions in a generation is below this value, convergence is declared. It is important that the threshold is configured low enough such that the optimization procedure is not halted prematurely. At the same time, an unnecessarily low variation threshold will cause the optimizer to keep going for too long, thereby wasting precious computational time. In Fig. 3.11, the cost function value of the best performing candidate solution after convergence is plotted against variation in $t_f$. With some margin, the threshold for convergence was selected as $t_f = 1 \times 10^{-6}$.

Since the threshold is defined based on the variation of the objective function, this setting is cost function specific. If changes are made in the cost function, the convergence threshold should be reevaluated.

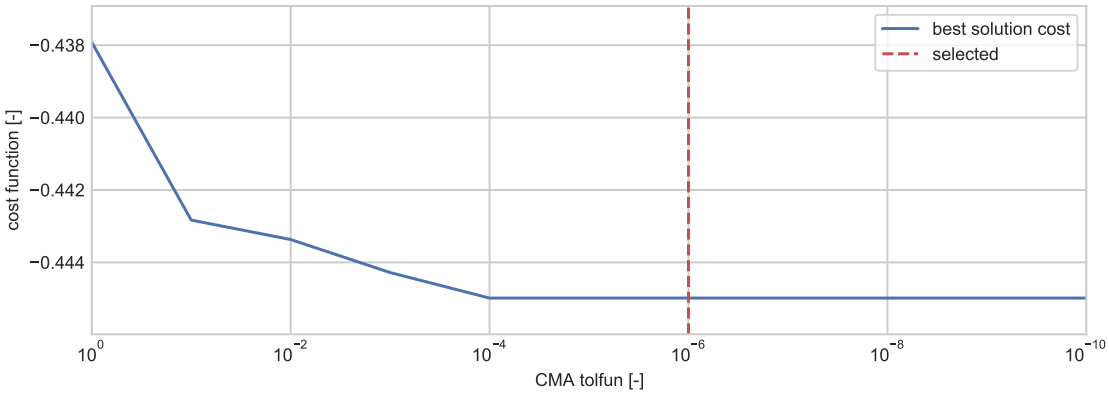



Figure 3.11: Variation of the cost function value of the best solution after convergence with different objective function thresholds.

The population size $\lambda$ is an important factor in the global convergence nature of CMA-ES [37]. Increasing $\lambda$ improves the global search capabilities and the robustness of the optimizer, however at the cost of reduced convergence speed. The convergence speed decreases at most linearly with the population size [49]. Therefore the population size was selected as large as could be reasonably afforded in terms of computational time. The population size was selected as $\lambda = 150$. Which corresponded to an average convergence time of $10.7$ seconds on a personal laptop (Intel® Core™ i7-4510U CPU, 8.00 GB RAM) and $7.5$ seconds on the OJF control room computer used during the wind tunnel experiments.

Because of the inherent randomness involved in the sampling of solution candidates from normal distributions, the optimization outputs can vary slightly between optimization runs. However, with large enough sample sizes the optimizer should repeatedly converge onto the same solution with very minor variations.

In order to verify the robustness of the optimizers' convergence, the optimization of the offline-trained on-board model using CMA-ES with the settings configured as presented thus far is repeated 100 times. The variations of the optimal inputs on which the optimizer converged are plotted in Fig. 3.12. The variation in the estimated optimal angle of attack is smaller than $0.004$ degrees. The variations in the virtual inputs that describe the local vertical trailing-edge displacements of the morphing wing are below $0.05$ mm. In conclusion, with the presented settings, the optimizer consistently converges on the same solution with minute variations that are lower than the precision with which these inputs can be actuated in the experiment.

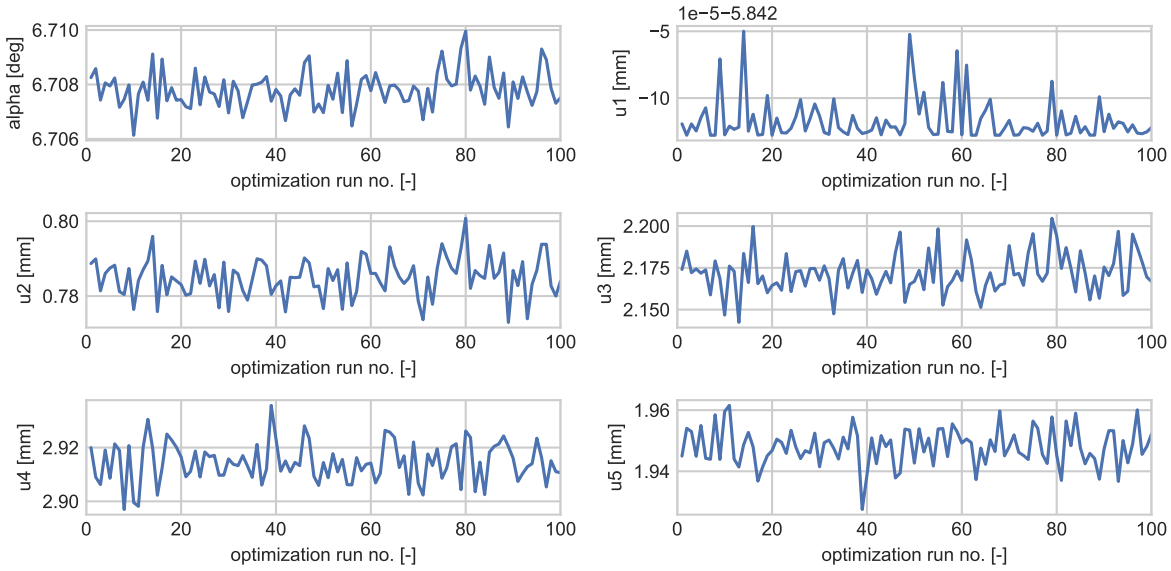



Figure 3.12: Optimal inputs repeatedly estimated with a population size of 150 for 100 optimization runs.

### 3.2.4. Offline optimization

By combining the optimizer and the aerodynamic model directly, Offline optimization can be performed. However, the large number of function evaluations needed for black-box optimization makes this process slow. The required number of function evaluations varies between 1500-3000 depending on the population size, scaling, and optimization landscape. As a result, the average time needed for direct shape optimization was 45 minutes. This makes direct shape optimization using CMA-ES infeasible for online application, especially considering the fact that measurement times for lift and drag coefficient estimations in any experimental set-up tend to be much longer than the evaluation times of the lower order aerodynamic model used in this study. Therefore, the online identified on-board model, which has a relatively low evaluation cost compared to either the wind tunnel or simulation experimental systems, will be used with the CMA-ES algorithm for online shape optimization. Ideally, the online shape optimization would be able to find the same optimal inputs, but in a much shorter time span. Nevertheless, the direct and offline way of optimization does provide insight into the shape the behavior of the optimized shapes. In the following section, the shape solutions of some of the online shapes are shown and investigated in more detail.

**Offline shape optimization shapes** Figure 3.13 depicts the wing shape solutions directly computed on the aerodynamic model offline for various target lift coefficients. The angles of attack and the virtual inputs corresponding to these offline optimal shape solutions are shown in Tab. 3.5. Each of the shape solutions comprises a decrease in the actuator angles, and by extension in the local camber towards the wingtip end. Additionally, in the $6^{\text{th}}$ morphing module, near the wing tip, all shape solutions command the lowest amount of camber achievable. This corresponds to the $11^{\text{th}}$ and $12^{\text{th}}$ actuators hitting the $-25$ deg limits. Most shapes show an approximately linear slope of actuator angle reduction for the center modules, with the exception of the solution shape corresponding to the highest target lift coefficient. The shape solution for $C_{\text{L}_\text{t}} = 1.20$ already commands the upper limit set for the angle of attack. In order to reach an even greater target lift coefficient of $C_{\text{L}_\text{t}} = 1.35$, the optimizer is forced to increase the wings camber even more. When servo 1 in module 1, which is nearest to the wing root reaches its upper deflection limit, the 'ideal' slope which is observed in the other shape solutions is compromised to allow an increased overall amount of camber on the wing. In this case, the spanwise decrease in local camber only starts at the fourth morphing actuator. In the case of the lowest target lift coefficient, the calculated optimal shape commands the highest amount of negative camber morphing almost across the entire wingspan even though the minimum angle of attack has not been reached yet, with the only exception occurring in module 1. Here too, the ideal slope of spanwise camber reduction is sacrificed for a more optimal overall amount of camber. Since the NACA6510 is already relatively highly cambered by default, it is perhaps unsurprising that the best course of action for achieving a small amount of lift is to decrease camber all along the wing.

It is thought that the main reason for the spanwise reduction in local camber observed towards the wingtips is the resulting reduction of the strength of the wingtip vortices, which in turn results in an improvement in the aerodynamic efficiency. These vortices are the result of the spanwise flow components caused by the "leaking" of high-pressure air from the bottom side of the wing around the wingtips towards the lower pressure regions on the upper wing surface [50]. It has been well established in the literature that optimal spanwise lift distribution in terms of the induced drag is the elliptical lift distribution. Even though the true spanwise lift distribution is unknown in flight, or even in the later wind tunnel experiments, the spanwise lift distribution can be calculated using the aerodynamic model in simulation.

Table 3.5: Offline optimized input parameters for various target lift coefficients

| $C_{\text{L}_\text{t}}[-]$ | $\alpha$[deg] | $u_1$[deg] | $u_2$[deg] | $u_3$[deg] | $u_4$[deg] | $u_5$[deg] |
|---|---|---|---|---|---|---|
| 0.25 | 1.10 | -24.48 | -0.96 | 0.62 | -0.28 | 0.07 |
| 0.50 | 3.57 | -21.18 | -5.50 | 1.47 | 0.81 | -0.68 |
| 0.75 | 5.86 | -16.40 | -11.14 | 1.21 | 1.84 | -0.49 |
| 1.00 | 8.32 | -12.22 | -15.93 | 0.84 | 2.47 | -0.05 |
| 1.20 | 10.00 | -7.02 | -21.67 | 0.04 | 3.07 | 0.85 |
| 1.35 | 10.00 | 3.42 | -30.05 | -6.58 | 5.31 | 3.65 |

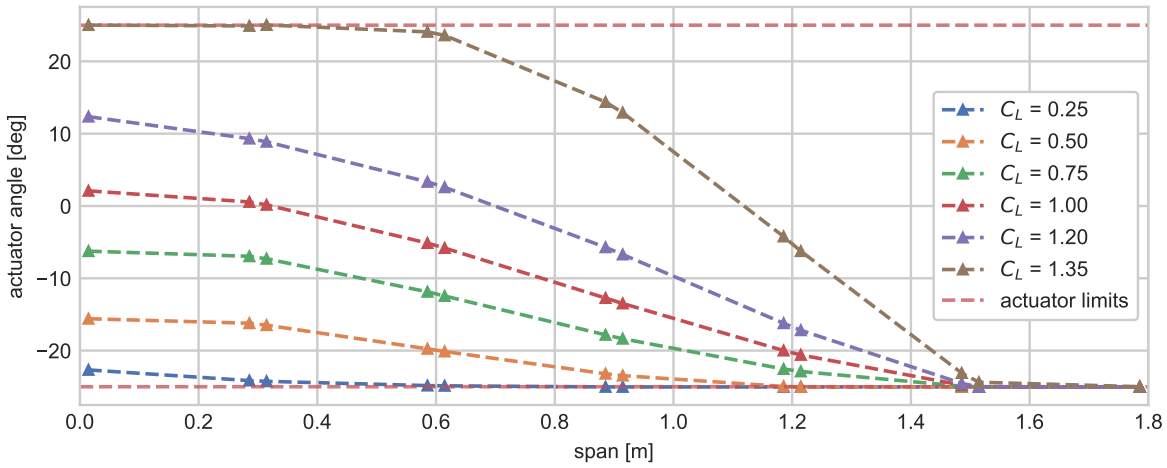



Figure 3.13: Offline optimized wing shapes of SmartX-Alpha for various lift coefficients.

**Spanwise lift distributions** In order to investigate the spanwise lift distributions of the offline-calculated optimal wing shapes and their corresponding optimal angles of attack, the lift distribution is reconstructed using the vortex strengths of the individual discrete panels that make up the wing geometry in the aerodynamic model. When the vortex strengths are known, the lift force per unit span $L'$ can be calculated using the Kutta-Joukowski theorem shown in Eq. (3.7). Here, $\Gamma'$ is the vortex circulation strength per unit span, and $\rho_\infty$ and $V_\infty$ are the freestream air density and flow velocity respectively.

$$L' = \rho_\infty V_\infty \Gamma' \tag{3.7}$$

$\Gamma'$ can be computed by summing the strengths of the discrete vortex panels $\Gamma_{i,j}$ in the chordwise direction at each of the 13 spanwise panel rows. When the resulting expression is divided by $\frac{1}{2}\rho_\infty V_\infty^2 S$, the 2d lift coefficients of these spanwise locations $C_{l_j}$ are obtained, as shown in Eq. (3.8).

$$C_{l_j} = \frac{\sum_{i=1}^{n_x} \Gamma_{i,j}}{\frac{1}{2} V_\infty S} \tag{3.8}$$

As shown in Eq. (3.9), the total 3D wing lift coefficient $C_L$ can be obtained through the summation of the discrete $C_{l_j}$ in the spanwise direction.

Using Eq. (3.8) the 2D lift coefficients of the unmorphed wing shape and an optimized wing shape are calculated. The optimized wing shape corresponds to the offline computed optimal wing shape for a target lift coefficient of 0.75. The lift distribution of the unmorphed, or jig, shape also corresponds to a lift coefficient of 0.75. Their spanwise lift distributions are shown in Fig. 3.14 together with the theoretically ideal elliptical distribution defined by Eq. 3.10, where $C_{l_0}$ is the 2D lift coefficient at the wing root and $y$ is the spanwise root coordinate measured from the wing root location. Based on the augmented aerodynamic model, the lift-to-drag ratio of the optimized wing shape is $4.6\%$ higher than the lift-drag-ratio of the jig shape at the same lift coefficient.

$$C_L = \sum_{j=1}^{n_y} C_{l_j} \tag{3.9}$$

$$C_l = C_{l_0} \sqrt{1 - \left(\frac{2y}{b}\right)^2} \tag{3.10}$$

The lift distribution of the optimized wing shape (green) is much closer to the theoretically ideal elliptical distribution (red) compared to the distribution of the jig shape (blue). The optimized wing shape entails negative camber morphing over the entire span of the wing, which corresponds to a reduction in the lift per unit span. Nevertheless, the 2d lift coefficient at the wing root is higher for this

shape than it is for the jig shape. This is because the angle of attack selected for optimal wing shape is approximately $2.4$ degrees higher than the jig shape's angle of attack at $C_{\mathrm{L}} = 0.75$. However, while closer, the distribution of the optimized wing shape does not perfectly coincide with the theoretical optimum. A further reduction of lift at the wing tip end would be beneficial. However, the morphing actuators near the tip are already at their maximum upwards deflections.

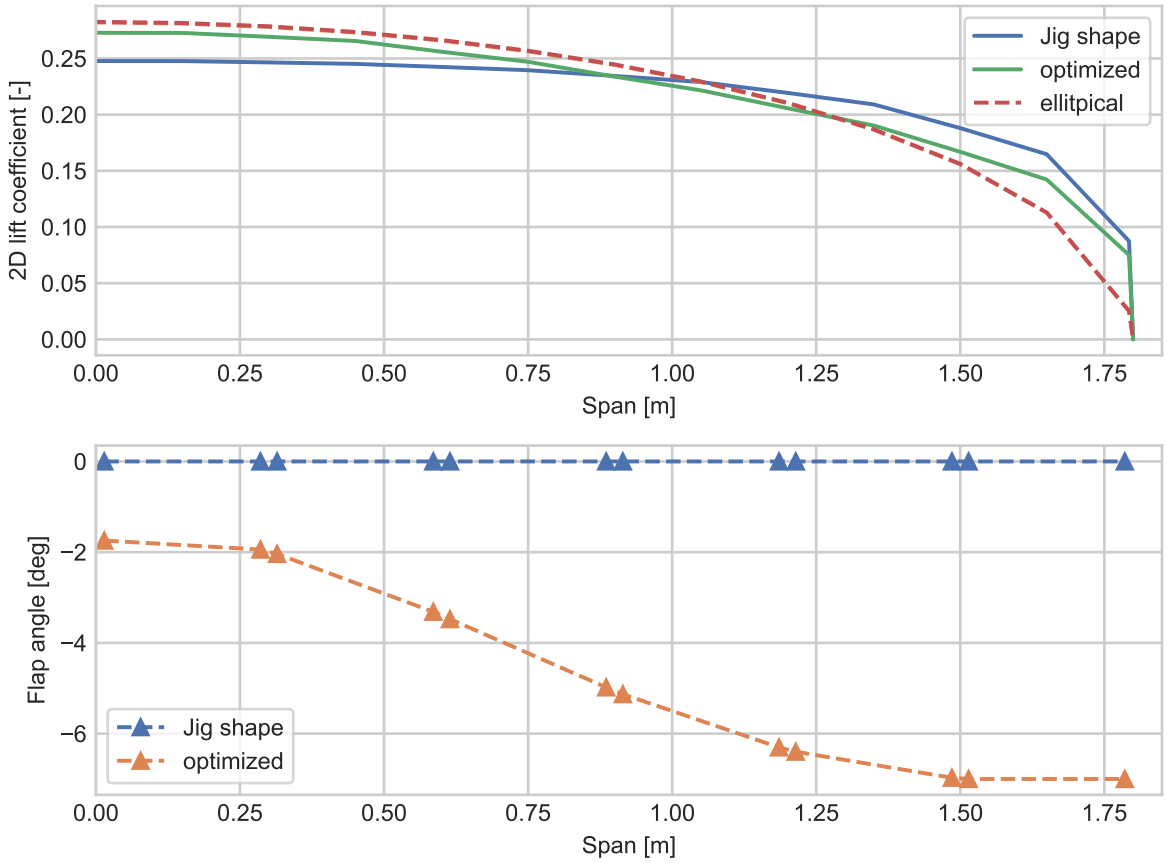



Figure 3.14: Offline optimized wing shape and corresponding changes in the spanwise lift distributions.

In order to investigate the effects of the optimized shape on the span and chordwise distributions of the lift, the discrete vortex circulation strengths are plotted at their corresponding panel locations. The resulting vortex strength surfaces of the jig (blue) and optimized (orange) shapes are shown in Fig. 3.15. Note that the planes that make up the surface plot do not correspond to the panels that make up the wing geometry in the aerodynamic model directly. Rather, the vertices of the plotted surfaces correspond to the locations of the panel one-quarter points. This is also the reason why the surfaces end slightly short of the physical edges of the wing. The spanwise trend of increased lift near the root end of the wing, and a reduction of lift near the tip end in the optimized distribution compared to the jig shape distribution is again apparent. However, in the chordwise direction, the differences in the two distributions are most significant on the trailing-edge side of the wing. This makes sense as the camber morphing only takes place at the quarter chord width section of the wing nearest to the trailing edge. However, the fact that the pressure distribution of the wing is mainly adapted on the trailing-edge side of the wing hints at another byproduct of this type of morphing. Specifically, if the pressure is increased or decreased spanwise only near the trailing edge, the effect on the wing pitching moment coefficient can be significant. In the case of the distribution plotted in Fig. 3.15, the optimized shape comprises a reduction of camber across the entire span of the wing. As a result, the pitching moment of the optimized wing is increased in the pitch-up direction compared to the jig shape. Consequently the pitching moment of the optimized wing is increased to $C_{\mathrm{m}} = -0.33$ from the jig shape's $C_{\mathrm{m}} = -0.38$, which constitutes a pitching moment coefficient magnitude reduction $12.9\%$.

In this thesis, only the isolated morphing wing is considered and the effects of morphing on the

moment coefficient are referred to as a byproduct. However, in future work, when the framework is applied to a complete aircraft, the moment coefficient effects can be exploited for aerodynamic efficiency improvement through the reduction trim drag from the horizontal tail plane.

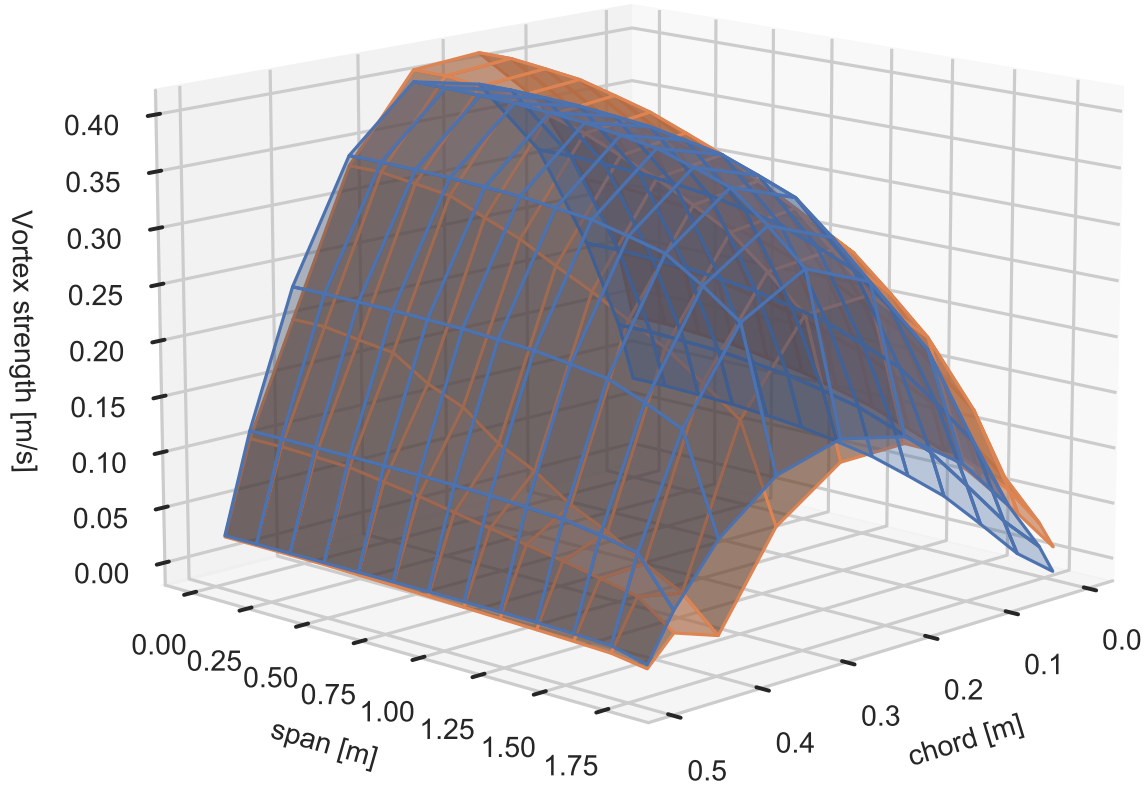



Figure 3.15: 3D surface plot of the panel vortex strengths for the jig shape (blue) and for an offline optimization solution (orange).

## 3.3. On-board model

### 3.3.1. Model requirements

The following requirements for the on-board model were identified:

1. The on-board model should be a black-box model.
2. The on-board model should be capable of identifying non-linear mappings.
3. The on-board model should be a global model.
4. The on-board model should be a relatively low evaluation cost.
5. The on-board model should be able to be (locally) improved online as new measurements become available.

First of all, since the research objective is to realize the most efficient wing shape on a morphing wing using a black-box online-learning-based optimization method, the on-board model should be a black-box model. Black-box models only seek to approximate an input-output relationship, rather than identify the internal states and mechanics of a system. Therefore, they are not limited by the degree of understanding of the fundamental principles at play. Additionally, this allows the model structure and solution method to be much simpler than those of complex white-box models [51]. As a result, the computational costs of black-box model identifications methods are lower for a complex system, than those of equivalent white-box models. The disadvantages of black-box models are that they are unable to make use of a priori knowledge of the system and that their parameters have no physical meaning as opposed to the parameters of analytical models and thus they provide little insight into the mechanics at play. This also limits the black-box models' applicability to the domains of the data on which they are identified, as their lack of knowledge of the true mechanics that cause the input-output relationship makes them unsuitable tools for extrapolation. However, when only a faithful interpolation of an input-output mapping is sought after, the black-box modeling approach is very effective [52].

Secondly, the method selected should be able to accurately model non-linear mappings. Since the relationships between the angle of attack and the virtual wing shape inputs, and the lift and drag coefficients are complex and non-linear, it is important that the on-board model is able to capture the nature and intricacies of these mappings.

Thirdly, in order to improve on the existing state-of-the-art methods for online wing shape optimization for morphing wings, which all use local models, the on-board model should identify a global model. The validity of the local models employed in the current state-of-the-art methods is limited to a region around a trim condition, which means that for changes in the flight conditions a new local model must be identified every time. The model identification maneuvers required for model identification generally involve sweeps of the morphing actuators and angle of attack, which cause bumpiness and flight path deviations and therefore severely limit the applicability of these methods on commercial aviation platforms. The need for these maneuvers could be eliminated with the identification of a global model.

Another requirement for the on-board model is that it should be able to be evaluated at a low computational cost. This requirement is important because the CMA-ES optimizer requires many function evaluations to converge. Hence, any amount of computational time is greatly magnified, and the total computational time of the optimization could quickly become unfeasibly long if a large number of function evaluations would need to be performed at high computational costs.

In order for the online optimization method to be adaptable, it is important that the model can be improved online based on new data. In terms of computational efficiency, it would be most ideal to not re-perform the entire modeling routine after each measurement, but rather to slightly adjust the model parameters accordingly. Each measurement acquired only corresponds to a single location in the multi-dimensional input space. Hence, each new observation sheds light on a particular part of the domain. It would then seem appropriate to locally refine the model instead of adapting it everywhere throughout the domain.

### 3.3.2. Model selection

For the independent in-flight operation of the framework, it is important that the on-board model is able to model the input-output relationship of the system under consideration with human supervision. Therefore, it is preferred that the modeling method does not require the tuning of, for example, the model order or structure online. Hence, the model should be able to capture any unknown nonlinear mapping with a fixed model structure.

It has been well established that multi-layer feed-forward neural networks are universal function approximators. In other words, with as few as one hidden layer, Artificial Neural Networks (ANNs) are capable of approximating any measurable function from one finite-dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available [53]. For this reason, it was decided to use ANNs as the basis of the global online-learning black-box on-board model.

ANNs can be classified into various sub-classes depending on the characteristics of their structures and the type of activation functions they use. An overview is given in Fig. 3.16. Feed-forward Neural Networks (FFNN) are networks in which data propagates only in the forward direction. They consist of an input and an output layer, and possibly several hidden layers in between. However, in feed-forward networks, the output signals of the neurons are only allowed to connect to neurons in subsequent layers, and not to any neurons in preceding layers. The main advantage of FFNN is that they can be

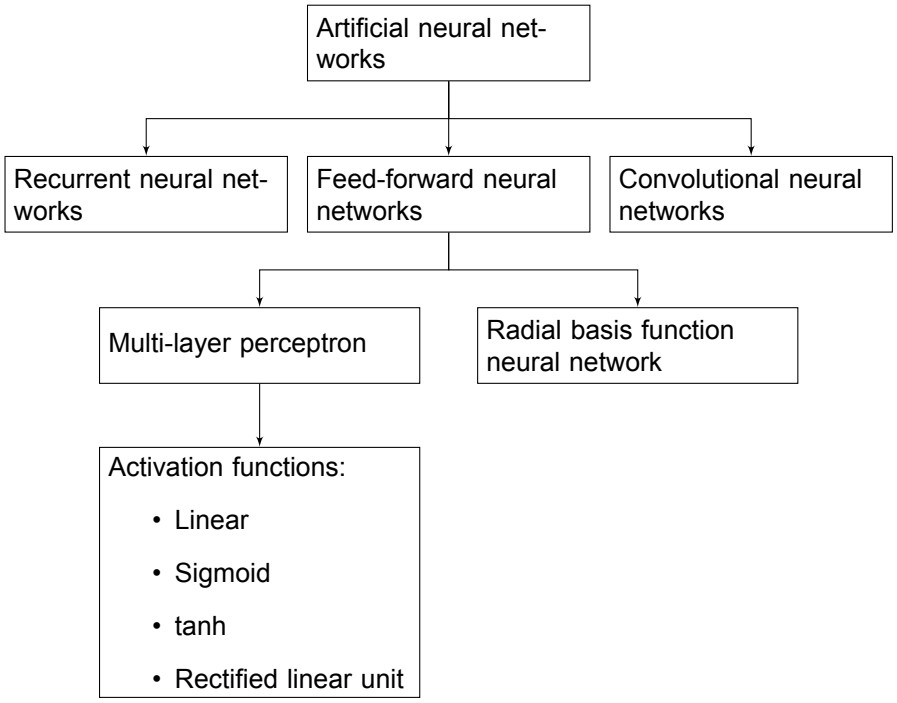



Figure 3.16: Overview of black-box optimization categories.

used on difficult and complex problems; however, their main disadvantage is that the training process normally takes a long time [54].

The output of Recurrent Neural Network (RNN) layers are connected back to their inputs. This helps to predict the new outcome of the layer based on the previous outcome. This so-called long short-term memory makes RNN especially effective for dealing with sequential data. For dynamic systems, the size of RNNs with the same approximation accuracy is is much more compact than for FFNNs. However, in practice, they are more difficult to be trained properly. Since in this research, the objective of the on-board model is to model the input-output mapping in steady-state trim conditions, the data is not temporally related. Hence, the benefits of RNN for the handling of sequential data are not needed for the on-board model.

Convolutional Neural Networks (CNNs) are comprised of convolutional layers preceding pooling and fully connected layers. In the convolutional layer collection of convolutional filters (kernels) is used for feature extraction. This allows CNN to independently learn the defining features without human supervision. Herein lies the main advantage of CNNs [55]. This makes CNNs very effective tools for image-based learning. However, since the numerical inputs to the on-board model are not image-based, CNNs offer no specific benefits over FFNN in the intended use case.

The multi-layer perceptron is a special case of feed-forward neural network in which all layers are fully connected. In fully connected networks, all neurons in one layer are connected to all neurons in the adjacent layers.

ANN can also be further classified based on their activation functions. Many different types of activation functions exist, and all have their advantages and disadvantages. The simplest type of basis function is the linear basis function. While it is the easiest to implement and works reasonably well on simple problems, the gradient of a linear activation function is independent of the input, which is a problem for error-back-propagation training. On the other hand, the sigmoid function is smooth and continuously differentiable. However, its output is not zero-centered which causes gradient updates to go too far in different directions. This makes optimization harder and leads to slow convergence. The hyperbolic tangent function improves on this limitation since it is zero-centered. Therefore, the tanh activation function is in practice preferred over the sigmoid function. The rectified linear function (ReLu) is linear for $x \geq 0$ and is zero everywhere else. This means that a large portion of the neurons return zero values, in other words only few neurons are activated and thus the network is sparse. This has been found to accelerate the convergence of stochastic gradient descent training compared to the

sigmoid and tanh functions.

$$\phi_j(x) = \exp\left(-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right) \tag{3.11}$$

Another type of feed-forward neural network which is characterized by its activation function is the Radial Basis Function neural Network (RBFN). The Radial Basis Function (RBF) shown in Eq. (3.11), is unique in the fact that it comprises a location in space. The output of each RBF neuron depends on the distance between the input point $x-$ and the RBF center location $\mu_j$. Therefore RBF neurons are locally sensitive. RBFNs have good generalization, strong tolerance to input noise, and online learning ability [54]. These properties make RBFNs very effective tools for designing flexible control systems. RBFN are recommended for function approximation problems of surfaces with regular peaks and valleys [56]. Additionally, RBFNs are more robust to noise in the input data than other types of multi-layer feed-forward neural network types [56]. RBFNs are also particularly suitable for multi-parameter and scattered data [57]. The data gathered in-flight is to be expected to be both heavily scattered and noisy. Combined with the fact that RBFNs are locally sensitive and can thus more easily locally adapt to new data, it is concluded that they are the preferred neural network type for the use-case of the on-board model.

### 3.3.3. Offline training set

For the offline training of the on-board model, a training data set was created using the nominal aerodynamic model. The network weight produced by the offline training procedure would be used as the initial weights, and hence serve as an initial guess, for the on-board model in later experiments. The training data set consisted of angle-of-attack and virtual inputs and their lift and drag coefficients as evaluated on the nominal aerodynamic model. The evaluated inputs were derived from a six-dimensional grid of point which were equally-spaced within each dimension. The defining parameters of the offline training data set grid are shown in tab. 3.6.

Table 3.6: Overview of the input bounds of the offline optimization solutions, and the selected bounds and mesh for the offline training data set.

| Parameter | Solution min | Solution max | Min window | Max window | Window size | Points | Step |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 1.096 | 10.000 | -2.500 | 10.000 | 12.500 | 9 | 1.563 |
| $u_1$ | -24.478 | 3.425 | -27.268 | 6.215 | 33.483 | 11 | 3.348 |
| $u_2$ | -30.050 | -0.956 | -32.959 | 1.954 | 34.913 | 11 | 3.491 |
| $u_3$ | -6.576 | 1.475 | -7.382 | 2.280 | 9.662 | 8 | 1.380 |
| $u_4$ | -0.276 | 5.312 | -0.835 | 5.871 | 6.705 | 6 | 1.341 |
| $u_5$ | -0.681 | 3.646 | -1.114 | 4.079 | 5.193 | 5 | 1.298 |

In order to improve the representative quality of the data set, the data set domain was focused on the sub-region of the domain where the optimal solutions were expected to reside. This represents a trade-off of increased spacial sample-density at the cost of domain width. This sub-region was based on the solutions obtained through direct offline CMA–ES optimization of the nominal aerodynamic model described in sec. 3.2.4. The bounds of the offline training data set were selected to be $10\%$ wider that this sub-region to provide a margin. With the number of equally space points per dimension as presented in tab. 3.6, the total size of the training data set was 261360 data points. The evaluation of these data points took approximately 90 hours of computational time.

### 3.3.4. Hyperparameter optimization

Hyperparameter Optimization (HPO), the optimization of neural network hyperparameters, has historically been challenging. The training of neural networks is complex, and the (combined) effects of adjusted hyperparameter settings cannot be analytically expressed. In other words, no shortcuts in the form of convenient analytical expressions of the ideal hyperparameter value exist. Therefore, the quality of a given set of hyperparameters can only be evaluated by training the network on the intended data set. Hence, HPO represents a black-box optimization problem.

The simplest, yet common-practice approach is tuning of the hyperparameters through simple trial and error. This method, however, heavily relies on the skill and experience of the engineer who performs it and offers little confidence in the eventual results. To avoid the time-consuming and unreproducible process of trial-and-error HPO, automatic HPO Algorithms have been proposed. Since the training of neural networks is time-consuming, it is very important that the black-box optimization algorithms which are to be used for HPO are sample-efficient. The time required for training is heavily dependent on the size of the training data set and the number of training epochs. However, even with a small data set and a moderate number of epochs, the time required per training run was around 2 minutes. While this is much shorter than the 23 hours required for the offline training with the complete data set for 2000 epochs, many function evaluations are required before a good set of hyperparameters may be found. Naturally, any optimization method that requires many function evaluations is considered infeasible for HPO, e.g. Random, or grid search methods, or evolutionary optimization methods like CMA–ES may simply require more than can be afforded within the time constraints of this thesis project.

Because of its sample-efficient black-box nature, Bayesian Optimization (BO) was selected for hyperparameter optimization. BO uses Gaussian process models of the objective function to trade-off exploration of regions with high uncertainty against exploitation of model knowledge by evaluating points with the highest likelihood of being optimal in a sample-efficient way. In order to reduce the time needed for this hyperparameter procedure, only $10\%$ of the offline training data set was used. This still amounted to 26136 samples. While the true optimal hyperparameter settings are data set specific, it is expected that a network architecture that is optimized for a representative subset of the data set may not represent the ultimate optimum, but a good configuration nonetheless. Even if the hyperparameter optimization were conducted on the entirety of the nominal aerodynamic model data set, the resulting architecture would still be sub-optimal for the interpolation of the data set which will be gathered in the wind tunnel experiment. Hence, the quest for the ultimate hyperparameter settings may not only be a virtually impossible effort but a futile one as well. Therefore, the aim of this procedure is to find sufficiently well-working hyperparameters which may also be expected to work well for similar data sets.

The BO hyperparameter optimization procedure was performed on training on the aforementioned reduced data set for 100 epochs per function evaluation call of the BO algorithm. In total, the algorithm was allowed 200 evaluations to produce an estimate of the best hyperparameters. The function call limit was selected in accordance with the time available. A decreasing trend was observed in the speed of the BO algorithm during optimization.

The BO algorithm was allowed to tune the number of neurons, the initial learning rate, the mini-batch size used and the adaptive learning rate optimization algorithm used. These were Adagrad [58], Adadelta [59], Adam [60].

The HPO search space was configured as:

- number of neurons, integer $\in [1, 1000]$
- initial learning rate, float $\in [10^{-4}, 10^{0}]$
- mini-batch size, integer $\in [2, 64]$
- optimizer, integer, [1=Adagrad,2=Adadelta,3=Adam]

**BOHB** After the initial HPO with BO, a more advanced HPO method was also tried after the simulation experiments. The Bayesian Optimization – HyperBand (BOHB) proposed in [61] was implemented. The proposed HPO algorithm combines the best features of BO and HyperBand (HB) and has been demonstrated to outperform both BO and HB on a wide range of problems. The algorithm was configured on the following search space.

- number of neurons, integer $\in [16, 1024]$
- initial learning rate, float $\in [10^{-5}, 10^{0}]$
- mini-batch size, categorical $[4, 8, 16, 32, 64]$
- optimizer, categorical, [Adagrad,Adadelta,Adam]

Unfortunately, due to the long computational time required, this procedure was unable to produce results before the start of the wind tunnel experiments. However, this HPO strategy is promising and may be used for improved hyperparameter optimization in future work.

### 3.3.5. Model parameter sweeps

The prediction accuracies of the offline and online-learned neural net models in different parts of the six-dimensional input space are investigated by means of parameter sweeps. The offline-trained model was trained for 2000 epochs on the offline NACA6510 training set (261 360 samples) which was derived from the nominal aerodynamic model, as described in section 3.3.3. The online-trained model was trained over 100 iterations of pseudo-random exploration on the surrogate aerodynamic model. During the parameter sweeps, one of the six inputs was varied while the others remained fixed and both the neural network on-board models as well as the nominal and surrogate aerodynamic models were evaluated at these points.

**Center of the input space** During the first sweep series, the input parameters which were not varied were fixed at the center of their allowable ranges. This is the part of the domain in which the evaluated data points are approximately equally surrounded by training data on all sides. The lift and drag coefficients resulting from these parameter sweeps are shown in Figs. 3.17 through 3.22. The y-axes of the two lift coefficient and the two drag coefficient plots have been given the same length such that the differences in prediction accuracy can be clearly viewed. From these results, two distinct trends emerge. First of all, the estimated outputs of the offline trained on-board model (orange) are close to the outputs computed on the nominal aerodynamic model (blue). Although not as close as the offline trained model and the nominal aerodynamic model, the online-trained neural network model (red) which was allowed only a fraction of the training time and training set that were available for offline training is still relatively close to the surrogate aerodynamic model (green). Secondly, most notably for the online-trained model, the most accurate predictions are observed for the sweeps of the angle-of-attack parameter which has the strongest influence on the lift and drag of the wing, with a decreasing accuracy trend with the higher-order virtual shape inputs.

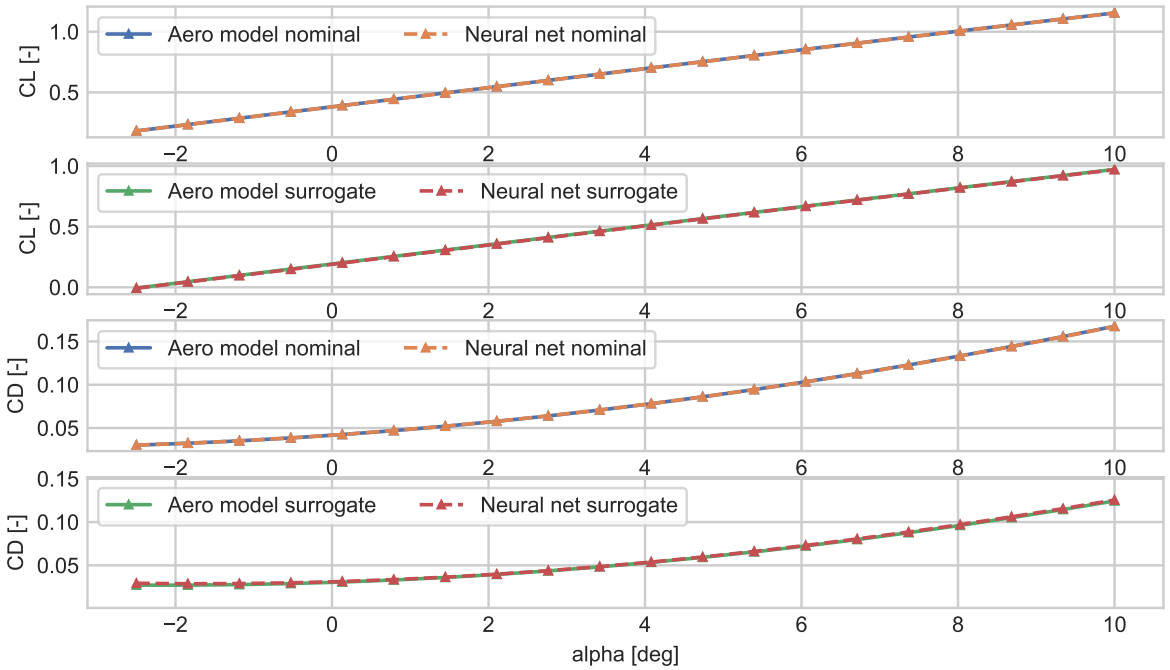



Figure 3.17: Parameter sweep of the angle of attack parameter with other parameters fixed at the center of their ranges, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

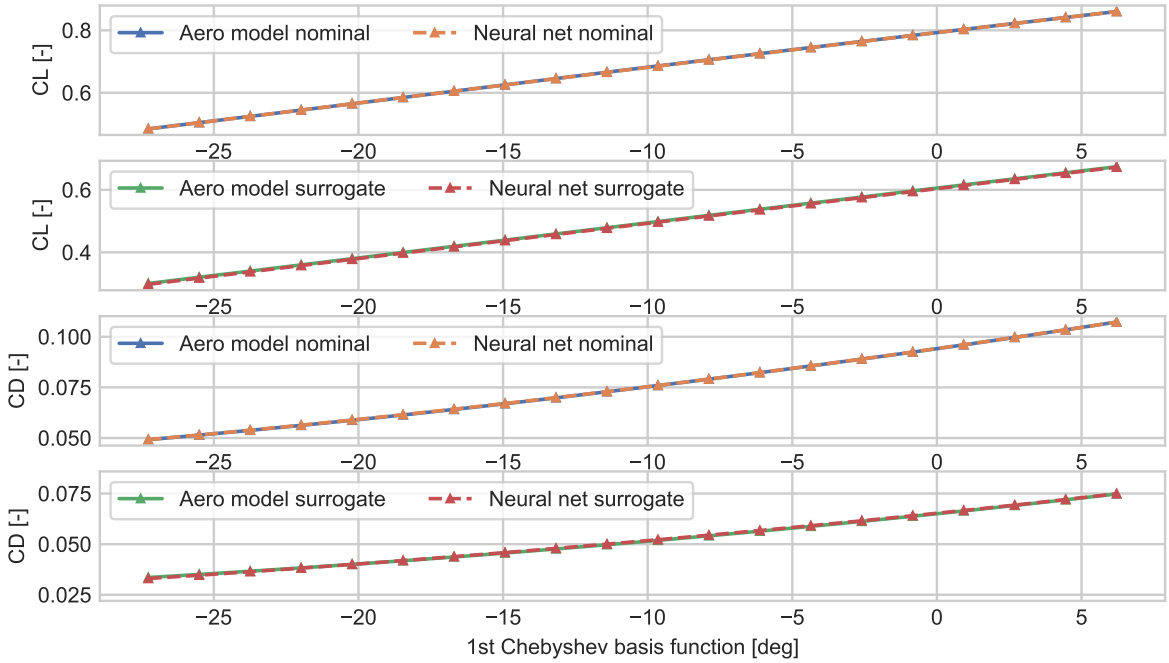



Figure 3.18: Parameter sweep of the first virtual input parameter with other parameters fixed at the center of their ranges, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

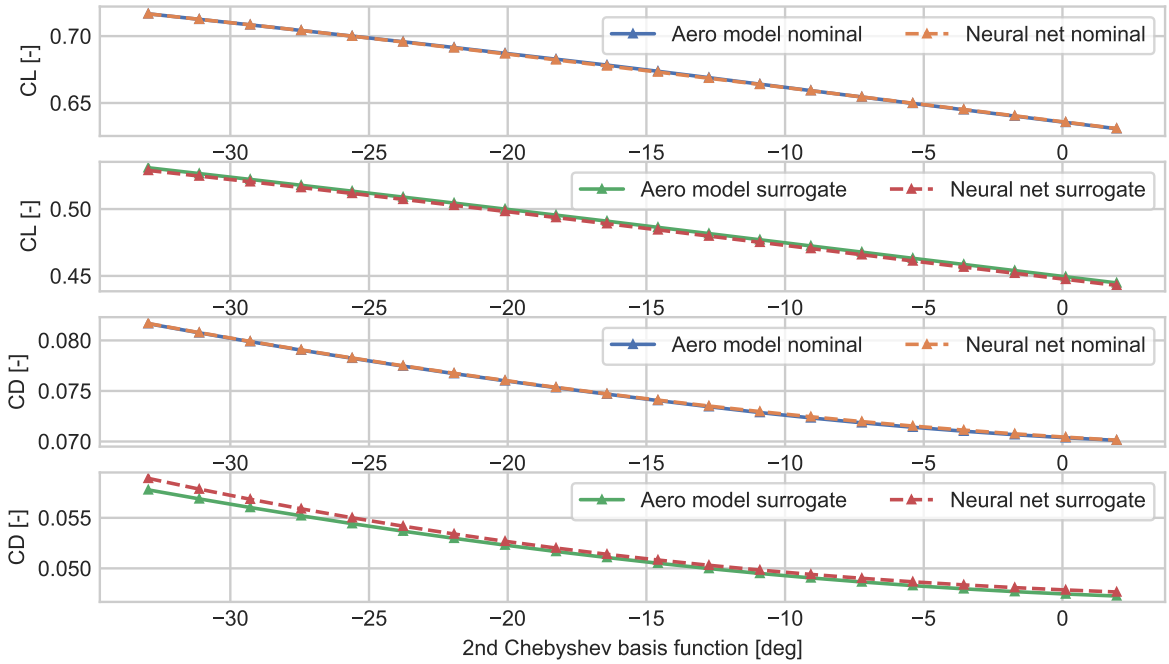



Figure 3.19: Parameter sweep of the second virtual input parameter with other parameters fixed at the center of their ranges, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

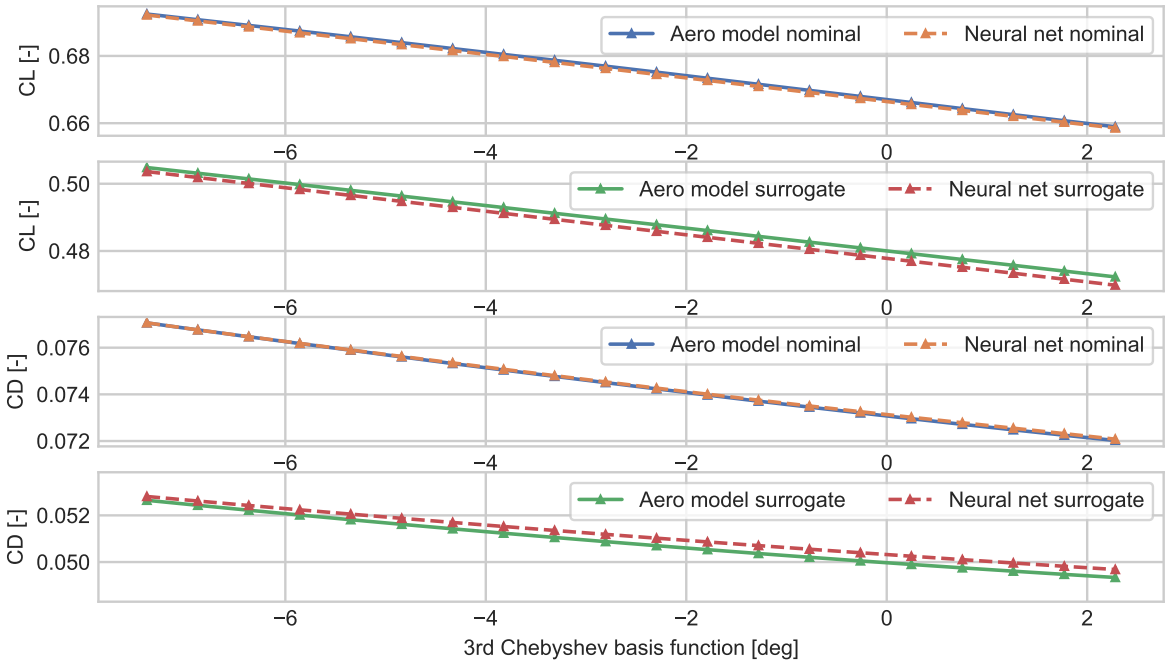



Figure 3.20: Parameter sweep of the third virtual input parameter with other parameters fixed at the center of their ranges, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

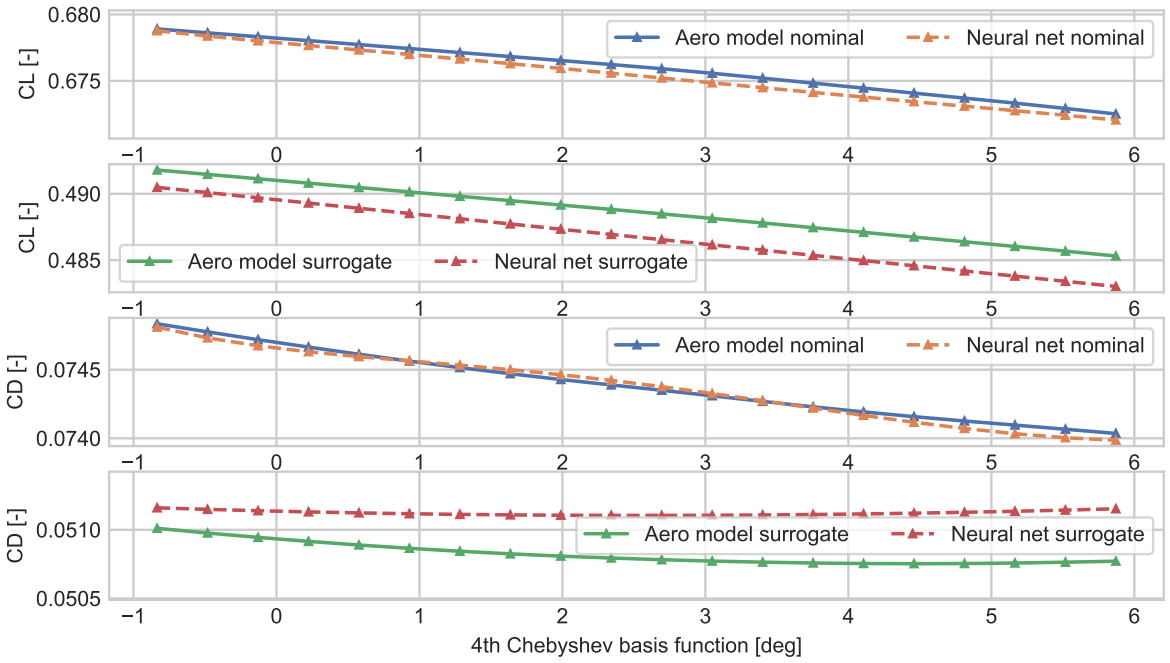



Figure 3.21: Parameter sweep of the fourth virtual input parameter with other parameters fixed at the center of their ranges, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

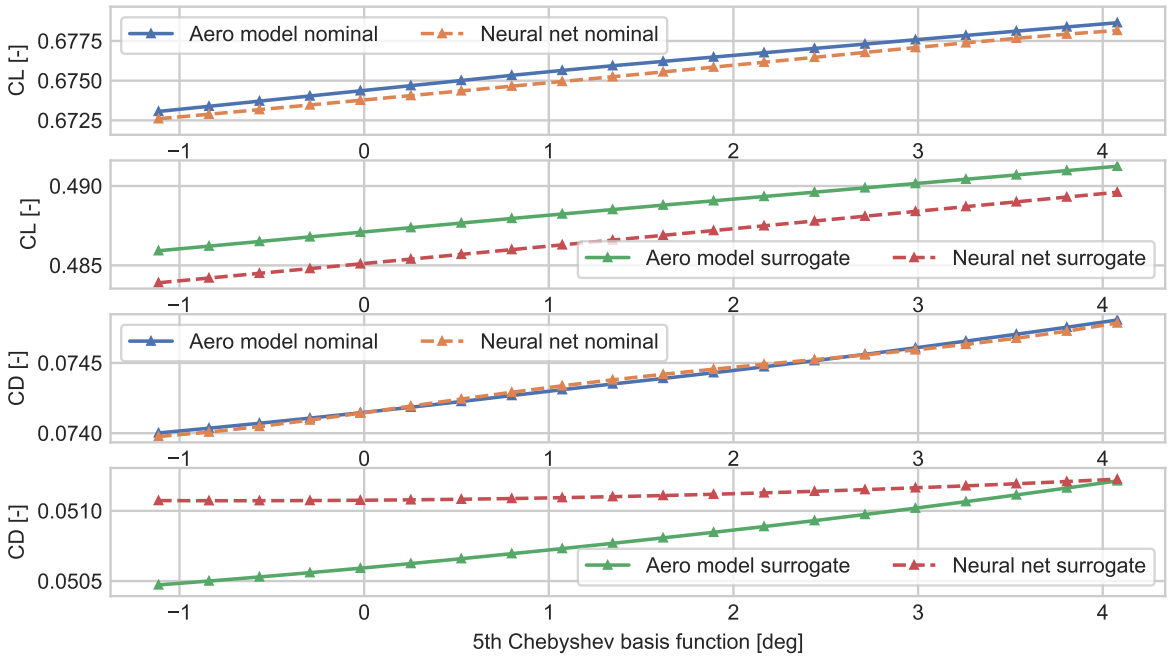



Figure 3.22: Parameter sweep of the fifth virtual input parameter with other parameters fixed at the center of their ranges, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

**Outside the interpolation domain** During the second sweep series, the input parameters which were not varied were fixed at $5\%$ over their upper limits. In this part of the domain, the evaluated data points are not surrounded by training data points on all sides. These evaluated points are not within the bounds of the training data, and hence, the model is not strictly interpolating, but extrapolating the information from the training data set. The number of nearby data points is smaller and the amount of data points is unequal in different directions. The lift and drag coefficients resulting from these parameter sweeps are shown in Figs. 3.23 through 3.28. Compared to the parameter sweeps at the center of the domain, an increase in model prediction error can be observed. This is most apparent for the higher order virtual inputs. Not only are the prediction errors of the surrogate model larger for the online-trained neural network in this instance, even the offline-trained model starts to have a larger prediction error in this case. This shows that the prediction accuracy of the neural network models decreases outside the edges of the training set domain. Therefore, it is important that the on-board model is not used to predict the aerodynamic coefficients of the wing shape and angle of attack inputs outside the training data set domain. It is for this reason that the bounds of the training data set were also used as the search space bounds for the optimizer.

## 3.4. Simulation procedure

In this section, the simulation procedure is explained. First, a description of the workflow of the simulation algorithm is given. Then, the two operational modes, or phases, used in the experiment are further elaborated on.

### 3.4.1. Algorithm description

A flow diagram of the algorithm used in the simulation experiment is shown in Fig. 3.29. On start-up, the relevant modules are loaded in and the configurable parameters are read. Then, a priori data is loaded in. These data consist of the neural network parameters, the SmartX-Alpha actuator locations, and noise data. Subsequently, the non-configurable parameters and objects such as the aerodynamic model and buffers are initialized. At the start of each new iteration, the operational mode is determined based on whether the current iteration number exceeds the set number of exploration iterations. Only two operational modes exist, these are “wandering” and “optimization”. If the current iteration number is smaller than the number of exploration (wandering) iterations to perform, the new iteration will be

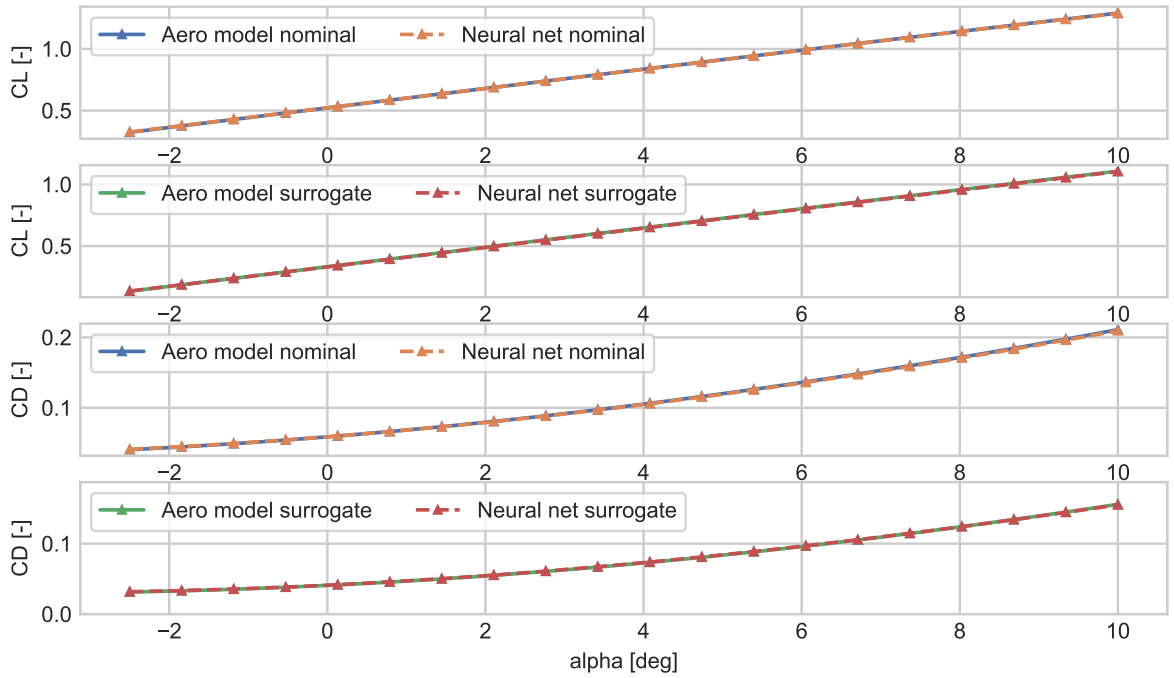



Figure 3.23: Parameter sweep of the angle of attack parameter with other parameters fixed at 5% over their upper limits, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

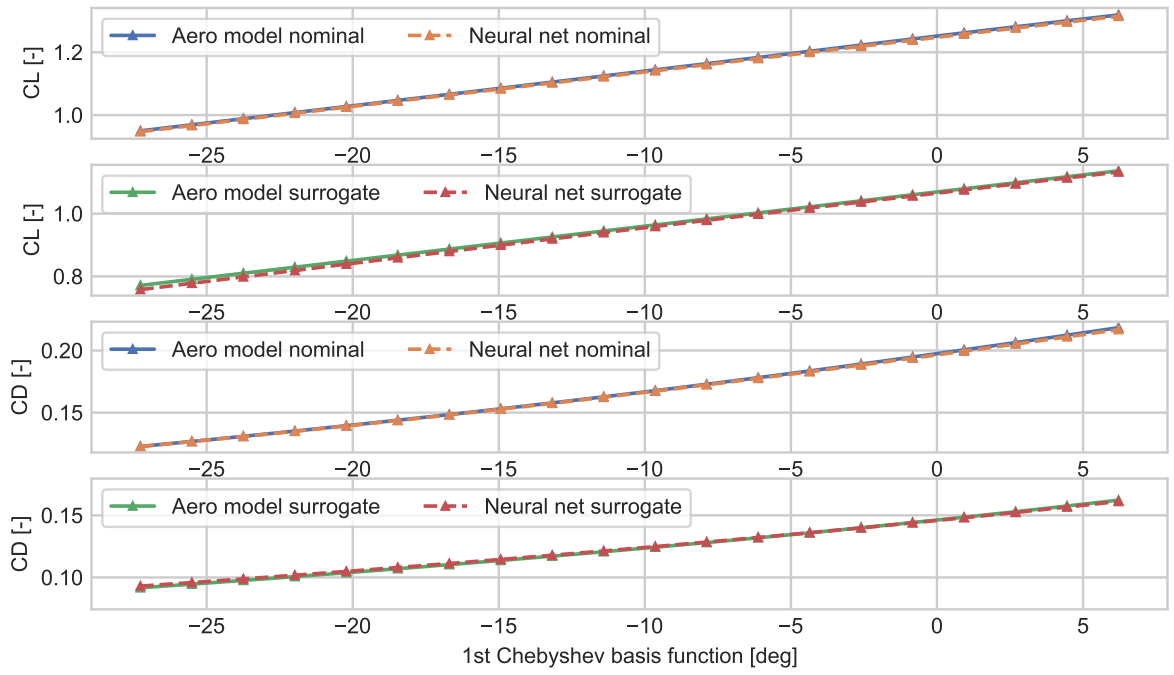



Figure 3.24: Parameter sweep of the first virtual input parameter with other parameters fixed at 5% over their upper limits, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

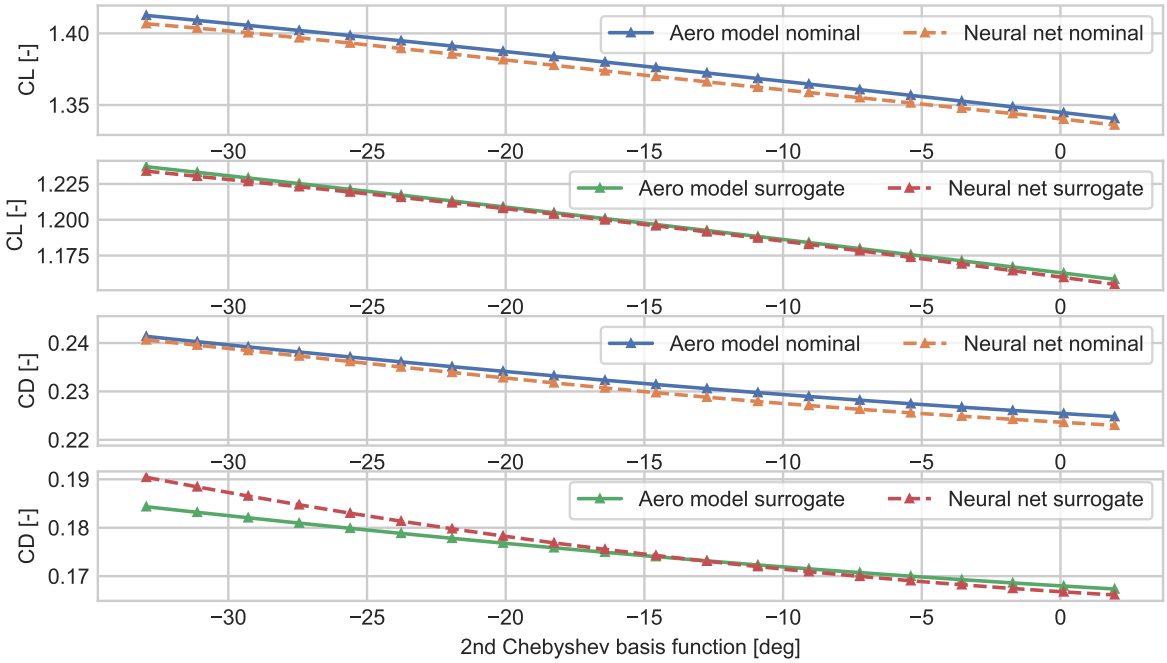



Figure 3.25: Parameter sweep of the second virtual input parameter with other parameters fixed at 5% over their upper limits, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

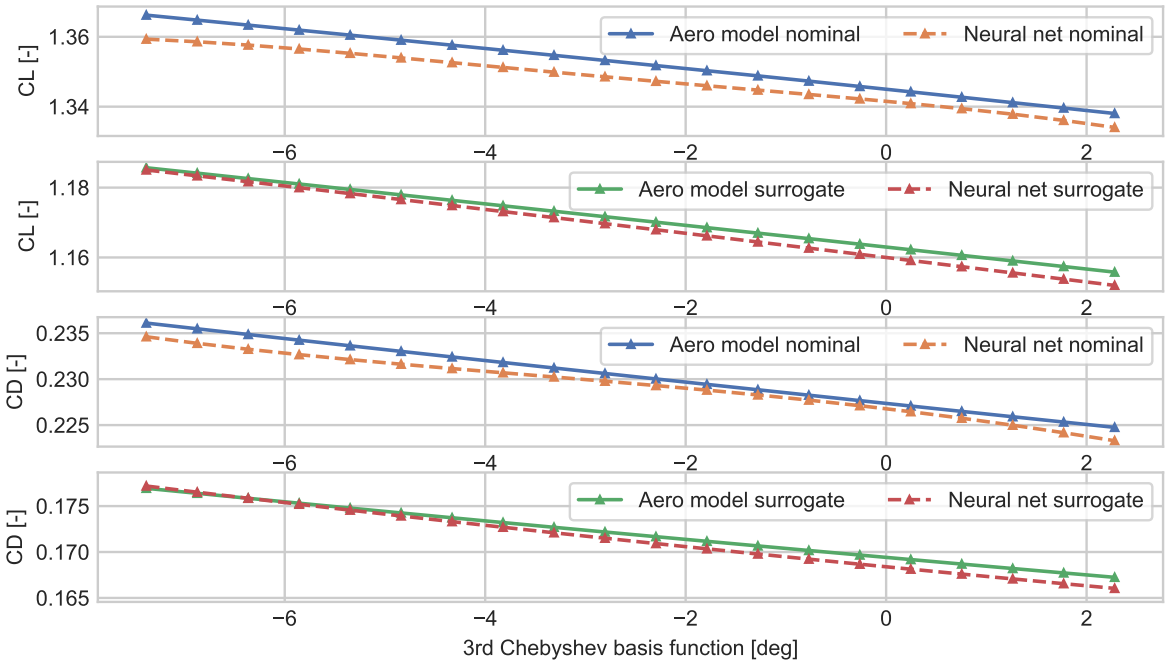



Figure 3.26: Parameter sweep of the third virtual input parameter with other parameters fixed at 5% over their upper limits, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

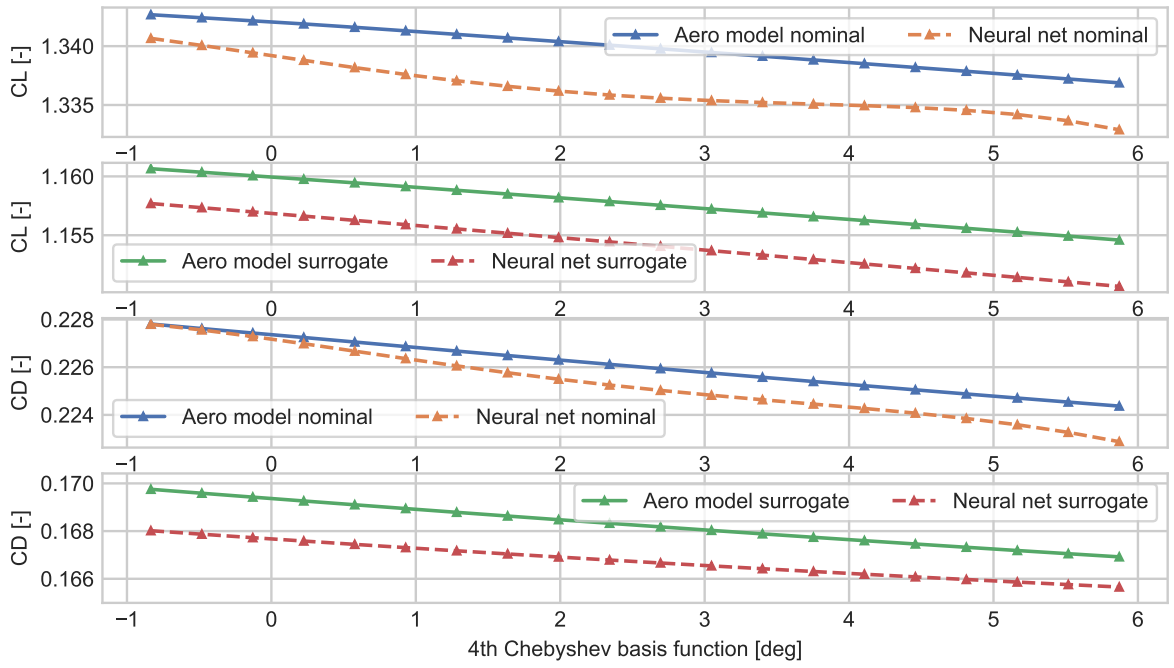



Figure 3.27: Parameter sweep of the fourth virtual input parameter with other parameters fixed at 5% over their upper limits, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

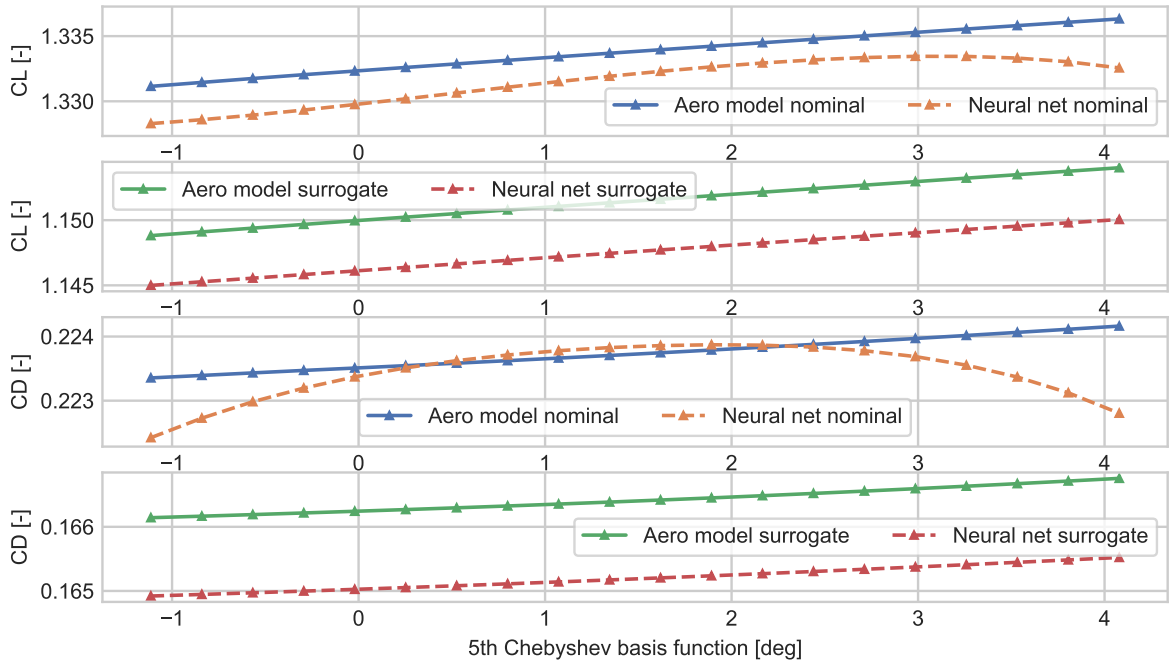



Figure 3.28: Parameter sweep of the fifth virtual input parameter with other parameters fixed at 5% over their upper limits, for the nominal and surrogate aerodynamic models and their corresponding learned neural network representations.

performed in wandering mode. A pseudo-random draw from a six-dimensional Sobol sequence is performed and the results are re-scaled onto the input variable axes. If at this point, any of these inputs violate either the angle-of-attack of actuator constraints a whole new pseudo-random set of inputs is drawn. When a satisfactory set of inputs is obtained the target lift coefficient for the current iterate is taken from a preset schedule. Next actuator saturation is simulated by capping any inputs that exceed their limits at their nearest bounds.

The resulting input is then actuated on the aerodynamic model and the corresponding lift and drag coefficients are obtained. The measurements of these coefficients are simulated by imposing a noise signal before averaging the result. Subsequently, the newest measurements and the corresponding input are added to the memory buffer (training data set). If the buffer has reached its maximum size, a nearest-neighbor search is performed and the best-connected sample data point in the buffer is replaced. Then, the data in the buffer is used to train the neural networks that make up the on-board model. Finally, the adapted on-board model is used by the optimizer to determine the most promising inputs to evaluate next. If the iteration number of the next iteration exceeds the number of iterations set for exploration, this newly computed "optimal" input will be actuated on the aerodynamic model during the next iteration. Finally, after the maximum number of iterations is reached, the data are saved and the results are plotted.

### 3.4.2. Experimental phases

The simulation experiments consist of two distinct phases, which are characterized by the nature of the actuated wing shapes. During the initial exploration phase, also referred to as the wandering phase, the wing shape and angle of attack inputs actuated are pseudo-random. During the later optimization phase, the most promising solutions according to the optimizer are actuated.

The goal of the wandering phase is to gather a training data set and to train the global on-board model on this data set. In this phase, a minimum amount of global data is gathered such that an estimation of the approximate location of the global optimum can be made. Since the scope of RBFN models is limited to the domain spanned by their training data, it is important that during the wandering phase points throughout the entire region of interest are visited. The aim of the subsequent optimization phase iterations is then to investigate the most promising wing shape and angle of attack combinations on the real system and to locally improve the on-board model based on the measured system outputs.

In applications of the optimization framework to real-world aircraft, the wandering phase would also be performed once, on a test flight. In regular operation, the optimization algorithm would operate as it does in the optimization phase of the experiment. While the test flight data points in the training set provide the global information necessary for the optimizer to find the region of the global optimum, the subsequent measurements provide additional information which helps the optimizer converge onto the real-world optimum for the given conditions.

**Wandering inputs generation** The angle of attack and the virtual inputs actuated during the wandering phase need to be distributed over the entire input domain. In order to guarantee a good distribution of the sampled points over the domain, they are generated using a Sobol sequence. The Sobol sequence, first proposed in [62], aims to generate an evenly multidimensional distribution for fairly small initial sets and to generate the best uniformity of the distribution for $N \to \infty$.

The most commonly used method of pseudo-random sampling from multi-dimensional distributions is Latin Hypercube Sampling (LHS) proposed in [63]. LHS methods generate more evenly distributed samples than the Sobol sequence per dimension. The Sobol sequence, on the other hand, generates a distribution of sampled points that is more evenly spread of the multidimensional sample space. Therefore, the advantage of LHS methods diminishes with increasing domain dimensionality. In [64], LHS and Sobol sequence sampling were compared for the purpose of sampling point distributions for high dimensional integration. The Sobol sequence was observed to have superior performance, i.e., fewer sampled points were required for convergence of the high dimensional integration. Hence, because the search domain is already six-dimensional in the current configuration, and more dimensions are expected to be added in future work, the Sobol sequence was selected as the method for generating the pseudo-random inputs for search space exploration during the wandering phase.

The sampling of the six-dimensional Sobol sequence is implemented through the Sobolengine included in Pytorch [65]. The sampled six-dimensional coordinate vectors $\mathbf{S}_{x_n}$ are normalized ([0,1]).

The sampled inputs $\mathbf{U} \in \mathbb{R}^6$ are given by Eq. (3.12). Note both the angle of attack and the virtual inputs are pseudo-randomly sampled, i.e., $\mathbf{U}$ comprises $\alpha$ and $\mathbf{u}$.

$$\mathbf{U} = \mathbf{S}_{x_n}(\mathbf{x}_{\max} - \mathbf{x}_{\min}) + \mathbf{x}_{\min} \tag{3.12}$$

The sampling of normalized points guarantees that the sampled angle of attack is within the set bounds. However, the bounds on the actuator angles are the result of the combined virtual inputs. Therefore it needs to be separately verified that the sampled virtual shape function inputs do not result in a shape that exceeds the actuator limits. To this end, the 5 virtual inputs are converted to the corresponding twelve actuator angles by evaluating Eq. (4.1) at the spanwise locations of the morphing actuators. If any of the twelve actuator angles exceeds the $\pm 25$ degree limits, the corresponding sample is skipped and a new sample is drawn from the Sobol sequence.

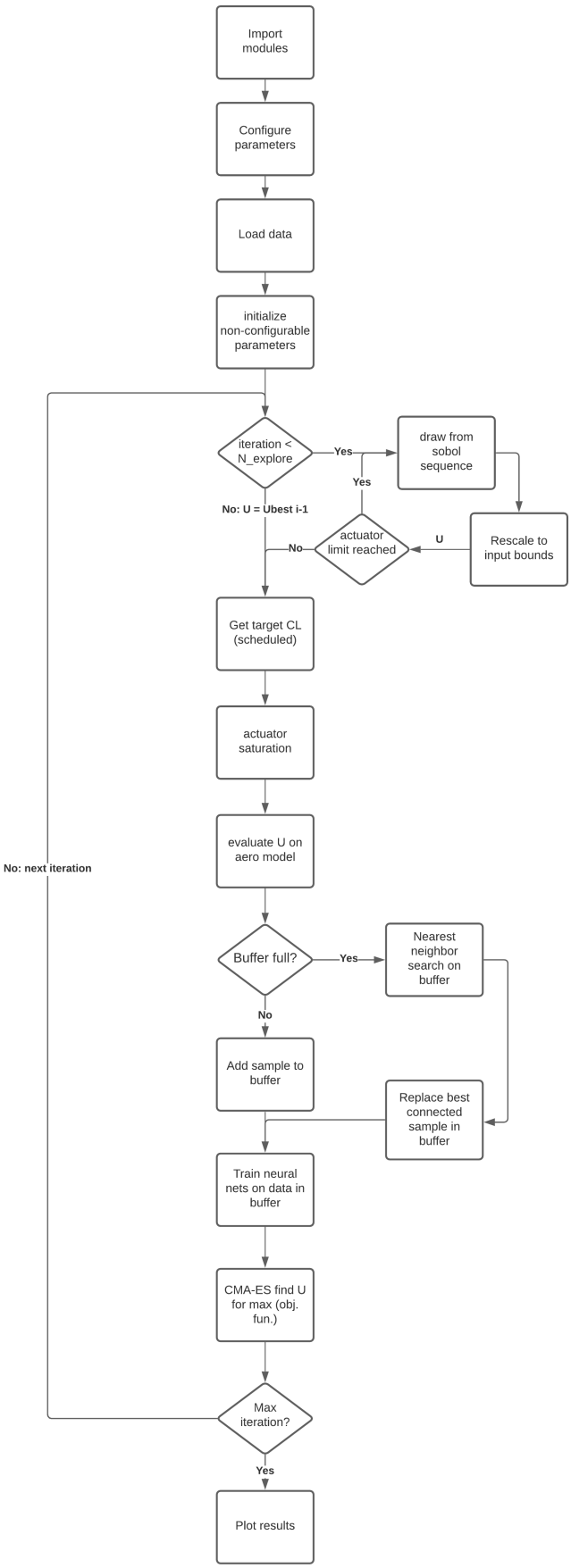



Figure 3.29: Flow diagram of online shape optimization simulation algorithm.

# 4

# Black-box Online Aerodynamic Performance Optimization for a Seamless Wing with Distributed Morphing Control

In this paper, the online wing shape optimization framework is presented, and the results from simulations on the aerodynamic model are discussed.

This paper has been accepted by the 2022 AIAA SciTech Guidance, Control, and Dynamics Conference. An updated version of this paper will be submitted to Journal of Guidance, Control, and Dynamics.


## Abstract

**Morphing is a promising bio-inspired technology, with the potential to make aircraft more sustainable through adaptation of the wing shape for best efficiency at any flight condition. This paper presents an in-simulation demonstration of a novel online black-box performance optimization strategy for a seamless wing with distributed morphing control. The presented method seeks to extend the scope of state-of-the-art online performance optimization methods by combining a global Artificial Neural Network (ANN) on-board model with a derivative-free evolutionary optimization algorithm. The effectiveness of the optimization strategy was tested on a Vortex Lattice Method (VLM) aerodynamic model of an over-actuated morphing wing that was corrected using previously collected wind tunnel data. Simulations show that the proposed method is able to control the morphing shape and angle of attack to achieve various target lift coefficients with better aerodynamic efficiency than the unmorphed wing shape. The global nature of the on-board model allows the presented method to find shape solutions for a wide range of target lift coefficients without the need for additional model excitation maneuvers. Compared to the unmorphed shape, up to $14.6\%$ of lift-to-drag ratio increases were achieved on the model.**


## Nomenclature

| | | |
|---|---|---|
| $A$ | = | amplitude |
| $A$ | = | aspect ratio |
| $C_{\mathrm{J}}$ | = | cost penalty constant |
| $C_{\mathrm{L}}$ | = | lift coefficient |
| $C_{\mathrm{L_t}}$ | = | target lift coefficient |
| $C_{\mathrm{D}}$ | = | drag coefficient |
| $C_{\mathrm{D_0}}$ | = | zero-lift-drag coefficient |

$c$ = chord
$D$ = drag force
$D_i$ = induced drag force
$e$ = Oswald efficiency factor
$f$ = frequency
$J$ = cost
$L$ = lift force
$S$ = power spectral density
$T_i$ = $i^{th}$ Chebyshev polynomial
$u_i$ = $i^{th}$ virtual input
$W$ = neural net weights
$X$ = buffer model inputs
$x_{\text{hinge}}$ = hinge location as a fraction of chord length
$x_0$ = initial solution point
$y$ = buffer model outputs
$y$ = spanwise coordinate
$Z$ = frequency domain signal
$z_{\text{te}}$ = trailing edge displacement
$\alpha$ = angle of attack
$\Delta_f$ = change in frequency
$\delta_{\text{f}}$ = flap deflection
$\phi$ = phase
$\sigma_0$ = initial standard deviation
$\theta$ = actuator angle

Subscripts

i = iteration
m = measured
s = saturated

# 4.1. Introduction

Over the past century, aircraft have become increasingly more efficient. During the 1960's improvements in engine technology and wing design led to large improvements in aircraft fuel economy. In recent years, this trend of increasing efficiency has started to stall out, as the conventional aircraft designs converged. In order to further reduce both the cost of flying and environmental pollution, more radical departures from the conventional aircraft design are needed. One promising technology is active wing morphing. Derived from the greek 'morphos' meaning shape, morphing technology enables the shape transformation of the wings in-flight. Morphing is far from a new technology. The Wright Flyer, the first successful heavier-than-air powered aircraft, relied on twist morphing of its fabric-wrapped flexible wings to achieve roll control [1]. However, as aircraft flew with ever increasing speeds, higher wing rigidity was required, which made morphing fade out in 1940's-2000's. In recent years, morphing is again made possible by advanced developments in material science such as shape memory alloys, compliant mechanisms and piezoelectrics [2].

The ability to reshape the wing in flight, introduces the problem of determining what that shape should be for a wide range of operation conditions. The current method for cruise drag minimization is the scheduling of configuration settings through lookup tables as a function of gross weight, airspeed, and altitude. These lookup tables generally depend on analytical models, validated with wind tunnel or test flight data. However, different operating conditions, aircraft production variances and repairs can result in uncertainties in the table-lookup method.

Online optimization has the potential to tailor the wing shape to any specific flight condition for achieving the best aerodynamic performance based on in-flight measurements. Much like birds, a smart morphing-wing aircraft could sense its environment and adapt its wings' shape to achieve the best performance in any condition, making it fully mission-adaptive. However, many challenges remain on the path towards operational smart morphing aircraft wings. To begin with, any online optimization method is reliant on the ability to accurately evaluate the aircraft's performance using on-board sensors.

Furthermore, only a very limited amount of search space exploration could realistically be afforded on a typical commercial flight. Ideally, a global optimum should be found with limited local explorations.

A Real-time adaptive least squares drag minimization approach has been proposed by Ferrier et al. [3] for the Variable Camber Continuous Trailing Edge Flap (VCCTEF) described by Nguyen [4, 5]. This strategy uses a recursive least squares algorithm to estimate the derivatives of the aerodynamic coefficients with respect to the system inputs. The optimal wing shape and elevator deflection are then calculated from a constrained optimization problem using the Newton-Raphson method. Improvements to the model excitation method, on-board model and optimization methods were demonstrated in wind tunnel experiments to achieve up to $9.4\%$ drag reduction on the Common Research Model (CRM) with the VCCTEF at off-design conditions at low subsonic speeds [6]. Simulations have also indicated that a $3.37\%$ drag reduction is achievable on the CRM with a distributed mini-plain flap system at Mach 0.85 [7].

Whilst the coefficients of the linear-in-the-parameters multivariate polynomial model adopted in [6, 7] can be estimated with relatively low computational cost, the model is only valid in the local region around the trim condition. This means that in order to perform real-time drag minimization across the entire flight envelope, the model parameters need to be re-estimated at every operational point. Moreover, the required model excitation maneuvers that comprise both angle of attack and flap deflection inputs would induce undesirable bumpiness and structural loads. Last but not least, the use of a local model together with a gradient-based optimization method makes the solution prone to converge onto a local optimum.

By contrast, a global on-board model, while more difficult to identify online, could allow for continuous drag minimization throughout the flight envelope. Additionally, when paired with a global optimization method, global optima with even better performance could potentially be found. The online performance optimization strategy presented in this study combines an online trained global Artificial Neural Network (ANN) on-board model, with an evolutionary optimization algorithm to improve the accuracy and flexibility of the existing methods. A high-level overview of the online optimization framework is shown in Fig. 4.1.

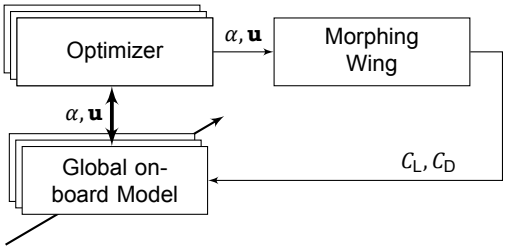


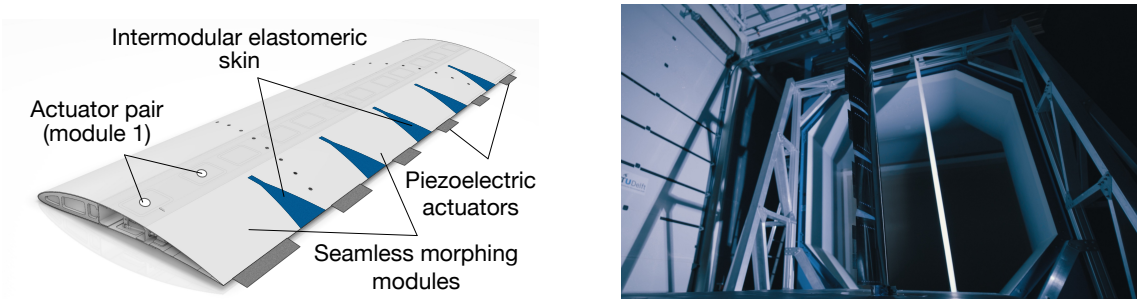

Figure 4.1: High-level overview of the online shape optimization framework.

During each iteration of the online optimization method, the evolutionary optimizer uses the on-board model to evaluate the performance of populations of wing shape and angle of attack combinations $\alpha, \mathbf{u}$ to find the most promising input combinations. The most promising input combination is actuated and evaluated on the system once per iteration. The resulting lift and drag coefficients $C_L, C_D$ are then used to improve the on-board model for the next iteration.

This online shape optimization strategy was developed to be tested on SmartX-Alpha in future wind tunnel experiments. SmartX-Alpha is an over-sensed and over-actuated seamless active distributed morphing wing [8, 9]. An overview of SmartX-Alpha is shown in Fig. 4.2a[1].

The $0.5$ m $\times$ $1.8$ m rectangular half wing is made up of six Translation Induced Camber (TRIC) morphing modules that comprise two actuators each and are seamlessly joined by a highly flexible elastomer skin. The TRIC morphing mechanism allows for both camber and twist morphing [10]. The ability to induce twisting of the trailing-edge within each module by asymmetric actuator deflection allows for smoother morphing shapes than the distributed VCCTEFs. The morphing modules enable the achievement of multiple objectives, including drag minimization, load alleviation, and flutter suppression. Control strategies for simultaneous gust and maneuver load alleviation have been demonstrated on SmartX-Alpha during wind tunnel experiments [11]. In-flight drag minimization with distributed mor-

[1]The project video can be found via `https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s`




(a) Overview of wing components.

(b) Wind tunnel setup.

Figure 4.2: Overview of the SmartX-Alpha wing demonstrator.

phing control has not been done on SmartX-Alpha. This is the focus of the study presented in this paper.

The main contributions of this paper are the presentation and first in-simulation demonstration of a novel flexible in-flight performance optimization strategy for morphing wings. The method has been tested on a VLM model that has been augmented based on previously collected experimental data in preparation of planned future wind tunnel demonstrations. This strategy improves upon the state-of-the-art by extending the scope from only the local region around a trim condition to a large range of operational lift coefficients by using a global on-board model. The on-board model consists of two artificial neural networks, serving as universal function approximators. Combined with a derivative-free evolutionary optimization method, this strategy is expected to find better performing wing shapes than the state-of-the-art gradient-based optimization because it has a low chance of converging on local optima in the optimization landscape.

The structure of this paper is as follows. The online shape optimization strategy and its individual components are described in Sec. 4.2. In Sec. 4.3, the simulation results are presented and discussed. Finally, the main conclusions are drawn in Sec. 4.4.

## 4.2. Optimization Architecture

In this section, the online shape optimization strategy and framework are presented. First, an overview of the complete optimization architecture is given. Then each of the individual components are elaborated upon in the following subsections.

The architecture of the online shape optimization framework is shown in Fig. 4.3.

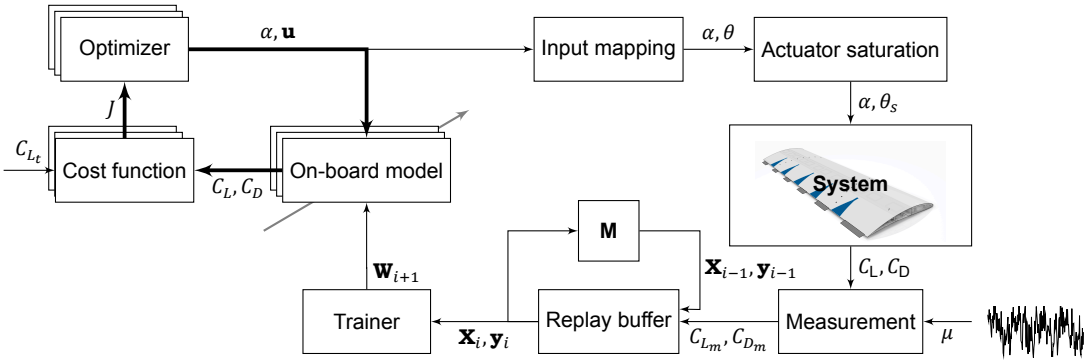



Figure 4.3: Online shape optimization architecture.

The optimization procedure involves a fast and a slow loop. The optimizer, on-board model, and cost function work together in the fast loop, marked by the shaded arrows. The optimizer evaluates angle of attack ($\alpha$) and wing shape combinations ($\mathbf{u}$) on the on-board model with a high frequency. The resulting lift and drag coefficients from the on-board model are valued with a cost function ($J$), which is also based on the target lift coefficient ($C_{L_t}$). These cost values are in turn used by the optimizer to produce a more promising set of inputs for the next iteration of the optimization loop. Once the

optimizer has converged onto the most promising set of inputs, they are evaluated on the system itself in the outer loop.

The on-board model and optimizer describe the wing shape in terms of 5 virtual inputs $u_1$, ..., $u_5$. However, since the shape of the morphing wing is controlled by 12 actuators $\theta_1$, ..., $\theta_{12}$, the $5 \times 1$ virtual input vector $\mathbf{u}$ needs to be mapped to the $12 \times 1$ actuator input vector $\theta$. Next, the actuator inputs are limited to their saturation limits of $\pm 25$ degrees. Subsequently, the wing shape and angle of attack are actuated on the system. In this study, the camber-morphing wing was simulated with an aerodynamic model of a wing with continuously distributed flaps. The resulting lift and drag coefficients are then contaminated with noise to simulate force-measurement based estimates $C_{L_m}, C_{D_m}$. The inputs and outputs of the latest evaluation are added to the replay buffer, with a replacement strategy aimed at maintaining a good global coverage of the input domain in memory. The model inputs $\mathbf{X}_i$ and model outputs $\mathbf{y}_i$ in the buffer make up the training set that is used to train the on-board model. The training of the artificial neural networks that make up the on-board model results in new network weights $\mathbf{W}_{i+1}$. From here on, a new optimization cycle is initiated with an improved on-board model.

In future operation, the network weights and replay buffer from the previous flight can be used on the first iteration. In the simulations presented in this study, weights from previous training on a different wing, and no initial buffer data were used to test the ability of the method to learn on a new system. In order to partly fill the empty buffer with data spread out over the input domain, the first 100 iterations were performed with quasi-random inputs instead of the optimizer-computed optima. This phase was dubbed wandering phase.

### 4.2.1. System

**Virtual inputs** The system described consists of 13 inputs, these are the deflections of the 12 actuators and the wing angle of attack. However, instead of using the actuator angles directly as system inputs, the optimizer and on-board model use a total of 5 virtual shape functions to describe the wing's shape. These virtual inputs $u_1, ..., u_5$ scale the five basis shapes described by the first five Chebyshev polynomials of the first kind, re-scaled onto the $[0, 1.80]$ m domain, where 1.8 m is the half-wing span. The spanwise distribution of the local actuator deflection is a linear combination of the virtual inputs and the Chebyshev polynomials $T_i(y)$ as stated in Eq. (4.1).

$$\theta_(y) = \sum_{i=1}^{5} u_i T_i(y) \tag{4.1}$$

The virtual inputs and their contributions to the actuator deflection at each actuator location are shown in Fig. 4.4, where the triangular markers indicate the actuator positions. The translation induced camber morphing mechanisms are modeled as a series of twistable plain flaps, whose local deflections vary linearly between the actuators. The deflection of each actuator is in turn dictated by the virtual inputs.

The virtual shapes reduce the 13 dimensional optimization domain for the real system to a six-dimensional one for the model, but in turn limit the number of shapes that can be formed. The basis shape functions enforce a certain degree of smoothness in the final morphed wing shape. Their use generally leads to smoother shapes than those resulting from 12 independent actuator deflections as they avoid shapes with large and frequent jumps in spanwise camber.

**Aerodynamic model** The actuator deflections described by the virtual inputs are transformed to local flap deflections to produce the geometry that is to be evaluated by the aerodynamic mode. First, the local vertical displacement of the trailing edge $z_{\text{te}}$ is computed with Eq. (4.2), which was derived from digital image correlation measurements of symmetric morphing on SmartX-Alpha [9].

$$z_{\text{te}} = \theta \cdot 5.6 \times 10^{-4} \tag{4.2}$$

The local plain flap deflection angle $\delta_{\text{f}}$ is then computed using Eq. (4.3), where $x_{\text{hinge}}$ is the location of the flap hinge as a fraction of the chord length.

$$\delta_{\text{f}} = \sin^{-1}\left(\frac{z_{\text{te}}}{c \cdot (1 - x_{\text{hinge}})}\right) \tag{4.3}$$

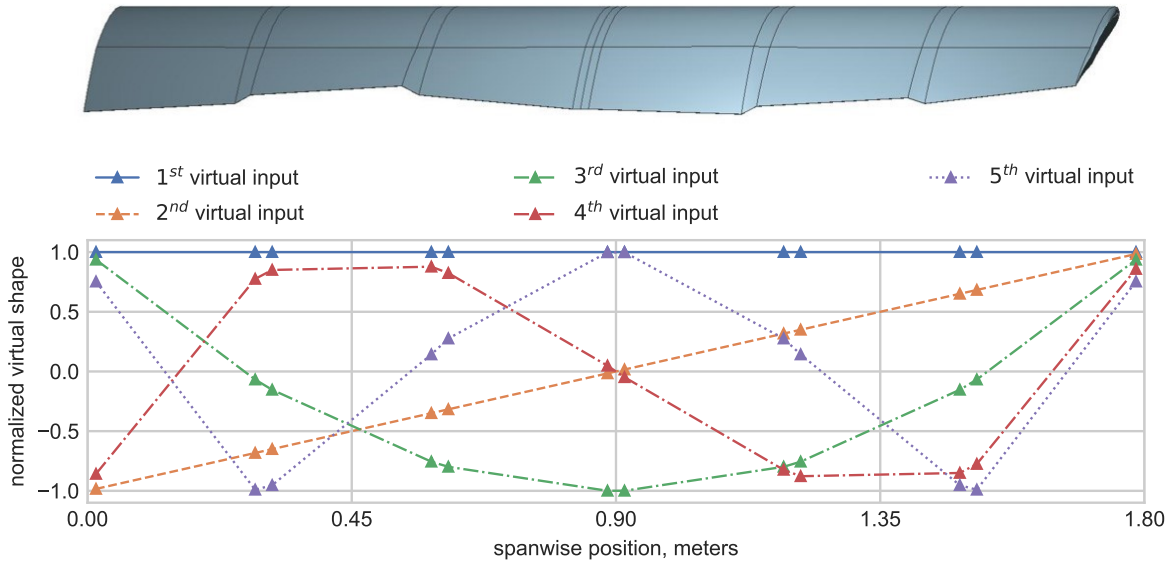



Figure 4.4: Virtual shape functions that dictate the amount of camber morphing at each actuator location.

Between the actuator locations, where the local flap angle is specified by the virtual inputs, the local flap angle varies linearly.

The aerodynamic performance of wing shape and angle of attack combinations are evaluated using a Vortex Lattice Method model implemented in the Aerosandbox python package [12]. This method is used because of its high computational efficiency and scriptability. Since, Aerosandbox is a relatively new open-source aerodynamic solver, and only one publication using this package exists in literature [13], its VLM implementation is verified against that of XFLR5 using the geometry of SmartX-Alpha. Figure 4.5 shows results from the Aerosandbox and XFLR5 VLM solvers, and wind tunnel measurements for constant spanwise actuator angle of -22 degrees. It can be observed from Fig. 4.5 that the outputs of Aerosandbox and XFLR5 VLM have high consistency.

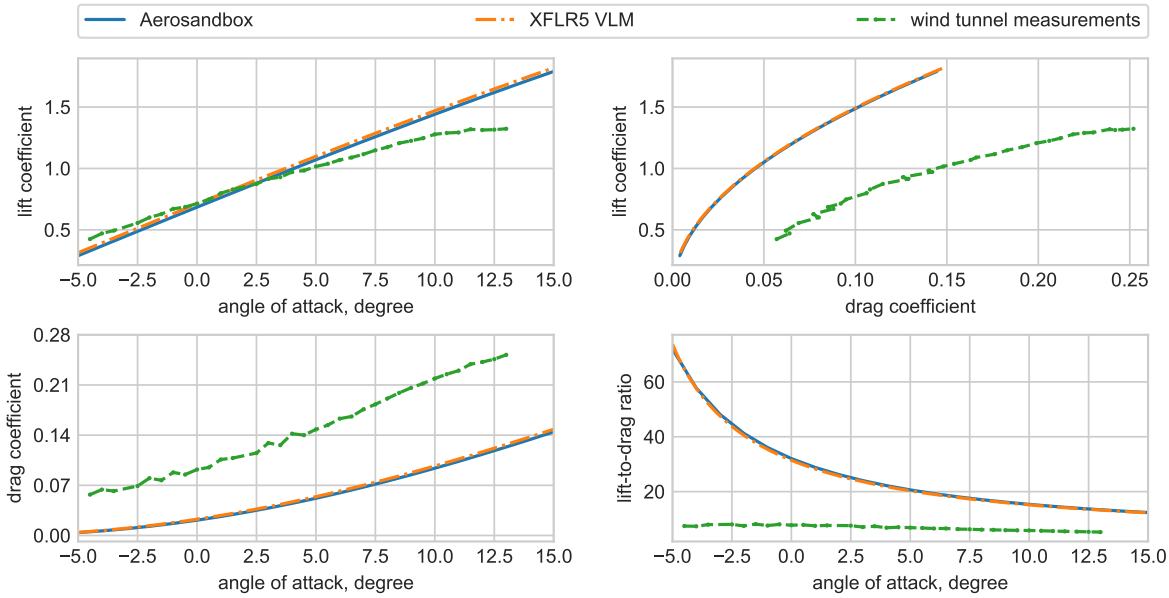



Figure 4.5: Comparison of VLM solvers with wind tunnel measurements for a constant actuator angle of -22 degrees.

However, VLM methods neglect the effects of viscosity and thickness, and can only be used to estimate lift and induced drag. As a result, the models slightly overestimate the lift slope, although their

lift predictions remain close to the wind tunnel measurements for the linear part of the lift curve. The drag on the other hand is consistently underestimated due to the lack of viscous drag effects in the model. Furthermore, while asymmetric flap deflections affect the lift-to-induced-drag ratio $\frac{L}{D_i}$ through reshaping the spanwise lift distribution, constant flap deflections along the wingspan do not affect $\frac{L}{D_i}$ at all. However, in order to optimize the morphing wings aerodynamic efficiency $\frac{L}{D}$, both the total drag and the effects of flap deflections on the lift-to-drag ratio should be modeled. Therefore, the model is augmented with an estimation of the zero-lift-drag coefficient $C_{D_0}$ and a correction to the Oswald efficiency factor $e$ based on wind tunnel data. Furthermore, the use of the corrected model is restricted to the linear part of the lift curve, $-5.0 < \alpha < 10.0$ degrees. Wind tunnel measurements from seven angle of attack sweeps at different spanwise constant actuator angles were used to estimate $C_{D_0}$ and $e$ by least squares estimation and Eq. (4.4).

$$C_D = C_{D_0} + \frac{(C_L)^2}{\pi A e} \tag{4.4}$$

The estimates for $C_{D_0}$ and $e$ were interpolated by a $1^{st}$ and $2^{nd}$ order polynomial respectively, as shown in Fig. 4.6.

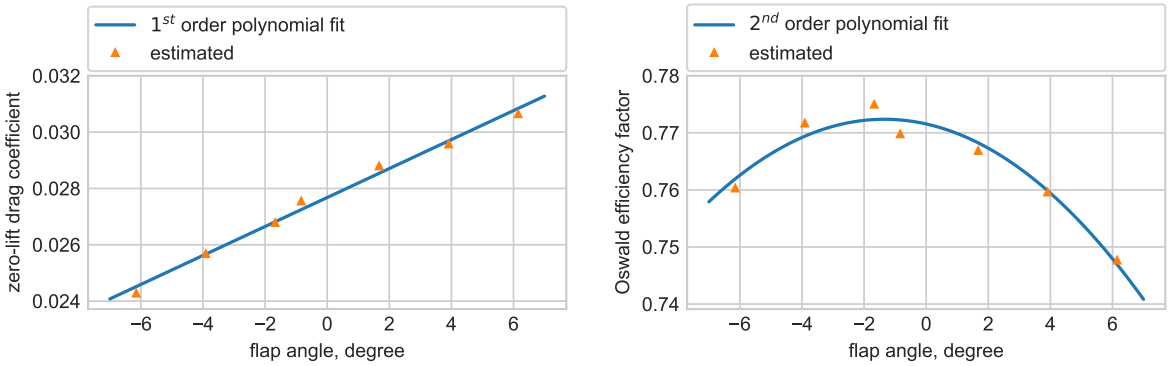



Figure 4.6: Two correction functions estimated based on wind tunnel measurements.

With these corrections, and the induced drag from the Aerosandbox model, the total drag is estimated with Eq. (4.5), where $\bar{\delta}_\mathrm{f}$ represents the mean flap angle and $e_0$ the efficiency factor of the constant deflection wing shape from Aerosandbox, estimated as 0.95.

$$C_\mathrm{D} = C_{\mathrm{D}_0}(\bar{\delta}_\mathrm{f}) + C_{\mathrm{D}_\mathrm{i}} \cdot \frac{e_0}{e(\bar{\delta}_\mathrm{f})} \tag{4.5}$$

The effects of the corrections functions are shown in Fig. 4.7 for the case of a constant -22 degree actuator angle. Compared with the uncorrected drag polar from Fig. 4.5, the zero-lift-drag correction yields a result that is much closer to the wind tunnel measurements. However, the drag is still underestimated consistently. After correcting the drag predicted by Aerosandbox with both the zero-lift-drag and the Oswald efficiency corrections, the resulting drag polar matches the wind tunnel measurements relatively closely.

Since the corrections were estimated using wind tunnel data, their validity is limited to the wing geometry and flow conditions that these measurements correspond to. In other words, the corrections shown in this section are only valid for the wing profile, the planform described above, and for an airspeed of 15 m/s at sea level air density.

**Surrogate model** For future hardware-in-the-loop experiments, the corrected model described above is intended only to be used to train the on-board model beforehand. A priori knowledge about the system is transferred with the weights of the on-board model. Although these will be adjusted during the online learning process, fewer adjustments are required than would be in the case of learning from scratch. In later stages of the technology, the network weights would hold the knowledge from previous flights.

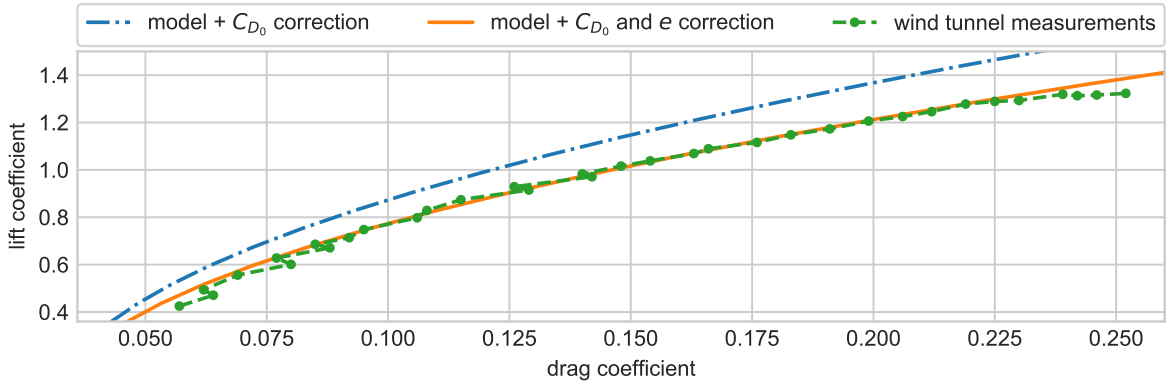



Figure 4.7: Drag polar of the corrected aerodynamic model for a constant actuator angle of -22 degrees.

In order to demonstrate the ability of the online learning shape optimization procedure to adapt to a change in the system to be optimized, a surrogate model is used in the simulation of the online shape optimization. The surrogate model represents a comparable, but yet distinctly different system.

The surrogate model is comprised of the same wing planform as the nominal model, but with a NACA4312 airfoil instead. As the VLM solver does not model the effects of airfoil thickness, only the maximum camber, and location of maximum camber are different between the nominal and surrogate models. Because there does not exist equivalent wind tunnel data for this kind of wing, the correction function estimation procedure cannot be repeated for the surrogate wing model. Instead, the correction functions are altered directly. Therefore, the surrogate model does not accurately model the aerodynamics of a known wing anymore. Instead, the surrogate model represents the aerodynamics of an unknown wing, which are relatively close to those of the nominal model. The correction functions for both the nominal and surrogate models are shown in Fig. 4.8.

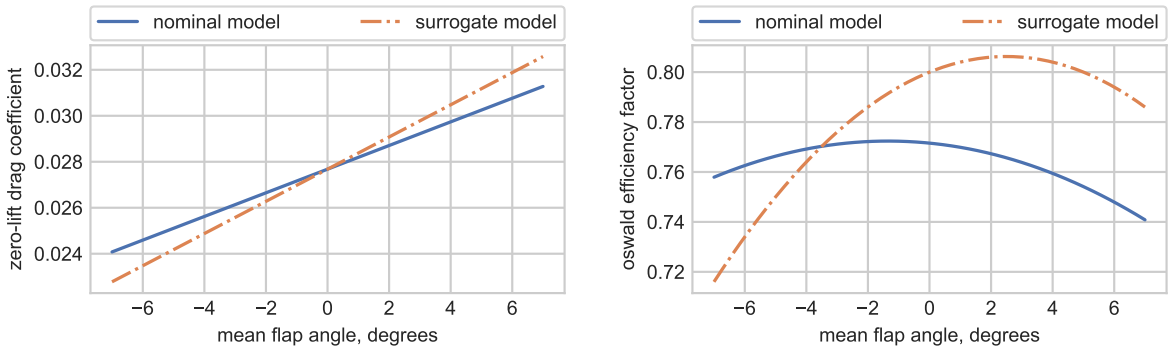



Figure 4.8: Two correction functions for the nominal and surrogate model.

**Noise simulation and filtering** Real-world measurements were simulated by adding noise to the aerodynamic model outputs. The noise realizations used were derived from sample noise measurements from the wind tunnel in which future hardware-in-the-loop experiments are planned. The Power Spectral Density (PSD) of the original noise signal, sampled at 1000 Hz was approximated by its periodogram.

The power spectral density $\mathbf{S}(f_n)$ is sampled at $n$ positive frequencies $f_n = \begin{bmatrix}\Delta_f & 2\Delta_f & \dots & n\Delta_f\end{bmatrix}^T$.

First, these power spectral densities are converted to amplitudes using $\mathbf{A}(f_n) = \sqrt{2 \times \mathbf{S}(f_n)}$, where $\mathbf{A}(f_n)$ is the $n \times 1$ amplitude vector. Subsequently, the $n \times 1$ phase vector $\phi(f_n)$ is built by assigning each spectral component a random phase between 0 and $2\pi$ radians. Next, a frequency domain signal $\mathbf{Z}(f_n)$ is constructed using Eq. (4.6).

$$\mathbf{Z}(f_n) = \mathbf{A}(f_n) \cdot e^{i\phi(f_n)} \tag{4.6}$$

Lastly, the frequency domain signal is transformed to a time domain signal using the inverse fast Fourier transform. The resulting time domain signal, and the original noise measurement are shown in Fig. 4.9.

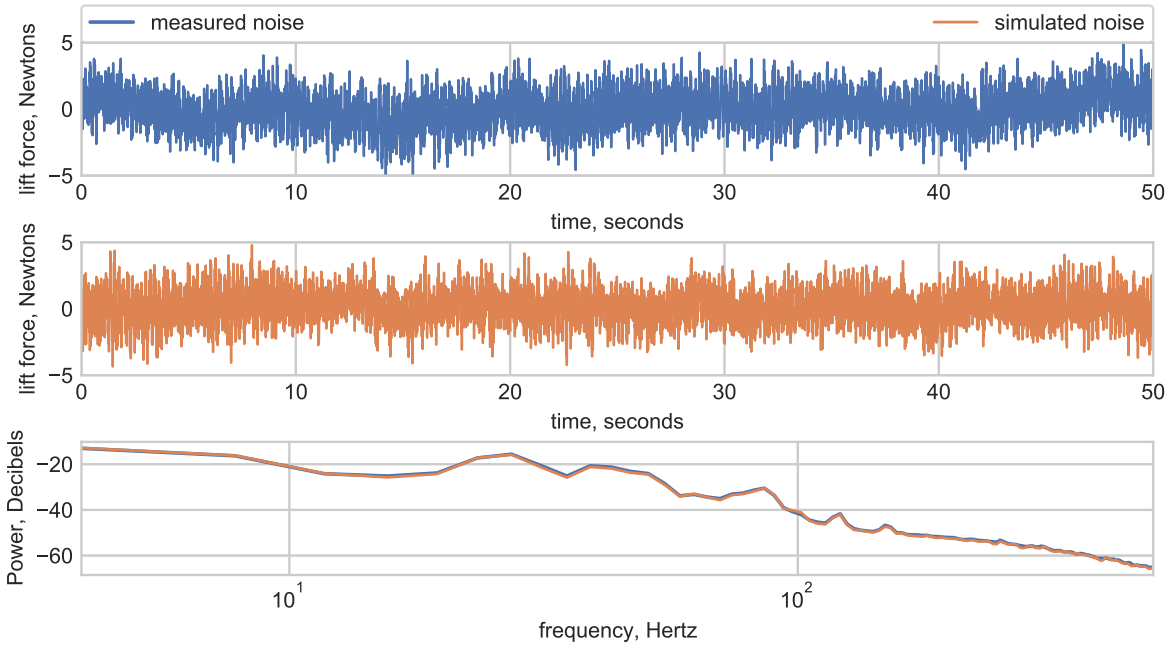



Figure 4.9: Measured and simulated lift force noise signals.

These noise realizations, although unique in the time-domain, all are made of the same spectral components. As such, the power spectral densities of both signals are nearly identical.

Finally, the system output measurements are simulated by averaging over the 50 second noise realization for noise attenuation.

### 4.2.2. Optimization

As depicted in Fig. 4.3, the optimizer does not work with the system directly, but rather on the on-board model, which can be evaluated with much lower computational costs. The genetic optimization queries the on-board model with a population of inputs to be evaluated. The quality of these inputs is then determined from the models outputs using a cost function. The optimizer in turn uses this information to generate a new group of candidate solutions. This loop is continued until the optimizer converges, after which this most promising input can be tested on the actual system.

The objective of the optimizer is to find the set of inputs $\alpha, u_1, \ldots, u_5$ that maximizes $\frac{C_\mathrm{L}}{C_\mathrm{D}}$ while meeting the target lift coefficient $C_{\mathrm{L_t}}$, and without violating the angle of attack or actuator limits. The mathematical representation of this optimization problem is shown in Eq. (4.7), where $\mathbf{u} \in \mathbb{R}^{5\times1}$ represents the virtual input vector.

$$\underset{\alpha,\vec{u}}{\mathbf{arg\,max}} \frac{C_\mathrm{L}(\alpha,\mathbf{u})}{C_\mathrm{D}(\alpha,\mathbf{u})}, \quad \textbf{subject to } \alpha \in [\alpha_{\min}, \alpha_{\max}], \quad \theta_{\min} < \theta(\mathbf{u}) < \theta_{\max}, \quad C_\mathrm{L}(\alpha,\mathbf{u}) = C_{\mathrm{L_t}} \tag{4.7}$$

**Cost function** As the optimizer queries the system with certain inputs, the corresponding outputs from the system need to be valued to in turn tell the optimizer how well the input performed. The inputs cannot simply be scored on their associated drag, as this would tempt the optimizer into minimizing the drag, by minimizing the lift produced. Instead, a promising angle of attack and wing shape combination should

result in both a low drag coefficient and a lift coefficient that is very close to the target lift coefficient. This is achieved with the cost function shown in Eq. (4.8).

$$J(C_{\mathrm{L}}, C_{\mathrm{D}}, C_{\mathrm{L_t}}) = -\frac{C_{\mathrm{L}}}{C_{\mathrm{D}}} \cdot \frac{1}{1 \times 10^{-4} + (C_{\mathrm{L}} - C_{\mathrm{L_t}})^2} \cdot \frac{1}{2 \times 10^5} \tag{4.8}$$

The cost of any set of system outputs is dependent on the lift and drag coefficients, as well as on the target lift coefficient. The cost varies linearly with the aerodynamic efficiency $\frac{C_{\mathrm{L}}}{C_{\mathrm{D}}}$ and is inverse-quadratically related to the difference between the target and actual lift coefficients. The $1 \times 10^{-4}$ and $\frac{1}{2\times10^5}$ terms are added to prevent the function from blowing up, and to scale the output to $[-1, 0]$, respectively. Two and three dimensional plots of the cost function for $C_{\mathrm{L_t}} = 0.50$ are shown in Fig. 4.10. Note that the cost increases rapidly for any deviation from the target lift coefficient, while steps in the drag-coefficient axis generally result in smaller cost variations. In other words, a solution that provides low drag at a wrong lift coefficient is valued the similarly to a solution that is associated with a higher drag at the right lift coefficient.

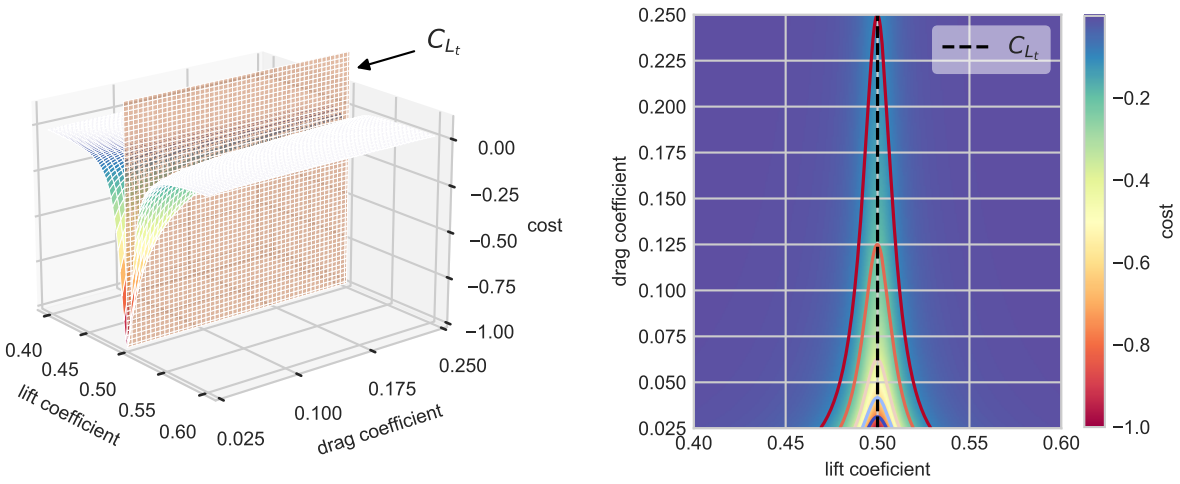



Figure 4.10: Isometric (left) and top-down (right) view of the cost function for $\mathbf{C_{L_t}} = \mathbf{0.50}$.

Additionally, the angle of attack and actuator constraints are also handled by the cost function. If a set of inputs violates any of these constraints, then its cost becomes as shown by Eqs. (4.9) and (4.10).

$$J = (\alpha_i - \alpha^\star)^2 + C_{\mathrm{J}} \tag{4.9}$$
$$J = (\theta_i - \theta^\star)^2 + C_{\mathrm{J}} \tag{4.10}$$

In the case that the angle of attack of a set of inputs to be evaluated is outside the bounds $[-2.5, 10.0]$, the associated cost will be the square of the difference between the angle of attack $\alpha$ and the middle of the domain $\alpha^\star = 3.75$ degrees plus a large constant $C_{\mathrm{J}}$. The valid range of $\theta$ is $[-25, 25]$ degrees, so $\theta^\star = 0$ and Eq. (4.10) reduces to $J = (\theta_i)^2 + C_{\mathrm{J}}$. This cost penalty constant is set to $C_{\mathrm{J}} = 10$ to ensure that the cost will always be higher than that of an input set that is not in violation of these constraints. The square term serves to provide a gradient towards the middle of the parameter domain.

**Optimizer** The optimizer's goal is to find inputs to the on-board model that minimize the cost of the model outputs as determined by the cost function described in the previous paragraph. This optimization is performed with the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) algorithm, introduced in [14]. CMA-ES is an evolutionary strategy for black-box optimization of non-linear, non-convex, and continuous problems. It can handle multi-modality and discontinuities in the function to be optimized and has desirable global performance.

CMA-ES iteratively generates a population of inputs that are subsequently evaluated on the on-board model. Based on the returned costs of these candidate solutions, the mean and covariance

matrix of the next generation's population are adapted. This process is repeated until the variation of the cost function converges to $1 \times 10^{-6}$. In the online shape optimization procedure, a population size of 150 was used. The middle of the input domain was used as the initial solution point $\mathbf{x}_0$. Furthermore, the initial standard deviation $\sigma_0$ and the scaling of the input variables were selected such that $\mathbf{x}_0 \pm 2\sigma_0$ spanned the width of the inputs domains.

The total required number of function evaluations is dependent on the population size used, and also varies naturally due to the stochastic nature of the evolutionary strategy. Optimization with higher population sizes generally requires fewer optimizer iterations, but also requires more system evaluations per iteration. With a population size of 150, an average of 180 optimizer iterations were needed with a total number of system evaluations of 27000.

### 4.2.3. On-board model

The on-board model consists of two Radial Basis Function Artificial Neural Networks (RBF ANNs) that model the mapping of the system inputs $\alpha$, $u_1$, ..., $u_5$ to the lift and drag coefficients. The $C_\mathrm{L}$ and $C_\mathrm{D}$ networks consist of a single hidden layer with 500 and 940 centers, respectively. More approximation power is needed for the $C_\mathrm{D}$ network than for the $C_\mathrm{L}$ network because of the higher degree of nonlinearity of the drag relation compared to the lift relation.

The training of the neural networks is done with mini-batch online training, with a batch size of 32. During training, the network weights are updated using the Adagrad algorithm proposed by Duchi et al. [15], with an initial learning rate of 0.01 and a mean squared error loss function.

The neural network models are not initialized with random weights, but rather with stored weights from a previous training session. In future applications, such a previous training session would be the online training performed during the most recent flight. For the simulations in this study, the starting weights for the online shape optimization will be weights from offline training on the nominal model. It is noteworthy that the simulated online optimization operates with the surrogate model in the loop. Therefore, the initial weights serve only as a starting point and do not yet constitute a model that is representative of the system to be optimized.

For the offline training of the on-board model, a data set consisting of 261,360 wing shape and angle of attack combinations and their resulting lift and drag coefficients on the nominal model was used, with $10\%$ of the data being reserved for validation. Both neural nets were trained from scratch for 2000 epochs, which equated to roughly 23 hours of training time on a laptop (Intel® Core™ i7-4510U CPU, 8.00 GB RAM).

Figure 4.11 shows the corresponding training and validation losses, converted to normalized root mean square errors (NRMSEs) for ease of comparison.

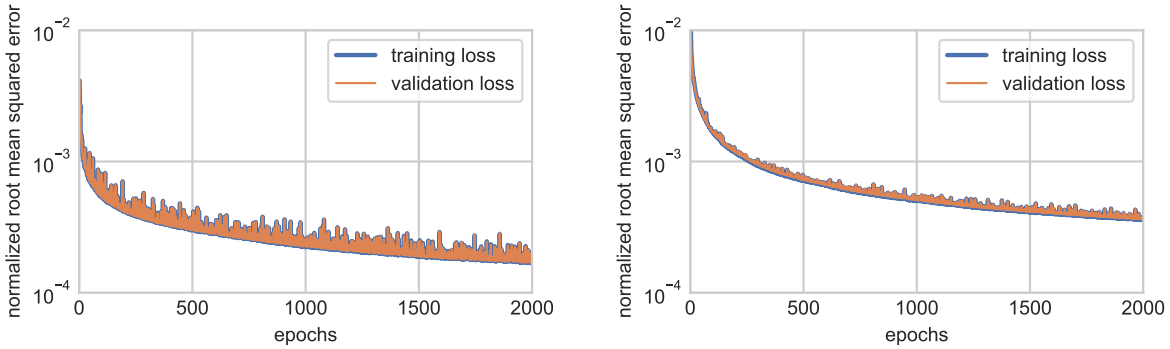



Figure 4.11: Training and validation losses for the lift (left) and drag coefficient networks (right) in offline training.

Even with the higher approximation power of the $C_\mathrm{D}$ network, the NRMSE of the $C_\mathrm{L}$ network is lower because of the lower degree of nonlinearity in the lift relation. The loss curves of both networks still exhibit a decreasing trend towards the end of the training session. The training cut-off at 2000 epochs is a trade-off between computational cost and starting point quality. The increased computational costs of further training yield an increasingly diminished return in accuracy, and the networks are only to serve as a starting point for the on-board model.

The main benefit of using the on-board model instead of direct system evaluations is the low computational cost. The CMA-ES optimizer typically requires thousands of function evaluations to converge on an optimum. On the neural network models, hundreds of input combinations can be evaluated in less than one second, whereas on the aerodynamic model each evaluation takes 1.5 seconds on average. In other words, the indirect optimization using the on-board model is approximately 2500 times faster than the direct optimization on the aerodynamic model.

On a real-world aircraft considerably more time would be required because of transients and noise filtering, making this type of optimization unfeasible for direct use on the system. Both direct optimization using the nominal aerodynamic model, and indirect optimization using the offline-trained on-board model were performed for a number of target lift coefficients. In order to make the computational time of the direct optimization more feasible, a population size of 9 was used for both. The resulting optimal shapes as computed by the CMA-ES optimizer are shown in Fig. 4.12.

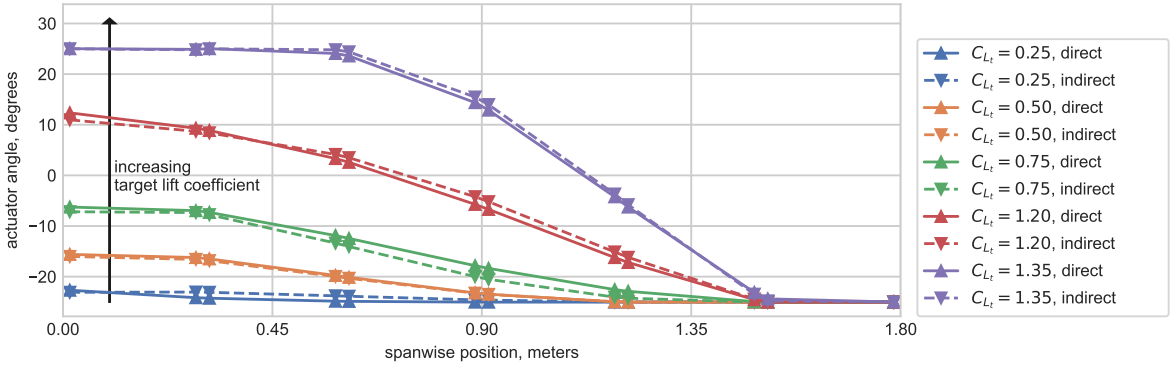



Figure 4.12: Optimal wing shapes computed directly and indirectly on the system, for various target lift coefficients.

The optimal shapes computed by indirect optimization are very close to those computed using the system directly. On average, the direct optimization took 44.7 minutes per target lift coefficient, whereas the average computational time of the indirect optimization was only 3.9 seconds (about 688 times faster).

**Replay buffer** During the online mini-batch training, the on-board model is trained on a set of training data kept in memory in the replay buffer. This buffer consists of a history of evaluated inputs and their corresponding lift and drag measurements. Since the on-board model is adjusted to adapt to this data, the contents of the buffer are of critical importance. If the training data set lacks data points in a region of the domain, then the neural nets will unlearn the previously learned from points in this region. This phenomenon, known as catastrophic forgetting was first described in [16]. Therefore, a simple first-in-first-out training set buffer will not be sufficient to learn and retain an accurate global on-board model.

Instead, the replacing of old data points when the buffer is full is based on a nearest neighbor search on all points in the buffer inspired by the coverage maximization strategy described in [17]. The data point with the lowest mean euclidean distance to its 10 closest neighbors is replaced with the latest available data point. This replacement strategy aims to maximize the coverage domain of the training set by replacing the data points in regions of high data density and holding onto samples in data scare regions of the domain.

In the future, the adaptation speed could be further improved by also prioritizing newer data points to older ones in the more densely populated areas of the domain. During the relatively short simulations presented in this study, the maximum buffer size was not reached. However, for longer operation on an aircraft it would be a critical part of the online optimization strategy.

## 4.3. Simulation Results and Discussions

In this section, the results from two simulation runs are presented. During the first simulation, the online optimization algorithm was run for 15 iterations with a fixed target lift coefficient of 0.75. During the second simulation, 275 iterations of online shape optimization were simulated with a target lift

coefficient varying between 0.25 and 1.25 and the aerodynamic efficiency of the resulting wing shapes was compared to that of the wing jig shape. The wing Jig shape is defined as the shape of the wing at rest, with all morphing actuators set to zero deflection. The wing jig shape does not have any twist.

### 4.3.1. Single target lift coefficient

The online shape optimization framework was run for 115 iterations, of which the first 100 were performed in wandering mode and the rest in optimization mode. The inputs that were evaluated on the system are shown in Fig. 4.13, where the optimization phase is marked with a red background. As expected, both the angle of attack and the virtual inputs vary within their bounds with no recognizable pattern during the wandering phase. The cost associated with these pseudo-random inputs is generally high, with one notable exception at iteration 26, where the resulting lift coefficient was relatively close to the target lift coefficient by coincidence. Quickly after the algorithm enters the optimization phase at iteration 101, the inputs plateau. At iteration 102, a shape is tried that results in higher cost than the shape from the previous iteration. Subsequently the inputs move away from this location and, the associated cost falls down and stabilizes.

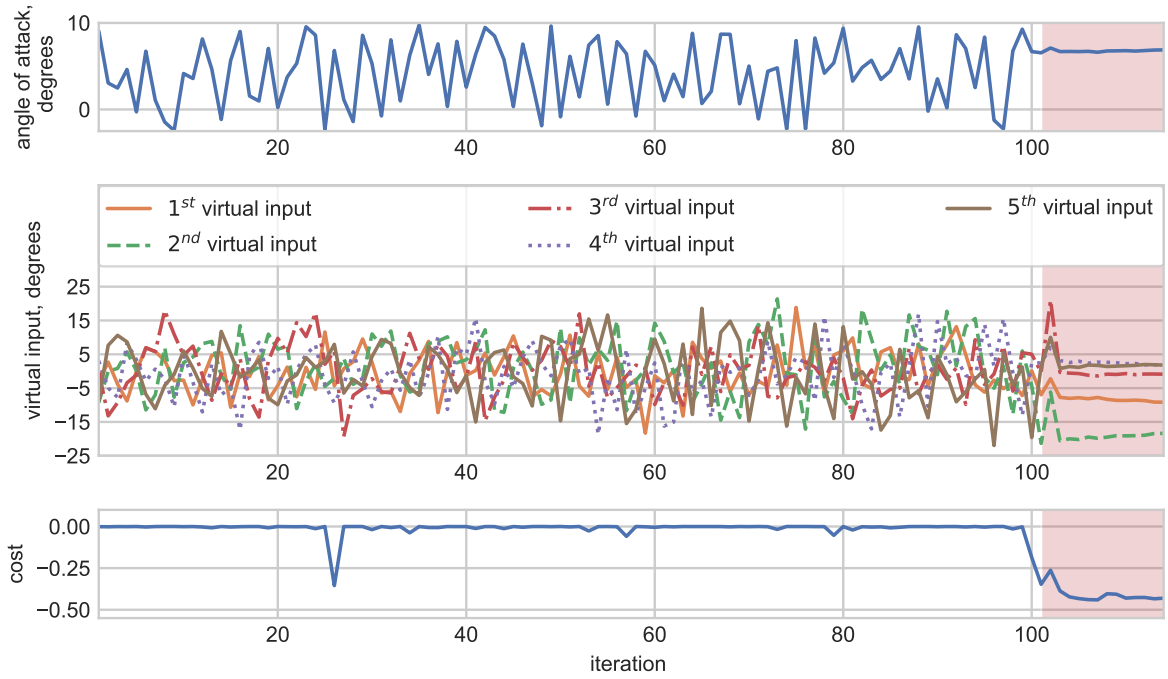



Figure 4.13: Input history for wandering and optimization (red background) with $C_{L_t} = 0.75$.

More insight into the inner workings of the optimization algorithm is provided by the optimal inputs as calculated by the optimizer, shown in Fig. 4.14. The optimal angle of attack, and optimal virtual inputs remain unchanged for the first 32 iterations of the wandering phase. During this period, the on-board model is adapting based on the measured system outputs. However, the optimal combination of inputs has not changed yet.

At iteration 32, the algorithms estimation of the optimal input changes with a sudden jump for the first time as the global minimum has been overtaken by a different input. Subsequently the estimation of the optimal inputs changes repeatedly as the on-board model keeps training on an increasing number of data points and starts to represent the system more accurately. The fact that the optimal inputs only change slowly during the optimization phase, where estimated optimal inputs are evaluated on the system, indicates that the on-board model has captured the trends in the exploratory data quite well during the wandering phase. Two spikes in estimated optimal input can be observed at iterations 99 and 102. These two points correspond to an input that seemed promising based on the on-board model at the end of the wandering phase, but once tested on the system actually yielded a lower performance than expected. After evaluation on the system, this input combination does not show up in the optimal

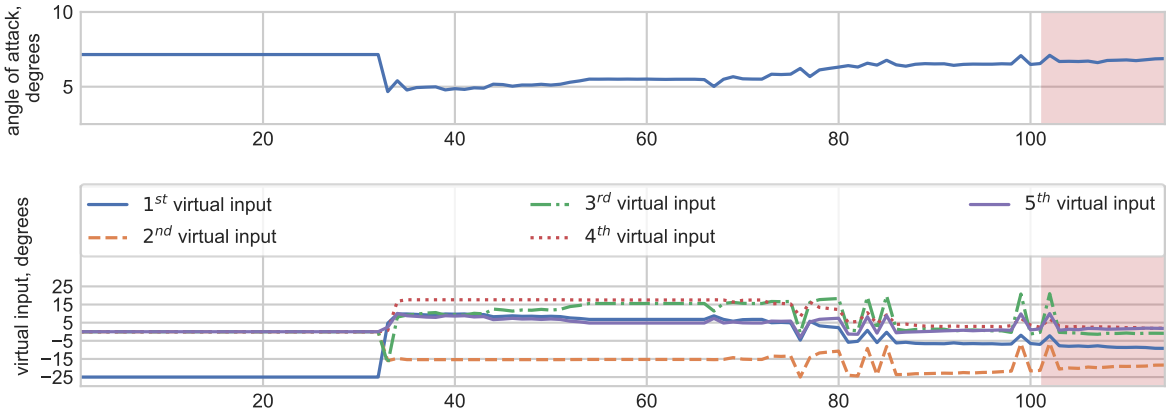



Figure 4.14: Optimal inputs as calculated by the optimizer for a target lift coefficient of 0.75.

inputs in later iterations.

The wing shapes evaluated on the system during both phases are shown in Fig. 4.15. The pseudo-random shapes span the full actuator domain. The optimal wing shape starts out with only minor changes in camber near the wing root, as compared to the wing's jig shape. Towards the tip of the wing, the camber of the wing is decreased until the actuators in the tip module hit their maximum negative deflection angles of -25 degrees. This morphing shape brings the spanwise lift distribution of this zero-twist rectangular planform wing closer to the theoretically ideal elliptic lift distribution and thereby reduces the induced drag.

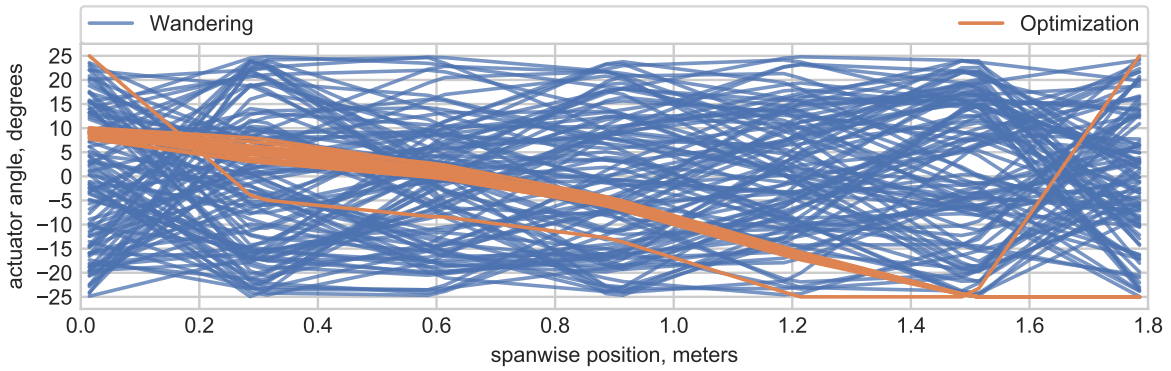



Figure 4.15: Morphing shapes evaluated on the system in the wandering and optimization phase.

One of the optimization phase shapes looks rather different from its counterparts. This is the shape that was tried on iteration 102 and resulted in an increase in cost compared to the previous iteration. In the following iterations, it was not repeated.

### 4.3.2. Varying target lift coefficient

In order to investigate the ability of the online shape optimization algorithm to find optimal inputs for different target lift coefficients without repeated exploring, the optimization phase was extended to include two repeated series of steps and a window of gradual changes in the target lift coefficient as depicted in Fig. 4.16. The quality of the solutions actuated on the system were also evaluated by comparing their lift-to-drag ratios to those of the wing jig shape.

From iterations 100 to 160, the target lift coefficient is increased by 0.25 every 15 iterations. The steps in target lift coefficient are marked with dashed vertical lines. As a direct result of the steps in target lift coefficient, steps in the computed optimal angle of attack and virtual inputs can be seen at the corresponding iterations. For the duration of the steps, the optimal inputs are stable. The cost associated with the corresponding system outputs is also stable, although it is more noisy due to the

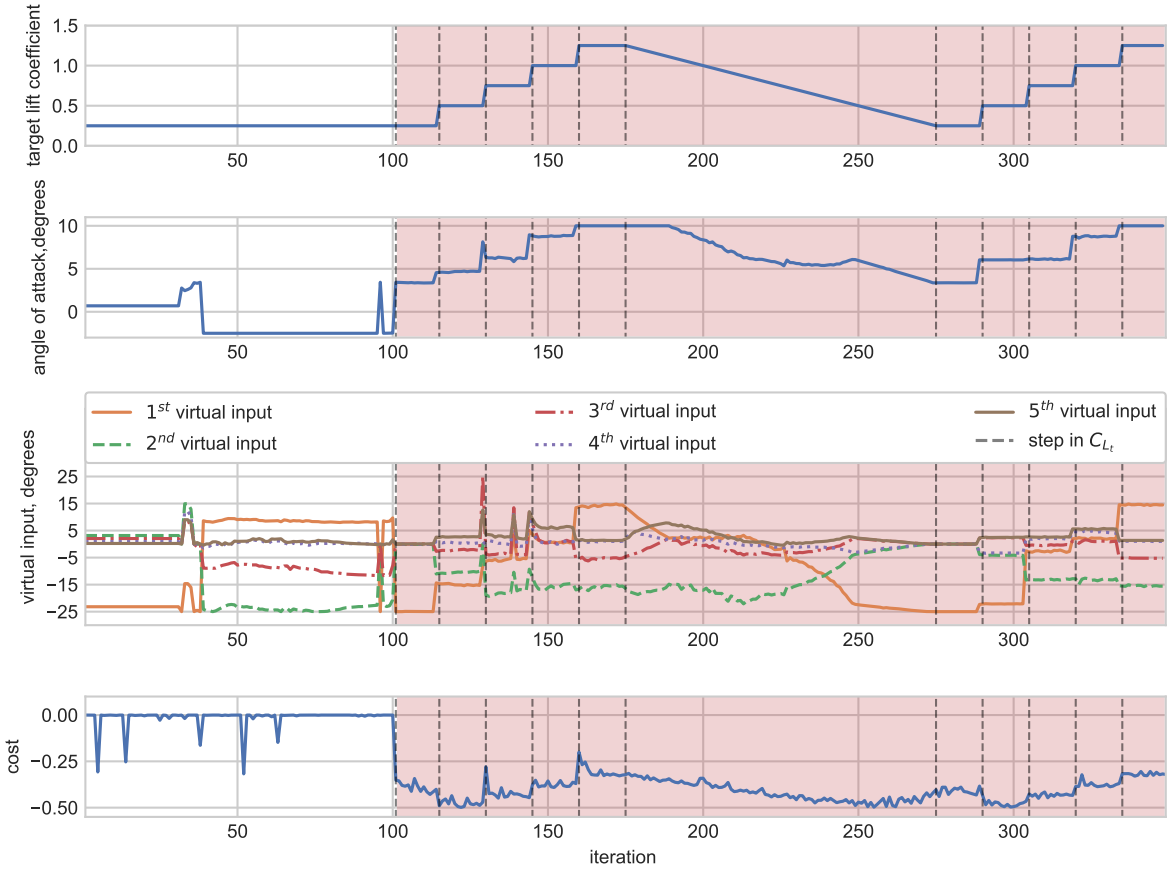



Figure 4.16: Optimal inputs computed during the wandering and optimization phases.

added measurement noise.

Between iterations 175 and 275 the target lift coefficient is decreased from 1.25 to 0.25 in steps of 0.01. As expected, the optimal angle of attack and mean camber of the optimal shape decreases as the target lift coefficient decreases. The first virtual input, which contributes a constant amount of camber morphing along the wingspan, decreases until it nears the negative actuator limit of -25 degrees between iterations 175 and 248. Meanwhile the second virtual input, which represents a linear increase in spanwise camber morphing, becomes less negative. Here the optimizer increases the negative $u_2$ input because the lower $u_1$ inputs leaves less room for spanwise lift reduction before the actuators at the wing tip hit their maximum negative deflections. Between iterations 248 and 275, virtual inputs $u_2$ through $u_5$ are decreased to zero so that $u_1$ can all the way to the -25 degree actuator limit. In other words, for the target lift coefficient of 0.25, the optimizer sacrifices the increased lift induction efficiency of a more elliptical spanwise lift distribution for an overall less cambered airfoil. This makes sense since the airfoil already is relatively highly cambered, which is more efficient for producing higher lift coefficients.

After iteration 275, the same steps in target lift coefficient are repeated. The optimal inputs are almost the same between the runs, with the exception of $C_{\mathrm{L_t}} = 0.50$ during iterations 290-305. Even though the inputs are different in this case, the costs are very similar. The average cost during iterations 115-130 is 0.475 with a standard deviation of 0.011, whereas the average cost during iterations 290-305 is 0.481 with a standard deviation of 0.018. This again highlights the importance of accurate lift and drag estimations. Any combination of inputs can only be determined to be more efficient as long as the difference is measurable. In simulations without simulated measurement noise, the revisited target lift coefficients yielded the same inputs.

The lift coefficients and lift-to-drag ratios measured during the wandering and optimization phases are shown together with those of the jig shape in Fig. 4.17. As shown in Fig. 4.17, the quasi-random shapes from the wandering phase produce lower lift-to-drag ratios than the jig shape in roughly $80\%$

of the cases. Many possible shape variations exist that are aerodynamically inefficient, whereas only a smaller subset of shapes yield better aerodynamic performance. By chance, some random inputs perform comparably or even better than the jig shape.

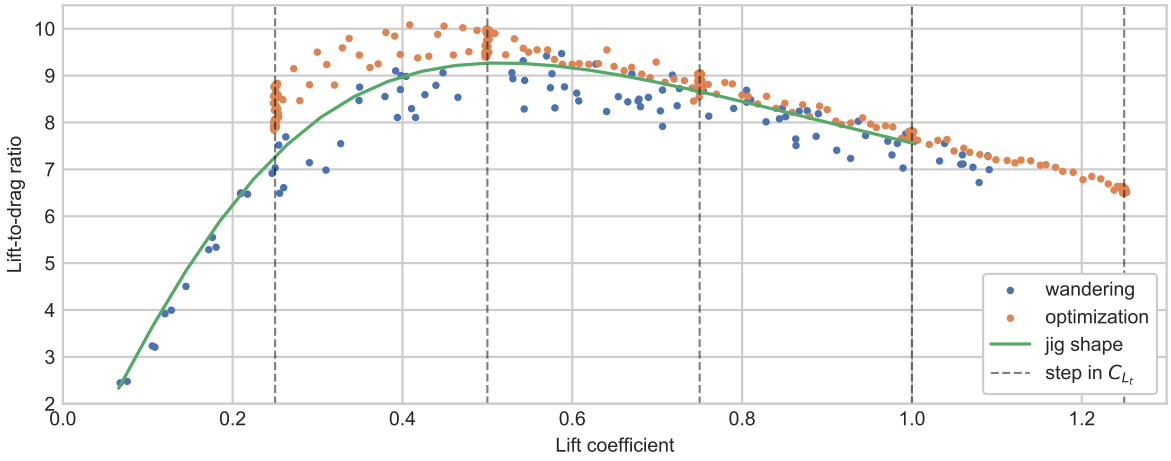



Figure 4.17: Performance comparison of the jig shape and the online optimization shapes.

With the exception of only two data points, the optimization points all outperform the jig shape in terms of aerodynamic efficiency. Although, for those two data points, the aerodynamic model output without simulated measurement noise does outperform the jig shape. Another effect of the measurement noise can be observed in the decreasing spread of the optimization point cloud with increasing lift coefficient. Naturally, as the lift and drag coefficients become larger, the lift-to-drag ratio becomes less sensitive to measurement noise. The clustering of optimization points at the target lift coefficients that were repeated for multiple iterations indicate that the optimizer is able to achieve the target lift coefficient very closely whilst also outperforming the jig shape.

As mentioned before, the relatively highly cambered airfoil is naturally efficient at inducing higher lift coefficients. Which is why the highest performance increases from active wing morphing are observed for low lift coefficients (0.25-0.50). At $C_{\mathrm{L}} = 0.25$ the lift-to-drag ratio is increased with approximately $14.6\%$. At higher target lift coefficients, less increase in aerodynamic efficiency can be gained from changing the average amount of camber. At $C_{\mathrm{L}} = 1.00$ the lift-to-drag ratio is increased with approximately $2.5\%$. Due to the rectangular planform, and absence of twist in the jig shape, reshaping of the spanwise lift distribution closer to an elliptical distribution yields aerodynamic performance increase at all lift coefficients.

## 4.4. Conclusions

In this study a novel online learning-based black-box approach to active morphing wing shape optimization was presented. Its objective is to maximize the steady-state lift-to-drag ratio for a given target lift coefficient using lift and drag measurements. The presented method combines an online-trained artificial neural network (ANN) on-board model with an evolutionary optimization algorithm. This optimization strategy was tested in simulation on a surrogate model of a seamless camber morphing wing and its performance was compared to the performance of the wing jig shape. Before optimizing, the algorithm was allowed to explore the optimization space with pseudo-random inputs for 100 iterations in the wandering phase. Subsequently, in the optimization phase, the on-board model was used by the optimizer to find the optimal wing shape and angle of attack to achieve the target lift coefficient on the surrogate wing model.

During the wandering phase, the radial basis function neural networks were able to sufficiently learn the mapping between the angle of attack, wing shape, and the resulting aerodynamic forces as to facilitate the optimizer to find wing shapes that outperformed the jig shape in terms of aerodynamic efficiency. Due to the global character of the on-board model used, the presented method was able to find wing shape and angle of attack combinations with lift-to-drag ratio increases of up to $14.6\%$ for a wide range of target lift coefficients without needing further exploration.

No published examples exist of black-box online shape optimization methods for morphing wings. To the best of the author's knowledge, this paper presents the first black-box online shape optimization approach for morphing wings. This data-driven approach is more adaptable, and potentially able to realize higher performance than conventional shape scheduling by look-up tables. The morphed wing shape could be tailored in flight to maximize the performance of the particular aircraft under consideration, rather than the performance of a model built from previous test fight data on a similar aircraft. Furthermore, the approach seeks to improve on existing local grey-box methods by retaining the information learned in a global on-board model so that a smooth and direct transition to another operating condition can be made. In other words, a desirable estimate for optimal inputs at other operation conditions is available without needing to excite the system at those conditions for model identification first.

In the present case the input space of the on-board model is comprised only of the wing shape and angle of attack. In actuality, the mapping of these parameters to the lift and drag coefficients is also influenced by the Reynolds number and Mach number. This limits the scope of the on-board model to the region where the variations of these coefficients with respect to the Reynolds and Mach numbers are small. However, due to the black-box nature of the on-board model, future studies could, with little effort, incorporate the Reynolds and Mach numbers as additional inputs to expand its scope to the full flight envelope of the intended platform. In future work, the presented online shape optimization strategy will be tested on the SmartX-Alpha seamless morphing wing in wind tunnel experiments.

# References


[1] H. R. Jex and F. E.C. Culick. "Flight Control Dynamics of the 1903 Wright Flyer." In: *AIAA Paper* (1985), pp. 534–548. ISSN: 01463705. DOI: 10.2514/6.1985-1804.

[2] Daochun Li et al. "A review of modelling and analysis of morphing wings". In: *Progress in Aerospace Sciences* 100.June (2018), pp. 46–62. ISSN: 03760421. DOI: 10.1016/j.paerosci.2018.06.002. URL: https://doi.org/10.1016/j.paerosci.2018.06.002.

[3] Yvonne Ferrier, Nhan Nguyen, and Eric Ting. "Real-time adaptive least-squares drag minimization for performance adaptive aeroelastic wing". In: *34th AIAA Applied Aerodynamics Conference*. Washington, D.C.: American Institute of Aeronautics and Astronautics Inc, AIAA, 2016. ISBN: 9781624104374. DOI: 10.2514/6.2016-3567.

[4] N. Nguyen et al. "Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing". In: *SAE Technical Papers* Sept. 2015.September (2015). DOI: 10.4271/2015-01-2565.

[5] Nhan Nguyen and James Urnes. "Aeroelastic modeling of elastically shaped aircraft concept via wing shaping control for drag reduction". In: *AIAA Atmospheric Flight Mechanics Conference August 2012*. Minneapolis, Minnesota: American Institute of Aeronautics and Astronautics Inc, AIAA, 2012. ISBN: 9781624101847. DOI: 10.2514/6.2012-4642.

[6] Nhan T. Nguyen et al. "Real-Time Adaptive Drag Minimization Wind Tunnel Investigation of a Flexible Wing with Variable Camber Continuous Trailing Edge Flap System". In: *AIAA Aviation 2019 Forum*. Dallas, Texas: American Institute of Aeronautics and Astronautics (AIAA), June 2019. DOI: 10.2514/6.2019-3156.

[7] Nhan Nguyen and Juntao Xiong. "Real-time drag optimization of aspect ratio 13.5 common research model with distributed flap system". In: *AIAA Scitech 2021 Forum*. virtual event: American Institute of Aeronautics and Astronautics (AIAA), Jan. 2021. ISBN: 9781624106095. DOI: 10.2514/6.2021-0069.

[8] Tigran Mkhoyan et al. "Design of a smart morphing wing using integrated and distributed trailing edge camber morphing". In: *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems, SMASIS 2020*. virtual event: American Society of Mechanical Engineers (ASME), 2020. ISBN: 9780791884027. DOI: 10.1115/SMASIS20-2370.

[9] Tigran Mkhoyan et al. "Design and Development of a Seamless Smart Morphing Wing Using Distributed Trailing Edge Camber Morphing for Active Control". In: *AIAA Scitech 2021 Forum*. virtual event: American Institute of Aeronautics and Astronautics (AIAA), 2021, p. 0477. DOI: 10.2514/6.2021-0477.

[10] Noud P.M. Werter et al. “Design and experiments of a warp induced camber and twist morphing leading and trailing edge device”. In: *24th AIAA/AHS Adaptive Structures Conference*. San Diego, California: publisher = American Institute of Aeronautics and Astronautics (AIAA), 2016. ISBN: 9781624103964. DOI: 10.2514/6.2016-0315.

[11] Xuerui Wang et al. “Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation”. In: *Journal of Guidance, Control, and Dynamics* (May 2021), pp. 1–14. ISSN: 1533-3884. DOI: 10.2514/1.G005870. arXiv: 2012.14520. URL: https://arc.aiaa.org/doi/10.2514/1.G005870.

[12] P. Sharpe. *AeroSandbox 2.2.11*. PyPi. Sept. 2020. [Online]. URL: https://pypi.org/project/AeroSandbox/2.2.11/.

[13] Jonathan S. Richter et al. “Performance study of a tapered flying wing with bell-shaped lift distribution”. In: *AIAA Scitech 2021 Forum*. virtual event: American Institute of Aeronautics and Astronautics (AIAA), 2021. ISBN: 9781624106095. DOI: 10.2514/6.2021-0461.

[14] N. Hansen and A. Ostermeier. “Completely derandomized self-adaptation in evolution strategies.” In: *Evolutionary computation* 9.2 (2001), pp. 159–195. ISSN: 10636560. DOI: 10.1162/106365601750190398.

[15] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. “Randomized smoothing for (parallel) stochastic optimization”. In: *Proceedings of the IEEE Conference on Decision and Control*. Maui, Hawaii: Institute of Electrical and Electronics Engineers (IEEE), 2012, pp. 5442–5444. DOI: 10.1109/CDC.2012.6426698.

[16] Michael McCloskey and Neal J. Cohen. “Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem”. English (US). In: *Psychology of Learning and Motivation - Advances in Research and Theory* 24.C (Jan. 1989), pp. 109–165. ISSN: 0079-7421. DOI: 10.1016/S0079-7421(08)60536-8.

[17] David Isele and Akansel Cosgun. “Selective Experience Replay for Lifelong Learning”. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. New Orleans, Louisiana: American Institute of Aeronautics and Astronautics (AIAA), 2018, pp. 3302–3309. ISBN: 978-157735800-8. URL: http://arxiv.org/abs/1802.10269.

# 5

# Online Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control

In this paper the wind tunnel experiments using the online wing shape optimization framework on SmartX-Alpha are presented, and the results are discussed.

This paper will be submitted to the IEEE Transactions on Control Systems Technology.

# On-line Black-box Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control

Tigran Mkhoyan, *Student Member, IEEE,* Oscar Ruland, Roeland De Breuker, Xuerui Wang

***Abstract*—Inspired by nature, smart morphing technologies enable the aircraft of tomorrow to sense their environment and adapt the shape of their wings in-flight to minimize fuel consumption and emissions. A primary challenge on the road to this future is how to use the knowledge gathered from sensory data to establish an optimal shape adaptively and continuously in-flight. To address this challenge, this paper proposes an architecture for online black-box aerodynamic performance optimization for active morphing wings. The proposed method integrates a global online-learned radial basis function neural network (RBFNN) model with an evolutionary optimization strategy. The actual wing shape is sensed via a computer vision system, while the optimized wing shape is achieved via distributed control. The effectiveness of the optimization architecture was experimentally evaluated on an active trailing edge camber morphing wing demonstrator with distributed sensing and control in an open jet wind tunnel. Compared to the unmorphed shape, a $7.8\%$ drag reduction was realized, while achieving the required amount of lift. Further data-driven predictions have indicated that even higher reductions in drag are achievable and have provided insight into the trends in optimal wing shapes for a wide range of lift targets.**

***Index Terms*—morphing, neural networks, evolutionary optimization, black-box optimization, vision-based control, wind tunnel experiment**

## I. Introduction

RECENT trends in aviation highlight the ever-increasing need for fuel economy and sustainability. Active morphing technology can offer significant benefits over conventional wing designs. Due to conflicting requirements [1], conventional wings are only optimized for a single flight condition (such as cruise). By contrast, the ability to morph wings into a desirable shape can allow aircraft to actively improve flight performance across the full flight envelope. In aeronautics, morphing has been historically investigated from the early days of heavier-than-air flight. Demonstrated in the 1900s by active roll control of the Wright Flyer in low-speed flights [2]. Current advancements in smart materials and actuators such as shape memory alloys, compliant mechanisms, and piezoelectrics [3] have allowed inventing more elaborate designs capable of achieving higher flight speeds.

While many challenges exist in designs of morphing mechanisms, the key challenge to efficiently benefit from active morphing during in-flight operation an adequate a shape optimization strategy that is highly adaptive and capable of overcoming local minima in the optimization landscape.

The currently practiced method of "determining" the optimal wing shape is through the selection of a wing shape from an offline-determined and fixed look-up table. However, the relationship between the wing shape and aerodynamic efficiency is highly nonlinear and depends on many uncertain parameters, which typically makes the look-up table method suboptimal. By contrast, a promising method is online data-driven nonlinear optimization, which can tailor the wing shape adaptively and optimally to any specific flight condition, like birds do [4].

Online shape optimization strategies for active morphing wings have been investigated extensively in the literature. In [5], a generative set search method was used to optimize the positions of eight leading and trailing edge control surfaces at a fixed angle of attack (AOA) to reduce the drag on a wind tunnel model. This local black-box optimization strategy uses a linear lift coefficient model, the parameters of which have to be identified before optimization through sweeps of the control surfaces for the given AOA. A primary disadvantage of this direct search method is the high likelihood of local optima encounters, contributed by the local scope of the linear lift model and the search character. In [6], a real-time adaptive least squares drag minimization approach has been proposed for the Variable Camber Continuous Trailing Edge Flap (VCCTEF) concept [7]. This strategy uses a recursive least-squares algorithm to estimate the derivatives of the aerodynamic coefficients with respect to the system inputs. The optimal wing shape and elevator deflection are then calculated using the Newton-Raphson method from a constrained optimization problem. Improvements to the model excitation method, onboard model, and optimization methods were demonstrated in wind tunnel experiments to achieve up to $9.4\%$ drag reduction on the Common Research Model (CRM) with the VCCTEF at off-design conditions at low subsonic speeds [8]. Simulations have also indicated that a $3.37\%$ drag reduction is achievable on the CRM with a distributed mini-plain flap system at Mach 0.85 [9]. While the proposed linear-in-the-parameters multivariate polynomial model has a low

Manuscript received October, 2021. This work was funded by Delft University of Technology.

Tigran Mkhoyan, Roeland De Breuker, and Xuerui Wang are with the Department of Aerospace Structures and Materials, Faculty of Aerospace Engineering, Delft University of Technology, Delft 2629HS, the Netherlands (e-mail: T.Mkhoyan@tudelft.nl, R.DeBreuker@tudelft.nl, X.Wang-6@tudelft.nl).

Oscar Ruland is with the Department of Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, Delft 2629HS, the Netherlands (e-mail: O.L.Ruland@student.tudelft.nl).

computational cost, a significant shortcoming is that the model coefficients are only valid around a trimmed equilibrium. The implications of this approach are that the model parameters need to be re-estimated at every operational point to perform real-time drag minimization throughout the flight envelope. Other challenges of this approach are, the necessity of re-identification by model excitation maneuvers (sweeps in AOA and control surfaces), assumption of model structure (e.g. polynomial model, order) and local nature of gradient-based optimization, causing increased fuel consumption and compromised ride comfort during the flight, reducing the adaptability of the gray-box model to different flight regimes and local optima encounters in the search domain [10]. .

Overcoming these issue with local optima and gray-box model structure can be achieved with online black-box model identification and global optimization methods. This will yield the following benefits: (i) global wing shape solutions can be found, yielding even better drag reductions, (ii) no additional model excitation maneuvers or changes to the model structure will be needed for operation at different flight conditions. The global optimization approach, however, does not come without challenges.

Global optimization methods generally require more objective function evaluations than gradient-based methods, which makes them impractical for direct application to complex aerodynamic shape optimization. Low evaluation cost global surrogate models may provide a solution by sample-efficiently generalizing the information gathered via onboard sensors [11], [12]. Widely used methods for surrogate modeling in the literature are Polynomial Regression (PR) [13], Artificial Neural Networks (ANNs) [14], [15], radial basis function (RBF) models [16], and Gaussian process (GP), also referred to as kriging [17], [18]. Evolutionary optimization methods, combined with aerodynamic surrogate models, have been effectively utilized as global black-box optimization method for aerodynamic shape optimization problems [19], [20]. In [21] further improvement to local minima have been proposed by two-stage approach for aerodynamic wing planform design, using global kriging surrogate model and an evolutionary gradient-based optimization

A critical factor for global and adaptable in-flight optimization is the online identification of surrogate models. A surrogate-assisted evolutionary optimization framework was proposed in [22], combining data-parallel global GP model and RBF models to accelerate convergence of gradient-based evolutionary optimization algorithm. In online data driven framework, ANNs are a suitable candidate for function approximation due to their adaptability and ability to approximate complex non-linear functions. However, designing an ANN-based approximation strategy requires problem-specific training, suitable network structure and tradeoff between computational cost and accuracy. RBF neural network structure has demonstrated faster learning and higher accuracy over multi-layer perceptron (MLP), for global model identification of nonlinear dynamics in power systems [23].

This paper proposes an online performance optimization strategy which, integrates a data-driven global ANN onboard model, with an evolutionary algorithm for global optimization to achieve effective and adaptable online performance optimization.

To obtain actual shape of the morphing wing in real-time, vision-based control is proposed in this study. Previous research, conducted on non-invasive motion tracking of flexible and morphing systems subject to gust excitation, has demonstrated effectiveness and high flexibility with vison-based systems [24]. Similar setups have also been used as a stabilization system for flapping-wings [25], obstacle avoidance for UAVs [26], and flight maneuvering tracking [27]. This methodology and corresponding framework with distributed control architecture is validated experimentally on a seamless morphing wing, the SmartX-Aalpha, in the Open Jet Facility (OJF) of Delft University of Technology. The SmartX-Alpha is an over-actuated and over-sensed morphing wing demonstrator with six distributed translation induced camber (TRIC) morphing modules [28][1]. An overview of the proposed online optimization framework is shown in Fig. 1.

The paper is organized as follows. Section II presents the online optimization methodology. An online optimization framework for with real-time vison-based control of the morphing wing is proposed in Sec. III. Section IV describes the wind tunnel experimental setup and procedures. The experimental results are presented in Sec. V. Conclusions and recommendations are discussed in Sec. VI.

## II. Optimization Methodology

### A. Chebyshev Polynomials

To maintain a degree of smoothness in the actuated wing shapes, and decouple the dimensionality of the optimization problem from the input dimensionality of the morphing wing, the shape of the wing is parameterized. The spanwise distribution of camber increments that constitute a morphed wing shape is described using Chebyshev polynomials. Many equivalent definitions of the Chebyshev polynomials of the first kind $T_n(x)$ exist, the simplest of which is $T_n(x) = \cos(n \cdot \arccos(x))$, where $n$ is a non-negative integer [29]. The expansion of this expression yields a sequence of polynomials which are orthogonal in the interval $[-1, 1]$, and where each polynomial $T_n(x)$ is of degree $n$. Within this interval, each of the polynomials is also bounded within $-1 \leq T_n(x) \leq 1$.

### B. Covariance Matrix Adaptation – Evolutionary Strategy

The optimization of the onboard model outputs is performed with the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) algorithm, proposed in [30]. CMA-ES is an evolutionary optimization strategy for black-box optimization of nonlinear non-convex continuous problems. With sufficiently large population sizes, CMA-ES has desirable global search performance [30]. CMA-ES iteratively samples populations of candidate solutions from a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$, which is uniquely identified by its mean $\mathbf{m} \in \mathbb{R}^n$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ [31]. Based on the returned costs of these candidate solutions, the mean and covariance

[1]The project video can be found via https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s

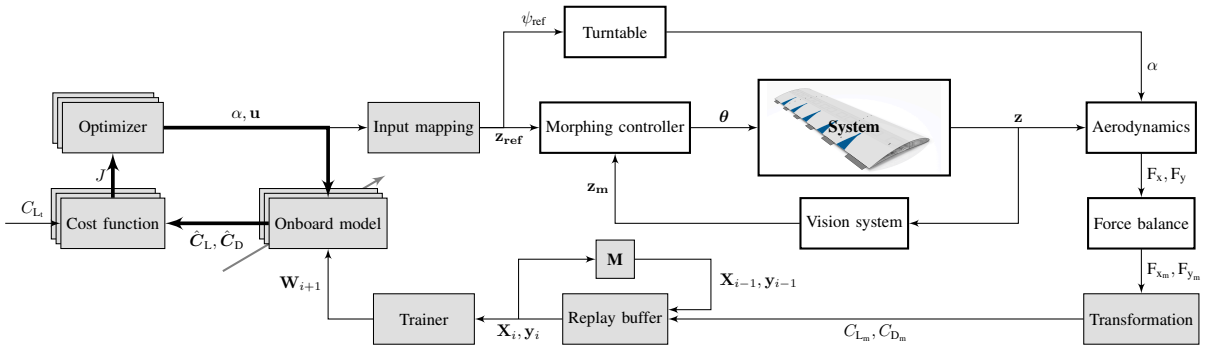



Fig. 1. Online shape optimization architecture diagram with components of the optimization algorithm marked in gray.

matrix of the next generation's population are deterministically adapted. This process is repeated until the variation of the cost function converges to below a set threshold.

With $g$ the generational counter, the $k^{\text{th}}$ offspring from the subsequent generation $g+1$ is sampled from a multivariate normal distribution $\mathcal{N}$, which is dependent on the current generation's mean search distribution value $\mathbf{m}^{(g)}$, overall standard deviation or step size $\sigma^{(g)}$, and covariance matrix $\mathbf{C}^{(g)}$ as shown in Eq. (1):

$$\boldsymbol{x}_k^{(g+1)} \sim \mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2, \mathbf{C}^{(g)}) \quad \text{for } k = 1, \ldots, \lambda \tag{1}$$

The distribution of $\mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2, \mathbf{C}^{(g)})$ is equal to $\mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)})$. From this expression, the influence of the three parameters is more apparent. $\mathbf{m}^{(g)}$ shifts the center of the multivariate normal distribution in a $n$-dimensional space, $\sigma^{(g)}$ scales the size of the distribution, and $\mathbf{C}^{(g)}$ adapts the shape of the distribution. During each iteration of the algorithm $\mathbf{m}^{(g)}$,$\sigma^{(g)}$, and $\mathbf{C}^{(g)}$ are updated based on the object parameter variations.

As shown in Eq. (2), the mean of the next generation is a weighted average of the $\mu$ best scoring search points from the sample $\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)}$. In other words, the center of the next generation's distribution is shifted in the average direction of the best performing candidates:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \, \mathbf{x}_{i:\lambda}^{(g+1)} \tag{2}$$

The adaptation equation for the covariance matrix is shown in Eq. (3), below:

$$\begin{aligned} \mathbf{C}^{(g+1)} = (1 - c_{\text{cov}})\mathbf{C}^{(g)} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \underbrace{\mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^{\mathrm{T}}}}_{\text{rank-one update}} \\ + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \\ \times \underbrace{\sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right) \left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right)^{\mathrm{T}}}_{\text{rank-}\mu\text{ update}} \end{aligned} \tag{3}$$

Here, $c_{\text{cov}}$ and $\mu_{\text{cov}}$ are the learning rate for updating the covariance matrix and weighting parameter between rank-one and rank-$\mu$ updates, respectively.

The rank-$\mu$ update uses information from previous generations to improve the reliability of the covariance matrix estimator for small population sizes. The rank-one update exploits the directional information from past generations using the evolution path $\mathbf{p}_c^{(g+1)}$.

The evolution path is a sum of successive steps as defined as follows:

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \tag{4}$$

, where $c_c$ is the learning rate for cumulation for the rank-one update, and $\mu_{\text{eff}}$ is the variance effective selection mass in Eq. (5), below:

$$\mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1} \tag{5}$$

The overall standard deviation $\sigma^{(g)}$ scales the size of the search distribution based on the length of the evolution path compared to its expected length under random selection as shown in Eq. (6):

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|\mathbf{p}_\sigma^{(g+1)}\right\|}{\mathrm{E}\,\|\,\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right) \tag{6}$$

In this equation, $c_\sigma$ and $d_\sigma$ are the learning rate for the cumulation for the step size control and a damping parameter, respectively. The scaling of the distribution with $\sigma^{(g)}$ can be used to either broaden the distribution's search space or to focus it. When an evolution path is relatively long, the successive steps are roughly in the same direction and the step size should be increased so that fewer iterations are needed to cover the distance. Conversely, when the evolution path is short, the successive steps, at least partially, cancel each other out and the step size should be decreased [31].

### C. Radial Basis Function Artificial Neural Networks

The global surrogate model, herewith referred to as the onboard model, approximates the mappings from the angle of attack and wing shape inputs to the lift and drag coefficient outputs using two Radial Basis Function Neural Networks (RBFNNs). These types of NNs, first introduced by [32], use RBF as activation function and are widely used as function

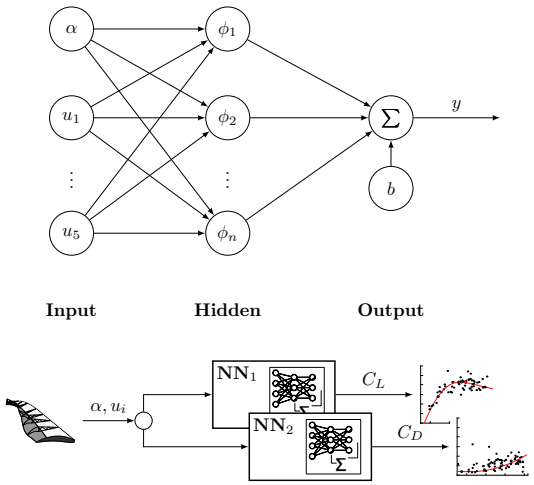



Fig. 2. Single-hidden-layer radial basis function neural network architecture.

approximators, particularly suitable for multi-parameter and scattered data [33] (e.g. aerodynamic flight data).

The architectures of the two single-hidden-layer RBFNNs used in this study are shown in Fig. 2. The six inputs to the networks are the angle of attack $\alpha$ and the five virtual inputs that describe the wing shape $u_1, \ldots, u_5$, and their respective outputs are the lift and drag coefficients $C_\mathrm{L}, C_\mathrm{D}$.

The RBFNNs consist of three layers. The first layer is the input layer. It transfers the 6-dimensional input data to each of the $n$ nodes in the subsequent (hidden) layer, consisting of $n$ neurons with RBF activation functions. Equation (7) represents the hidden unit activations given by the basis functions $\phi_j$ (e.g., Gaussian basis functions). These depend on the input activations from the previous layer $\boldsymbol{x}$, and on the parameters $\boldsymbol{\mu}$ and $\sigma_j$ [33], where $\boldsymbol{\mu}$ represents the RBF location in $\mathbb{R}^k$ and $\sigma_j$ the RBF radius.

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\mu_j}\|^2}{2\sigma_j^2}\right) \tag{7}$$

Both the center locations and the radii of the RBF basis functions are determined by the network training process. In the third (linear output) layer, the activations of the hidden layer neurons are multiplied with their respective weights $w_j$ and summed to form, together with a bias parameter, the network output. These weights and the bias parameter are also determined by the training process.

### D. Adaptive Gradient Descent

The quality of an ANN model with a specific set of network parameters is expressed in a prediction error metric (loss). Gradient descent is a commonly used algorithm for stepwise optimisation of network parameters by iteratively approach a loss minimum. The next set of network parameters at timestep $t$ are given by $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}_t} J(\boldsymbol{\theta}_t)$, where $\nabla_{\boldsymbol{\theta}_t} J(\boldsymbol{\theta}_t)$ is the gradient of the loss function with respect to the network parameters, and $\eta$ is the learning rate, which determines the size of the steps taken.

The critical parameter, responsible for convergence of the training, is the learning rate $\eta$. In methods such as stochastic gradient descent (SGD), the tuned value of $\eta$ is problem-specific, lower values corresponding to slow convergence rate and higher values faster rates and higher risk to divergence. Adaptive gradient descent (AdaGrad) is a widely used method for automatic tuning of $\eta$, generally yielding improved training [34]. . The AdaGrad optimizer adapts one learning rate for each dimension using historical data as shown in Eq. (8):

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i} \tag{8}$$

Here, $g_{t,i}$ is the gradient of the loss function w.r.t. parameter $i$ at time $t$, and $G_{t,ii}$ is the cumulative sum of the squares of the past gradients as shown in Eq. (9):

$$G_{t,ii} = \sum_{\tau=1}^{t} g_{\tau,i}^2 \tag{9}$$

The $G_{t,ii}$ in the denominator in Eq. (8) is responsible for gradual diminishing of $\eta$, based on gradient histories.

## III. Optimization Architecture

In this section, the optimization strategy is presented. First, an overview of the architecture is described, followed by more detailed explanations of the individual components.

An overview of the optimization architecture is shown in Fig. 1. The optimization method involves a fast model optimization inner loop and a slower system evaluation outer loop. In the fast loop, the evolutionary optimizer uses the onboard model to estimate the performance of generations of candidate angle of attack and wing shape combinations $\alpha, \mathbf{u}$. The estimated performance $\hat{\boldsymbol{C}}_\mathrm{L}, \hat{\boldsymbol{C}}_\mathrm{D}$ of these candidate solutions is scored with a cost function $J(C_\mathrm{L}, C_\mathrm{D}, C_{\mathrm{L}_\mathrm{t}})$, where $C_{\mathrm{L}_\mathrm{t}}$ is the target lift coefficient, and this information is fed back to the optimizer so that a new generation of improved candidate solutions can be generated.

After the optimizer has converged on a single "most promising" input, it will be actuated on the physical system. First, the virtual shape inputs $u_1, \ldots, u_5$ are transformed to the reference trailing edge displacements at the twelve actuator locations $\boldsymbol{z}_\mathrm{ref} \in \mathbb{R}^{12}$. The angle of attack is converted to a reference turntable angle with $\psi_\mathrm{ref} = \alpha - \varepsilon$, where $\varepsilon$ is the angular difference between the relative wind direction and the turntable's zero angle direction. Then, the reference table angle and trailing edge displacements are sent to the turntable and morphing controllers, respectively. The morphing controller uses the TE (Traling Edge) displacement feedback $\boldsymbol{z}_\mathrm{m}$ from the vision system to steer the TE displacements to their reference values by controlling the morphing actuators $\boldsymbol{\theta}$. The vision system uses five infra-red cameras to track the positions of markers on the wing to determine the amount of trailing-edge displacement realized.

After the controllers have converged and the intended wing shape and angle of attack are actuated on the system, the resulting aerodynamic forces are measured using a force balance which is mounted to the turntable. The resulting measurements $\mathrm{F}_{\mathrm{x}_\mathrm{m}}, \mathrm{F}_{\mathrm{y}_\mathrm{m}}$ are transformed to the lift and drag

coefficients $C_{\mathrm{L_m}}, C_{\mathrm{D_m}}$. Both the inputs used, and the resulting aerodynamic coefficients, are added to the training set which is kept in memory. Lastly, the outer loop is completed by training the onboard model, which results in updated model parameters $\boldsymbol{w}_{i+1}$. The key blocks in Fig. 1 are elaborated in Sec. III-A-Sec. III-E.

### A. Morphing Wing

The central block in the diagram represents the autonomous morphing wing system. The wing has six distributed translation induced camber (TRIC) modules, allowing independent camber and span-wise twist morphing of local span segments [28]. The skin is actuated by an internal mechanism, allowing smooth and seamless morphing along the chord. The morphing target shape is commanded with 12 high-performance Volz DA-22-12-4112 servos per module (two per module) embedded in the wing box, allowing the trailing edge bottom skin to slide in chord-wise and span-wise direction along a guided sliding interface. This is illustrated in Fig. 3. To limit the adverse aerodynamic impacts from gaps introduced between independently moving adjacent surfaces, elastomeric skin segments are integrated between the modules. The elastomeric skin is tailored to allow sufficient flexibility for symmetric and differential actuation (inter and intra module) of $\pm 25$ degree for each servo unit. The rotational motion is converted to linear sliding motion through a ball joint linkage system, which results in morphing of the airfoil from its nominal NACA6510 aerodynamic jig shape [28].

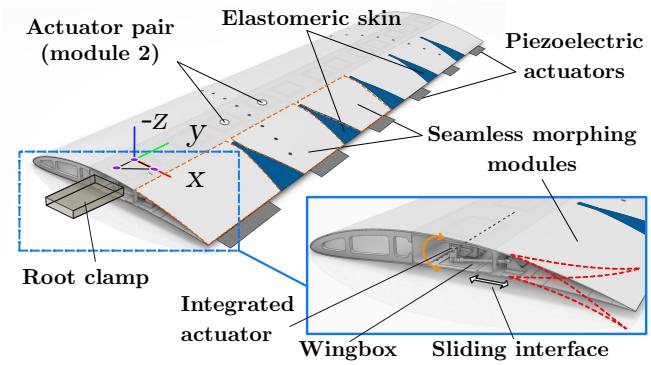



Fig. 3. Morphing wing actuation mechanism and coordinate system.

Because of nonlinear backlash effects in the morphing mechanism, the nominal NACA6510 aerodynamic jig shape can not be consistently realized with only zero degree actuation of the servo units. Since the effects of the nonlinear backlash are dependent on the actuation history, a standard actuation maneuver is included in the jig shape definition. The jig shape is defined as the shape of the wing after a pre-designed maneuver, which consists of a 20 degree doublet input to all servos, followed by a constant zero degree actuation command to all servo units.

### B. Optimization Algorithm

The goal of the optimization procedure is to realize the highest lift-to-drag ratio for a given target lift coefficient using measurements of the aerodynamic forces. The optimization algorithm comprises three main parts – namely the cost function, onboard model, and the optimizer – shown on the left side in Fig. 1. A pseudocode description of the online optimization algorithm is given by Algorithm 1.

**Algorithm 1:** Online performance optimization.

**Input:** $\alpha$, vritual inputs $u_1, \ldots, u_5$
**Output:** $C_{\mathrm{L}}, C_{\mathrm{D}}$
**while** running **do**
  **while** averaging $F_x, F_y$ **do**
    **while** $\epsilon > 1 \times 10^{-6}$ **do**
      Populate target candidate space;
      Obtain initial $C_{\mathrm{L}}, C_{\mathrm{D}}$ Estimate;
      Evaluate score;
    **end**
    Transform $u_1, \ldots, u_5 \rightarrow z_1, \ldots, z_{12}$;
    Command wing $\alpha$ and $z_1, \ldots, z_{12}$;
  **end**
  Update buffer;
  Retrain onboard model;
**end**

During each iteration of the optimization procedure, the evolutionary optimizer generates a population of candidate solutions and queries the onboard model with their angle of attack and wing shape combinations $\alpha, \mathbf{u}$. In turn, the onboard model predicts the steady-state lift and drag coefficients resulting from each of these inputs, and the cost function scores the desirability of these predicted outputs based on the target lift coefficient $C_{\mathrm{L_t}}$. Subsequently, the scores of the evaluated input combinations are used by the optimizer to generate a more promising population of input combinations. This cycle is repeated with a frequency of approximately 15 Hz, depending on the population size and available computing power. This process continues until the optimizer converges onto a single most optimal angle of attack and wing shape combination. This input combination is then actuated on the real system and the resulting measurements are subsequently used to improve the onboard model.

*1) Virtual Inputs:* Within the optimization algorithm, the shape of the wing is represented by the virtual inputs $u_1, \ldots, u_5$ rather than the trailing edge displacements at the twelve actuator locations directly. These virtual inputs scale five basis shapes, which are described by the first five Chebyshev polynomials of the first kind (Sec. II-A), re-scaled onto the $[0, 1.80]$ m domain, where 1.8 m is the half-wing span. In other words, the five virtual inputs are the coefficients of a fifth-order Chebyshev approximation of the spanwise camber distribution function that describes the morphed wing shape. This distribution function is given in Eq. (10):

$$z(y) = \sum_{i=1}^{5} u_i T_i(y) \tag{10}$$

Even though the virtual input parameterization is used to describe the wing shape within the optimization framework,

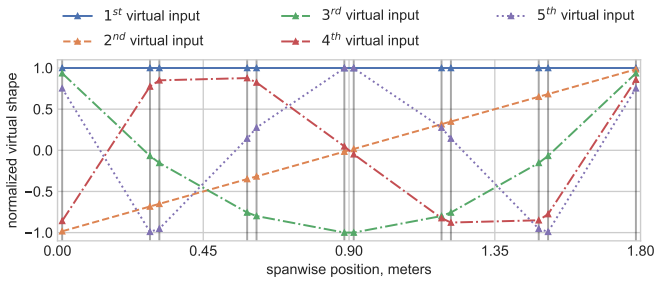



Fig. 4. Virtual input basis functions.

it is the local trailing edge deflection that directly cause the measured aerodynamic performance. The local trailing edge displacement $z_i$ at the actuator $i$ is given by $z_i = z(y_i)$, where $y_i$ is the spanwise position of the actuator measured from the wing root. The shapes described by these basis polynomials and their contributions to the amount of camber at the actuator locations are shown in Fig. 4.

By using the virtual inputs to describe the wing shape to dictate the trailing edge displacements at the actuator locations, the dimensionality of the optimization problem is reduced from 13 to 6 (five virtual shape inputs and one angle of attack input). This significantly reduces the computational load of the optimization, and also makes the computational load independent of the number of actuators on the morphing wing. Furthermore, the basis functions also serve to enforce a certain degree of smoothness in the final shapes.

*2) Optimization Problem:* The goal of the online shape optimization is find the most aerodynamically efficient wing shape and angle of attack combination without altering the intended flight path of the aircraft. The maximization of the aerodynamic efficiency $\frac{C_\mathrm{L}}{C_\mathrm{D}}$ results in reduced fuel consumption of the aircraft. Additionally, the right amount of lift force must be induced in order to maintain level flight. This constraint is posed by the target lift coefficient $C_{\mathrm{L}_t}$. Thus, the objective of the optimizer is to find the set of inputs $\alpha, u_1, \ldots, u_5$ that maximizes the lift-to-drag ratio $\frac{C_\mathrm{L}}{C_\mathrm{D}}$ on the onboard model while meeting the target lift coefficient, and without violating the angle of attack or trailing edge displacement limits.

The mathematical representation of this optimization problem is shown in Eq. (11), where $\mathbf{u} \in \mathbb{R}^{5\times 1}$ represents the virtual input vector:

$$\begin{aligned} \underset{\alpha,\,\mathbf{u}}{\arg\max} \quad & \frac{C_\mathrm{L}(\alpha,\mathbf{u})}{C_\mathrm{D}(\alpha,\mathbf{u})} \\ \text{subject to} \quad & \alpha \in [\alpha_{\min}, \alpha_{\max}], \\ & u_1, \ldots, u_5 \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}], \\ & z_1, \ldots, z_{12} \in [z_{\min}, z_{\max}], \\ & C_\mathrm{L}(\alpha,\mathbf{u}) = C_{\mathrm{L}_t} \end{aligned} \tag{11}$$

Because of the backlash effects in the actuation mechanism, the required actuator angles for any given wing shape are not unique and are unknown beforehand. Therefore, limits were imposed on the commanded local z-displacements of the trailing edge at the actuator positions $z_1, z_2, \ldots, z_{12}$ rather than on the actuator angles. The maximum absolute displacement achievable at any actuator position is dependent on the actuation of the neighboring actuators. Actuator pairs that deflect in unison are able to effectuate larger trailing edge displacements than actuator pairs that deflect in opposite directions. The minimum and maximum local vertical displacements allowed for the optimizer were selected as $-10$ mm and 10 mm respectively. This was done not to completely prevent all occurrences of actuator saturation, but rather to eliminate wing shapes that would not be achievable. The virtual inputs were constrained by the bounds of the data on which the onboard model is trained, as the ANN are not capable of extrapolating outside the data domain. The virtual inputs and angle of attack bounds pose constraints on the inputs and are implemented directly as constraints on the input space. By contrast, the TE displacement bounds and target lift coefficient pose constraints on intermediate and output variables and are implemented indirectly through the cost function scoring.

*3) Optimizer Configuration:* The CMA-ES algorithm was used to solve the optimization problem posed by Eq. (11). A relatively large population size of $\lambda = 150$ was used to improve the global search performance of the algorithm. The middle of the input domain was used as the initial solution point $\mathbf{x}_0$. To allow global convergence, the standard deviation $\boldsymbol{\sigma}_0$ and the scaling of the input variables were selected such that $\mathbf{x}_0 \pm 2\boldsymbol{\sigma}_0$ spanned the width of the domain in each of the input axes (approximate neighborhood). A suitable cost function variation convergence threshold was found at $1 \times 10^{-6}$, such that optimizer yielded adequate convergence without impacting the computational time. The computational time was further reduced by parallel query of (150) candidate solutions.

*4) Onboard Model:* The onboard model on which the candidate solutions from the optimizer are evaluated comprises two single-hidden-layer RBFNNs. These networks are continuously improved with training being performed each iteration using training data kept in memory in the replay buffer. After balanced tradeoff between approximation power and computational load, the lift and drag coefficient networks were configured with 500 and 940 neurons, respectively. The higher neuron count in the drag coefficient network (higher approximation power) was deemed necessary due to nonlinearity in the virtual input parameters (i.e. dependency on specific allocation setting), versus proportional relationship between lift and camber increase for the lift coefficient.

*5) Buffer and Training:* The online training of the networks is performed with mini-batch gradient descent with a mini-batch size of 32. During training, the RBF center locations, radii, network weights, and the bias parameter are updated using the Adagrad adaptive gradient descent algorithm with a mean squared error loss function. The initial learning rates for both networks were configured as 0.01.

The training data, comprised of previously evaluated inputs and their lift and drag coefficients, is stored in a buffer. Since the onboard model's parameters are adjusted to minimize the loss on the training data, the contents of the buffer are of critical importance. If the training data set lacks data points in a region of the domain, then the neural nets will unlearn the previously learned information from points in this region.

This phenomenon, known as catastrophic forgetting was first described in [35]. Therefore, a simple first-in-first-out training set buffer will not be sufficient to learn and retain an accurate global onboard model.

Instead, the replacement of old data points when the buffer is full is based on a nearest neighbor search on all points in the buffer, inspired by the coverage maximization strategy presented in [36]. The data point with the lowest mean euclidean distance to its 10 closest neighbors is replaced with the latest available data point. This replacement strategy aims to maximize the coverage domain of the training set by replacing the data points in regions of high data density and holding onto samples in data scare regions of the domain.

The buffer, containing approximately 240 samples, was selected such as to ensure that the memory was not depleted for a given measurement frequency (1 new measurement every 70 seconds) during a $\approx$ 4.5 hrs total measurement time. For commercial aircraft operation with long operational times, adequate adjustment of this buffer may be necessary.

*6) Cost Function:* The system outputs are $C_{\mathrm{L}}, C_{\mathrm{D}}$ The cost function is chosen such as to maximize the lift-to-drag ratio, while limiting the error between the actual and the target lift coefficient. Hence, the cost function depends not only on the system outputs, $C_{\mathrm{L}}, C_{\mathrm{D}}$, but also on the target lift coefficient.

The cost function (Eq. (12)), consist of three terms and defined as follows:

$$J(C_{\mathrm{L}}, C_{\mathrm{D}}, C_{\mathrm{L_t}}) = \underbrace{-\frac{C_{\mathrm{L}}}{C_{\mathrm{D}}}}_{\text{efficiency}} \cdot \underbrace{\frac{1}{1 \times 10^{-4} + (C_{\mathrm{L}} - C_{\mathrm{L_t}})^2}}_{\text{deviation from lift target}} \cdot \underbrace{\frac{1}{2 \times 10^5}}_{\text{scaling}} \tag{12}$$

Here, a minus sign is added to the first term is, the lift-to-drag ratio, to yield a more negative cost value (lower indicating a more desirable system output) for higher ratios. A scaling value is added to the second term, the inverse of the squared error between the actual lift coefficient and the target lift coefficient, to prevent singularities for small error values. The efficiency and lift target terms are multiplied rather than added, such that a low cost can only be achieved when high efficiency and the correct amount of lift are achieved simultaneously. Lastly, the third term serves to scale the cost function output to be in $[-1, 0]$. As an example, the cost function for $C_{\mathrm{L_t}} = 0.50$ is shown in Fig. 5.

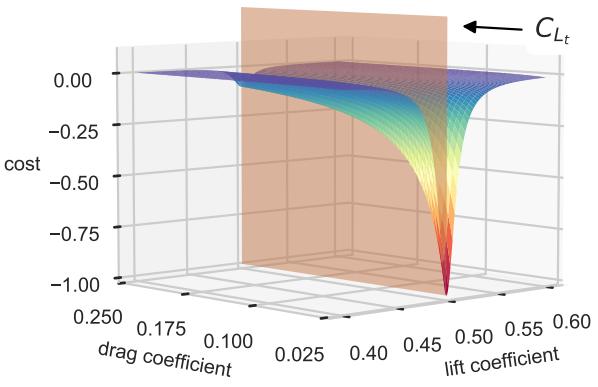



Fig. 5. The cost function for $\mathbf{C_{L_t} = 0.50}$.

It can be observed from Fig. 5 that a solution that deviates from the target lift coefficient is undesirable even if it provides low drag.

Additionally, the constraints on the vertical TE displacements $z_1, \ldots, z_{12}$ are also handled in the cost function. If the local TE displacement at any of the actuator locations is outside the bounds, the cost of the associated inputs becomes as shown in Eq. (13):

$$J = \max(|\boldsymbol{z}|)^2 + C_{\mathrm{J}} \tag{13}$$

Here, $C_{\mathrm{J}}$ is a large positive constant such that the cost will always be higher than those of inputs that do not violate the constraints. The square of the maximum absolute local TE deflection provides a cost gradient to aid the evolutionary optimizer in steering the populations of candidate solutions back to the feasible space.

### C. Vision System

*1) Vision-based Shape Reconstruction:* A real-time vision-based system is developed to meet a critical aspect of backlash compensation, and the optimization control strategy, namely accurate knowledge of the morphing wings' shape. The variable of interest to the controller is the local vertical displacement of the wing trailing edge with respect to a body-fixed coordinate system. Due to the relatively stiff wing box design (Sec. III-A), the body-fixed coordinate system, $\mathcal{F}_B$, is chosen to be near the root of the wing in the wing box section, with origin $O_B$. The displacement of the trailing edge, denoted as the $\boldsymbol{z} = [z_1, z_2, ..., z_{12}]^{\mathsf{T}}$ along 12 stations of the span, is reconstructed in the $\mathcal{F}_B$ frame in real-time by means of a 5-view vision-based tracking system. Each morphing module is fitted with a pair of active Infrared (IR) light-emmiting diodes (LEDs), with 3 additional markers for the definition of $\mathcal{F}_B$, as illustrated in Fig. 7. An overview of the vision-based tracking pipeline in the experimental setup is shown in Fig. 6.

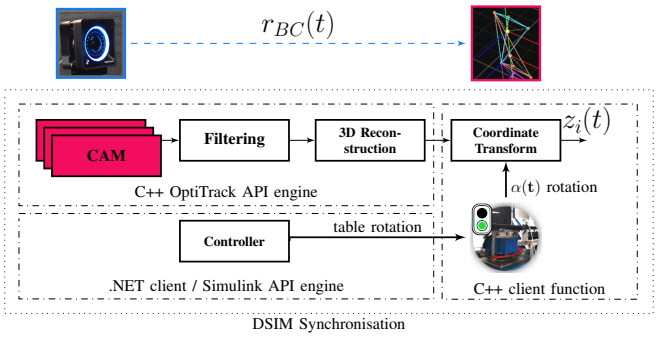



Fig. 6. Real-time vision-based tracking pipeline with continuous morphing wing turntable rotations.

*2) 3D Reconstruction:* The n-view 3D reconstruction problem is concerned with finding the optimal estimation of an object $\hat{X}$ in a 3D global coordinate frame (i.e., locations in the $x$, $y$ and $z$-axes), which is observable in noisy $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ points correspondence in $n$ camera views. The point correspondences $\bar{x}_i$ are generally defined by markers in $u, v$ coordinates of a 2D image plane and transformed to a camera

fixed reference frame, $\mathcal{F}_C$ via triangulation [37]. Prior to this process, several successive image processing steps are implemented to refine observation of point correspondences. The most common setup for triangulation is a calibrated 2-view stereo camera setup, which was demonstrated in a previous study for reconstruction of flexible wing motion [24]. However, the tracking accuracy and redundancy can be improved with the addition of more camera observations ($> 2$). As highlighted in [24], this is particularly beneficial for objects subject to adverse environmental conditions in the wind tunnel (flow conditions and mechanical vibrations), where calibration drift can be accumulated over time. Therefore, a five-camera setup was used in this study.

The principle of n-view reconstruction relies on back-projecting the 3D point onto the respective camera views, allowing to define a minimization problem for the re-projection error, $E = \sum_{k=1}^{n} ||x_k - \bar{x}_k||^2$. The n-view minimization problem is commonly solved by an expanded linear system of equations similar to singular value decomposition (SVD) in a Direct Linear Transform (DLT) procedure [38]. Global optimization methods can be applied, such as, algebraic, matrix inequality and the $\mathcal{L}_\infty$ approach [39]. More computationally intensive methods are applied, such as bundle adjustment, when the camera calibration parameters are not known a priori and included in the minimization problem [37], [38]. This study does not focus on the development of a particular n-view triangulation method. In our setup, a proprietary 3D point cloud reconstruction engine is used by OptiTrack API system [40] in real time. Multi-camera calibration is performed by wandering process, resulting in an average calibration error of 0.25 mm for all cameras. The accuracy of a similar setup has been verified in [41].

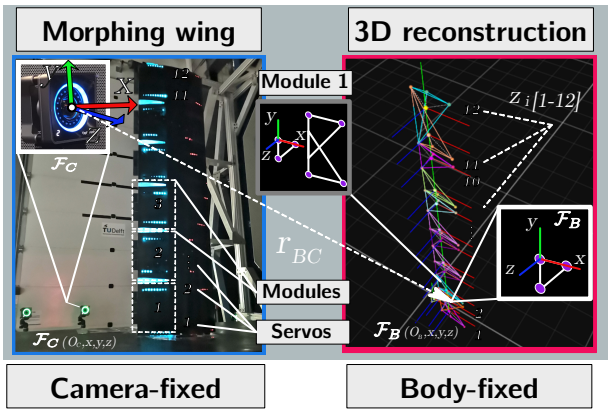



Fig. 7. Experimental setup of the vision based-control system.

*3) Coordinate Transformation:* To arrive at the desired control input for the controller, a final step is performed in the 3D reconstruction: a coordinate system transformation from the system, $\mathcal{F}_C$ with an origin $O_C$, to the frame, $\mathcal{F}_B$ with an origin $O_B$ as illustrated in the 3D view of Fig. 8. Here, the coordinate frames and their respective origins, located at approximately 2 meters away, are connected by a vector $r_{BC}$. The transformation $\mathcal{F}_C \rightarrow \mathcal{F}_B$ is performed by a successive translation, followed by 3-axis rotations in pitch, roll, and yaw axes ($\theta$,$\phi$,$\psi$). Due to continuous rotation of the turntable and thus frame $\mathcal{F}_B$ attached to it (Fig. 8), the transformations are performed continuously in real-time. The applications for processing, reconstructing and accessing the data are written in low level C++ programming language for best performance. The average total processing latency of the complete processing pipeline as shown in Fig. 6, was found to be in the range of 5-7 ms, which is smaller than the control sampling interval (16.67 ms).

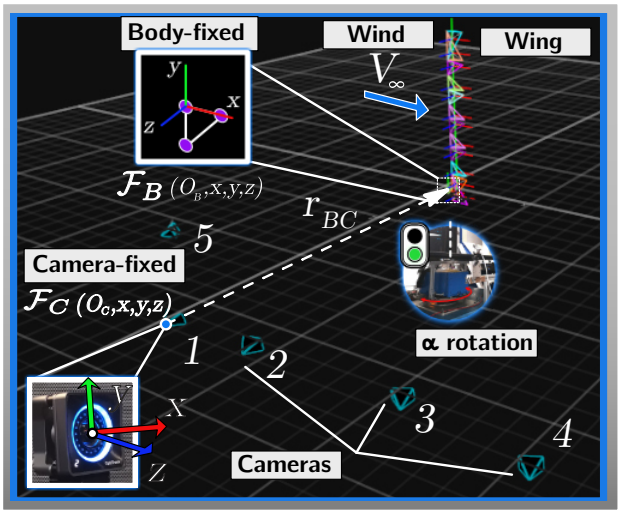



Fig. 8. 3D orientation of the cameras with respect to the morphing wing positioned in the wind tunnel.

### D. Online Shape Compensation

In experimental valuations for the morphing wing design, nonlinear backlash was observed which limited the range of travel of the morphing trailing edge and maximum attainable positive or negative camber and thus the desired shape. Owing to the backlash, the output of the morphing mechanism, which is the trailing edge displacements $z_i$, not only depends on the actuator inputs at the current time instant but is also determined by the actuation history, leading to an undesirable hysteresis phenomenon [42].

To compensate for backlash, we proposed a vision-based adaptive nonlinear incremental control approach. A data-driven incremental model is constructed online based on the input and output data. This identified model is then used by a nonlinear dynamic inversion controller for shape command tracking. Compared to a controller that only uses a feedforward input–output mapping, this approach decreases the tracking errors by more than 62% in spite of external disturbances. In view of its effectiveness, this control approach is applied for shape realization in current research.

### E. Aerodynamic Force Transformation

As the force balance is mounted to the turntable, it rotates with the turntable. Consequently, all measured forces are in the body-fixed reference frame $\mathcal{F}_B$, shown in Fig. 8. The lift and drag forces are defined as the components of the total

aerodynamic force perpendicular and parallel to the relative wind. Therefore, the measured axial $F_x$ and normal $F_y$ forces must be transformed into the aerodynamic reference frame to obtain the lift and drag forces as follows:

$$\begin{bmatrix} \mathrm{L} \\ \mathrm{D} \end{bmatrix} = \begin{bmatrix} \sin\alpha & \cos\alpha \\ -\cos\alpha & \sin\alpha \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{14}$$

Note in Eq. (14) that the positive axial force direction is defined into the relative wind, rather than along the relative wind direction. The lift and drag coefficients $C_{\mathrm{L}}, C_{\mathrm{D}}$ are defined as $C_{\mathrm{L}} \equiv \frac{L}{q_\infty S}$ and $C_{\mathrm{D}} \equiv \frac{D}{q_\infty S}$, where $S$ is the wing area and $q_\infty$ the dynamic pressure.

## IV. Experimental Design and Setup

To assess the performance of the proposed online shape optimization architecture, an experiment was conducted with the SmartX-Alpha at the Open Jet Facility (OJF) located at the Aerospace Engineering Faculty of the Delft University of Technology.

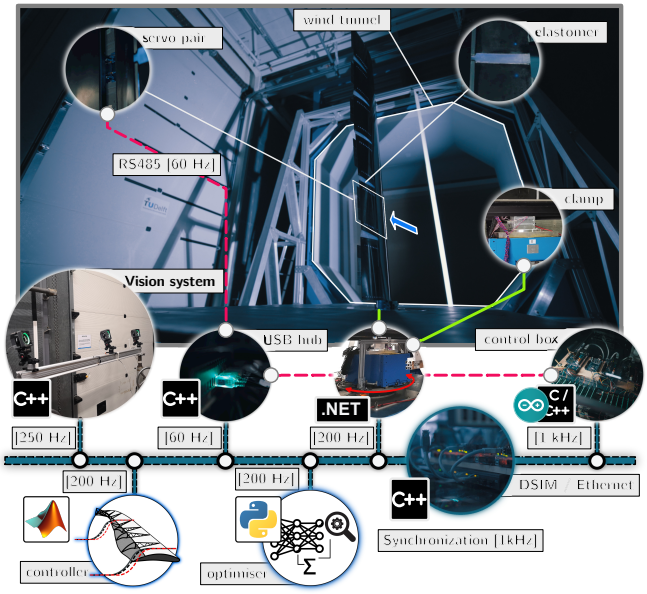



Fig. 9. Experimental apparatus with various hardware, software, and vision-based control components. The green, red, and blue lines represent mechanical, electrical, and synchronization paths, respectively.

*1) Apparatus:* The experimental apparatus is shown in Fig. 9. The system consists of the SmartX-Alpha wing, mounted vertically on an actively controlled turntable system, and placed in front of the wind tunnel test section. The operational point was selected as $V = 15$ m/s. The wing is clamped on a three-axis external balance measurements system, allowing to measure root reaction forces and moments at 1000 Hz. The shape command and the resulting lift distribution are achieved by twelve high-performance servos, connected to an array of RS-485 devices communicating serially via the RS-485 protocol. The update rate is constrained by the physical USB host interface with a fixed time delay of 15 ms. The actuation angles of the servo are constrained to $\pm 25^\circ$ as to not exceed the physical limits of the morphing system.

*2) Real-time Synchronization:* To command the experimental system, various hardware and software components are needed to cooperate coherently and share data in real-time. To facilitate this, a distributed data-sharing architecture was developed based on the decentralized communication principle, which allows parallel integration of hardware and software components in various programming languages (Python, Matlab, Simulink, C++, .NET, etc.) and various communication protocols (RS485, Ethernet, ModBus). The architecture software is developed in C++ with the real-time D-SIM framework, connecting several PC nodes over a local Ethernet network [43]. D-SIM is a software framework, written in C++, that facilitates the synchronization of real-time variables (e.g., $F_x, F_y, z_i$) over a distributed Ethernet network. The resulting synchronization architecture allowed synchronization depicted in the bottom part of Fig. 9. This approach provides several key benefits over conventional centralized systems: (i) running hardware and software processes in parallel at non-uniform sampling rates with a 1 kHz synchronization of shared variables between processes; (ii) mixing various programming languages and protocols for various experimental components; (iii) scalability and easy to modify system structure.

*3) Vision-based Shape Control:* A vision system combined with a nonlinear compensator is responsible for ensuring that the desired commanded shape is achieved globally. An array of IR-LEDs of type 3528 850NM WLP PLCC2, characterized by 850 nm wavelength is installed on the wing bottom surface and powered by a 2V direct current (DC) power supply. The brightness is actively controlled by the IRF520 Power metal–oxide–semiconductor field-effect transistor (MOSFET) dimmer circuit. Five Prime$^x$41 4.1 megapixel IR cameras are responsible for marker tracking at a frame rate of 250 frames per second (FPS) [40]. The shape-reconstruction algorithm is written in C++ and deployed on Dell Optiplex 7400 desktop system.

*4) Turning Table:* In order to continuously control the angle of attack of the wing, a real-time control command loop has been implemented. The Franke turning table of type LTB 400 is equipped with a brushless TC-60-1.3 1.3 Nm motor with encoder and breaking system. The table angle is measured by an MSR 40 MOR rotary encoder. The servo is controlled by a proportional–integral–derivative algorithm whose parameters are tuned to provide smooth table angle command tracking while satisfying servo rate and position limits. The servo commands are communicated via RS232 protocol over the USB controller. A .NET-based software control interface is developed to set control parameters and received the encoder feedback signal, which is interfaced to the synchronization framework in real-time at 200Hz.

A control flow timeline is shown in Fig. 10, responsible for the operation of various system components during the measurement phase. The order and measurement conditions of the performed runs are shown in Tab. I. The three types of runs performed are baseline, wandering, and optimization. The baseline runs are AOA sweep with a fixed (jig) wing shape, to establish a performance baseline. During the wandering phase runs, pseudo-random (PR) inputs were actuated on the system to explore the input space for onboard model

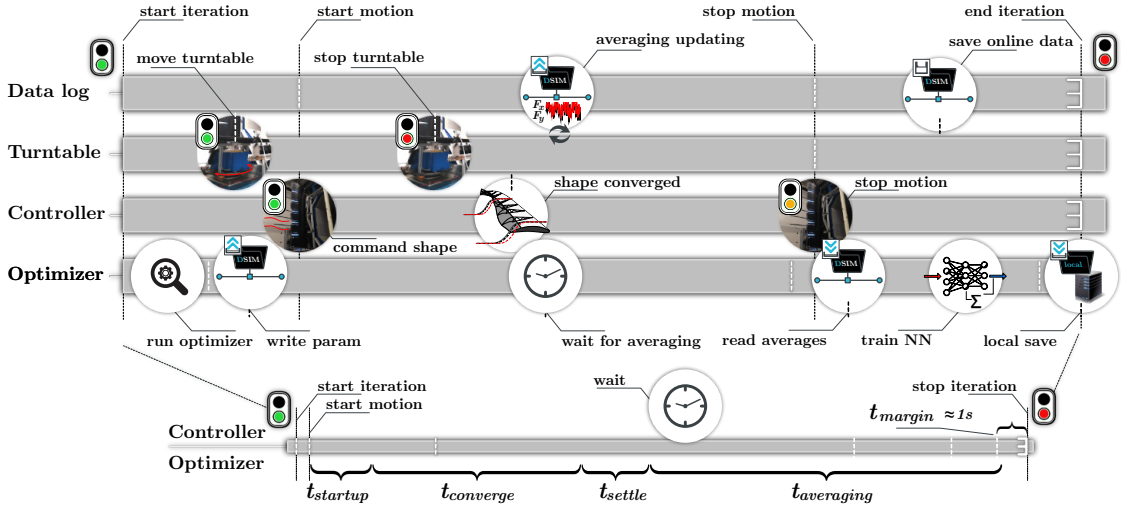



Fig. 10. Control flow and timeline of the experimental system.

TABLE I
EXPERIMENT TEST MATRIX, MEASUREMENT WITH ANGLE-OF-ATTACK BIAS IN GRAY (PR = PSEUDO-RANDOM, PO = PERCEIVED OPTIMAL).

| $C_{L_t}$ | $\alpha$ | wing shapes | type | no. samples |
|---|---|---|---|---|
| - | sweep | jig shape | baseline | 18 |
| - | PR | PR | wandering | 150 |
| 0.65 | PO | PO | optimization | 15 |
| - | sweep | jig shape | baseline | 18 |
| - | PR | PR, reduced bounds | wandering | 57 |
| 0.40 | PO | PO | optimization | 30 |
| 0.75 | PO | PO | optimization | 30 |
| 0.90 | PO | PO | optimization | 40 |

identification. Throughout the optimization runs, the optimal angles of attack and wing shapes established by the optimizer were commanded to maximize the lift-to-drag ratio.

After the first optimization run, the zero angle of the turntable was reset by a necessary reconfiguration of the turntable hardware and software as part of the shut-down and start-up procedure. The shift in the turntable zero position was approximated by matching the force balance readings to earlier established benchmarks through variation of the table angle. However, later analysis revealed this approximation method to be less accurate than was supposed at that time. Post-processing of the measurements using Algorithm 2 showed that the shift in the table angle zero position was overestimated by approximately 1.6 degrees. As a result, the true angles of attack, and through the angle-of-attack-dependent transformation, the lift and drag coefficients of the second wandering phase were unintentionally biased. These biases were found to negatively affect the performance of the later optimization runs. These known biases were corrected in a later post-processing, yielding the correct wing shape predictions.

## V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, the experimental results are discussed. First, a performance baseline is established, and the measurements from the wandering phase are presented in subsection V-A. Then, in subsection V-B, the results from online optimization for a single fixed target lift coefficient of 0.65 are elaborated upon. Finally, a post-experiment simulation using experimental data is used to make a more general prediction about the potential drag reductions achievable at a wider range of target lift coefficients in Sec. V-C.

### A. Baseline and Wandering Phase

A performance baseline was established by measuring the aerodynamic forces of the wing jig shape at various angles of attack. This wing jig shape was realized by performing the shake maneuver (simultaneous actuation sequence of all modules from [25,-25,0]) without wind. Subsequently, forty-second averaged force measurements were taken at table angles from $-18$ to 10 degrees in increments of two degrees at a wind speed of 15 m/s. Since the zero position of the turntable did not coincide with a zero AOA, the true aerodynamic angles of attack of these measurements were unknown. The angular difference $\varepsilon$ between the commanded turntable angle $\psi$ and the aerodynamic angle of attack $\alpha$, was estimated using the iterative approach outlined in Algorithm 2. During this procedure, the turntable misalignment constant $\varepsilon$ is iteratively estimated by matching the measurement-based estimated zero-lift angle of attack $\hat{\alpha}_{C_L=0}$ to the theoretical zero-lift angle of attack $\alpha^*_{C_L=0}$. This theoretical zero-lift angle of attack was determined to be $-6.5570$ degrees, with an evaluation of the Smart-X-Alpha geometry using the 3D panel viscous solver *XFLR5 v 6.48*.

The angular misalignment was estimated to be $\varepsilon = -9.5789$ degrees. This value was used to actuate the turntable based on the desired aerodynamic angle of attack during subsequent measurements.

The performance baseline was established by interpolation of the jig shape measurements with a $16^{\text{th}}$ degree polynomial. This relatively high model order was selected because single low order polynomials represent a poor fit of the performance curve which is linear for only one section of the domain. The validity of this interpolation in the region of interest was not

**Algorithm 2:** Table angle correction estimation.

**Input:** $\psi$,$F_x$, $F_y$, $\alpha^*_{C_L=0}$
**Output:** $\varepsilon$
**Function** `calcAlpha` $(\psi, \varepsilon)$**:**
  $\alpha = \psi + \varepsilon$
**Function** `estAlpha0` $(\alpha, F_x, F_y)$**:**
  Transform $F_\mathrm{x}, F_\mathrm{y} \rightarrow C_\mathrm{L}, C_\mathrm{D}$;
  Fit $1^{\mathrm{st}}$ order polynomial;
  Estimate zero lift angle of attack $\hat{\alpha}_{C_L=0}$;
**Function** `updateError` $(\hat{\alpha}_{C_L=0})$**:**
  $\epsilon = \hat{\alpha}_{C_L=0} - \alpha^*_{C_L=0}$;
**while** $\epsilon < 1 \times 10^{-8}$ **do**
  Evaluate $\varepsilon = \varepsilon + 0.5 \cdot \epsilon$;
  `/* output cascaded through functions */`
  `updateError (estAlpha0 (calcAlpha));`
**end**

compromised by Runge's phenomenon [44] as the bounds of the jig sweep measurements are much wider than the region of interest. Both the jig shape measurements and the fitted model, within the region of interest, are shown in Fig. 11.

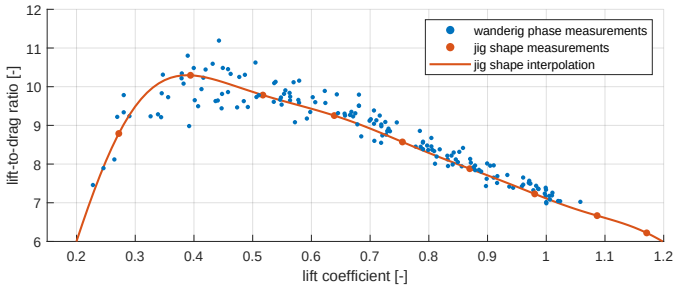



Fig. 11. Wing jig shape baseline performance measurements (orange dots), wing jig shape performance fitted model (orange line), and performance measurements from pseudo-random exploration (blue).

During the wandering phase, where the actuation space was explored with pseudo-random inputs. Out of the 191 inputs evaluated, the last 57 were generated with reduced bounds on the virtual shape inputs because the hitherto evaluated shapes frequently led to saturation for the majority of the morphing servos. These last measurements were performed after the angle-of-attack bias described in Tab. I was introduced.

Figure 11 shows the performance of the inputs explored in the wandering phase. Naturally, the performances of the pseudo-random wandering inputs are distributed around the jig shape performance curve. While the jig shape curve serves as a baseline, the distribution of the wandering phase performances roughly indicates the physical bounds of the attainable performance with active morphing for SmartX-Alpha.

### *B. Online Optimization*

After the first 150 wandering phase measurements with nominal bounds were taken, 15 iterations of online wing shape optimization were performed with a target lift coefficient of $C_{\mathrm{L_t}} = 0.65$.

The lift and drag coefficients measured during the online optimization experiment are shown in Fig. 12. During the first six iterations of online optimization, the discrepancies between the targeted and the measured lift coefficients are relatively large: $\pm 0.25$. At the same time, the measured drag coefficients also fluctuate considerably. This is to be expected, as the total drag is dominated by the lift-induced drag contribution.

Furthermore, wing shapes evaluated initially appear suboptimal, attributed to the observation of increased amounts of camber near the root and tip. While the instantaneous optimal shape is not known exactly, it should be expected to follow an elliptical distribution, with decreased amounts of camber at those locations. Note that for a rectangular planform and no wash out angle, a decreasing (less camber) twist distribution would result in an elliptical distribution, hence reduced wing tip vortices and induced drag [45]. Due to pressure leakage at the root interface, a smaller amount of spanwise camber reduction is expected.

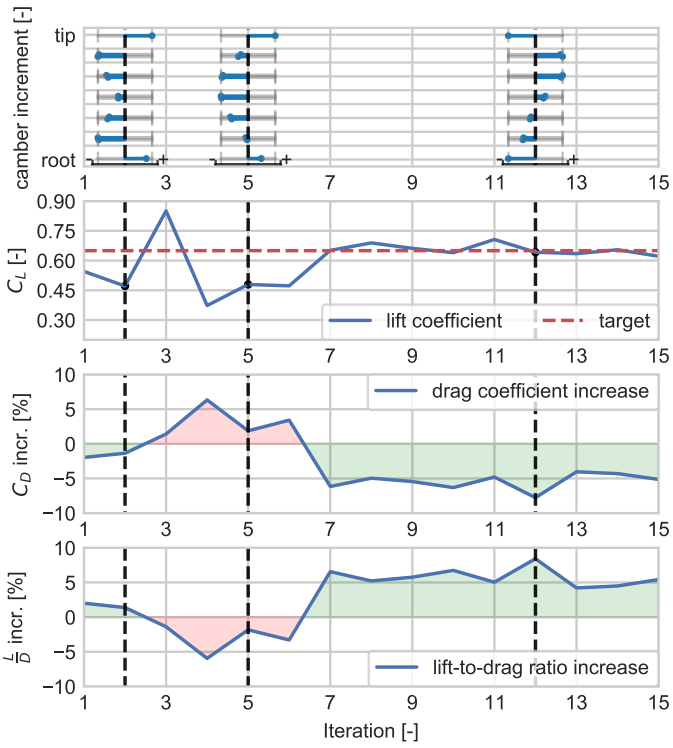



Fig. 12. Measured lift and drag coefficients, and selected wing shapes from online optimization with a target lift coefficient of 0.65 for 15 iterations.

After iteration six, the measured lift coefficient approaches the target lift coefficient. The largest measured deviation from the target lift coefficient was 0.056, which was measured at iteration 11. At the same time, the increase in drag coefficient relative to the drag coefficient of the jig shape at the same lift coefficient decreases below zero. This means that the wing shape and angle of attack combinations evaluated from iteration seven onward not only realized a lift coefficient close to the target, but did so with lower drag than the wing jig shape as well. Consequently, the relative increase in the lift-to-drag ratio is also positive for these iterations.

The most desirable performance was measured for the input combination evaluated during iteration 12. The measured lift coefficient during this iteration was 0.642. The measured lift-to-drag ratio at this iteration was 10.015. This corresponds to an 8.4 % increase in lift-to-drag ratio compared to the jig shape performance at this lift coefficient. This is equivalent to a 7.8 % drag reduction compared to the wing jig shape at the same lift coefficient.

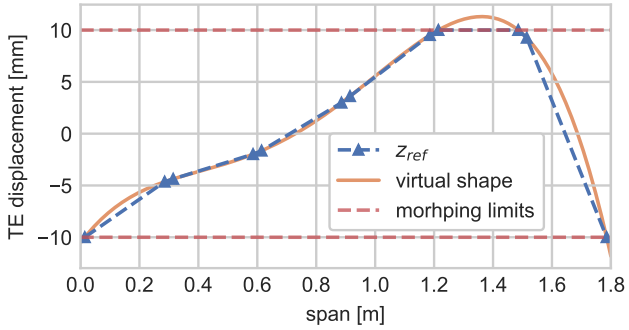



Fig. 13. Shape evaluated at iteration 12 during online optimization with a target lift coefficient of 0.65.

The wing shape evaluated at iteration 12 is shown in Fig 13. This shape comprises maximum positive camber between 1.4 and 1.6 m from the wing root, which corresponds to module 5, a steep decrease at the wing tip, and a gradual reduction of the local camber towards the root end of the wing. It is mainly the reduction of the local camber near the wing ends, which is supposed to reduce the strength of the wing tip vortices, that results in the observed improvement in the aerodynamic efficiency. These vortices are the result of the spanwise flow components caused by the "leaking" of high pressure air from the bottom side of the wing around the wing tips towards the lower pressure regions on the upper wing surface [45]. While the wing-root-fuselage interfaces of aircraft usually do not allow this kind of pressure leakage, the interface between the wing root and flow table in our set up was not airtight. An approximately 1 cm wide clearance gap existed between the wing root and the flow table, which was needed to allow free rotation of the wing as the flow table was not part of the assembly rotated by the turntable. It is also noteworthy that the commanded shape shown in Fig. 13 differs from the realized wing shape where the commanded amount of differential TE displacement within one module exceeds the twist morphing limit of the TRIC module.

Nevertheless, the optimal location of the maximum camber is expected to be in between the wing root and the center of the wing, which corresponds to an interpolation between the elliptical distributions associated with wings with one and two free ends. Hence, it is thought that even though the given shape already offers a 7.8 % drag reduction over the wing jig shape for this target lift coefficient, even more efficient wing shapes do exist. An advanced efficient wing shape is made using the measurements from the second wandering phase, which were bias-corrected in post-experimental processing. These results are discussed in the following section.

### C. Experimental Data-driven Optimization Predictions

To improved estimation of the optimal shape and the corresponding drag reduction and sample from more target lift coefficients, the online training was simulated using experimental data collected during wandering phase. Additionally, the following post-processing was performed on the wandering phase measurements:

1) The samples measured after the angle-of-attack bias was introduced in the turntable, were corrected.
2) Samples whose tracking signals did not fully converge within the measurement window, or were not recorded, were omitted.

After post-processing, 25 % of the samples were reserved for validation. The remaining samples were fed to the optimization algorithm on a per-sample basis to simulate the wandering phase experiment. The validation set samples were used to estimate the predictive accuracy of the trained model.

With a target lift coefficient of 0.65, the optimal wing shape as computed by the optimizer on the trained onboard model, was as shown in Fig. 14. With an angle of attack of 0.8 degrees, the predicted lift-to-drag ratio of this shape was 10.35. This corresponds to a predicted aerodynamic efficiency improvement of 11.1% compared to the wing jig shape. Furthermore, the shape shown quite closely represents the expected optimal shape described in Sec. V-B, with a gradual reduction of airfoil camber towards the free wing tip end, and a more moderate amount of camber reduction at the root end as a result of pressure leakage at the root intersection.

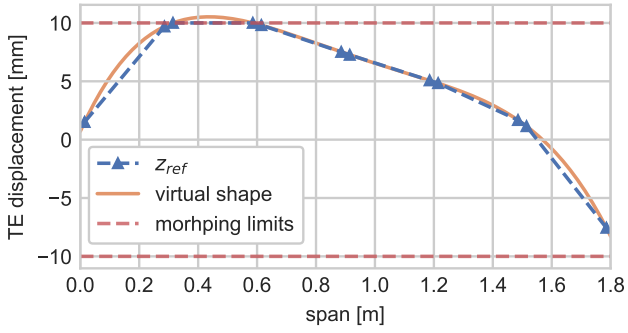



Fig. 14. Post-experiment simulation based optimal shape for $C_{\mathrm{L_t}} = 0.65$.

The predictive accuracy of the trained onboard model was evaluated using error measures computed on the validation data set. The standard deviations of the prediction errors, Root Mean Square error (RMSE), and their normalized counterparts are shown in Tab. II. The Normalized Root Mean Square Error (NRMSE) of both the lift and drag coefficient RBFNN models are close to each other. The NRMSE values were normalized using the domain width of the corresponding output variables, i.e. NRMSE $=$ RMSE$/(y_{\max} - y_{\min})$. Since the lift-to-drag ratio is not approximated with a dedicated neural network, but by the ratio of two estimated outputs, the lift-to-drag ratio NRMSE is higher. In conclusion, based on the relatively small training set data, the onboard model is able to predict the lift and drag coefficients of the validation data set samples with an

TABLE II
MODEL ERROR MEASURES ON VALIDATION DATA SET.

| | RMSE [-] | NRMSE [%] |
|---|---|---|
| $C_\mathrm{L}$ | 0.0147 | 2.34 |
| $C_\mathrm{D}$ | 0.0026 | 2.50 |
| L/D | 0.2818 | 7.75 |

average prediction error that is approximately 2.5 % of their respective domain widths.

In order to investigate the trend in the estimated optimal wing shape for different target lift coefficients, the shape optimization was also conducted for target lift coefficients of 0.35, 0.50, and 0.80. The predicted optimal shapes for these lift coefficients are shown in Fig. 15. In general, the maximum amount camber morphing is commanded at approximately one-quarter span, with a gradual reduction approaching the minimum camber limit towards the wing tip. Additionally, a more modest reduction of camber is observed towards the wing root end. For higher target lift coefficients, the area under the virtual shape curves, which can be thought of as the overall amount of camber, is increased.

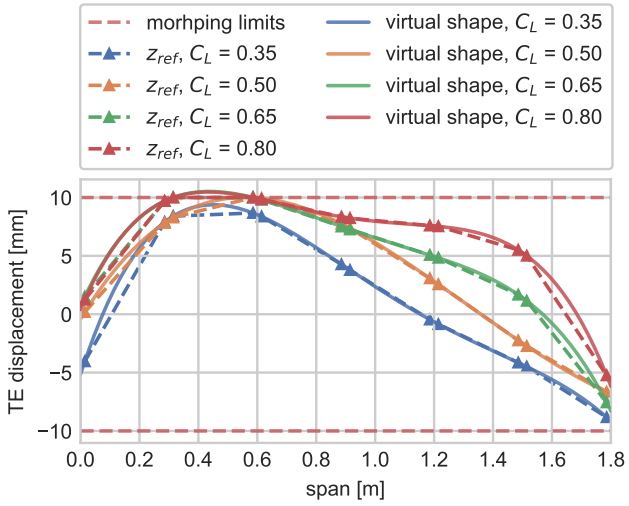



Fig. 15. Post-experiment simulation based optimal shapes for four target lift coefficients.

For active camber morphing airfoils, the amount of lift generated can be changed with changes of both the angle of attack and the airfoil camber. Hence, multiple combinations of the angle of attack and camber morphing can be employed to realize any given target lift coefficient. However, due to the underlying aerodynamics, these different solutions will not necessarily come with the same drag penalties. Computational Fluid Dynamics (CFD) simulations for active camber morphing airfoils have shown that the trailing edge deflection of camber morphing airfoils for best lift-to-drag ratio increases with increasing lift coefficients [46]. In the case of a three-dimensional distributed active camber morphing wing, target lift coefficients can be realized with different combinations of the angle of attack and the spanwise camber distribution. Earlier wind tunnel experiments on SmartX-Alpha have shown that for constant spanwise camber morphing, the optimal amount of trailing edge displacement also increases with increasing lift coefficients. Therefore, the trend of increasing overall camber of the predicted optimal distributions with increasing target lift coefficients conforms to the expectations.

As long as the morphing limits are not reached, both the best overall amount of camber and the ideal spanwise lift distribution, which is elliptical for induced drag reduction, can be achieved simultaneously. Comparing the camber distributions for $C_\mathrm{L} = 0.35$ and $C_\mathrm{L} = 0.50$, the largest difference between the two distributions is an an increased, nearly linear camber offset along the entire wing span. For the $C_\mathrm{L} = 0.50$ distribution, the fourth and fifth servo units have already reached their upper limits. Unlike previous shapes, no linear offset is observed, rather a redistribution starting from the fifth servo unit. This observation can be explained by the conflict between larger camber needed for higher lift coefficients and desire to maintain ideally shaped spanwise lift distribution, as the morphing saturation limit is reached locally. The needed target lift increment is compensated by the remaining servos (i.e. where increasing camber is still possible) at the cost less ideal distribution. It can be noted that, as long as the area under the curve is comparable, alternative shape are possible, potentially yielding a better tradeoff in terms of total drag.

TABLE III
ANGLES OF ATTACK AND AERODYNAMIC COEFFICIENTS CORRESPONDING TO PREDICTED OPTIMAL WING SHAPES.

| $C_\mathrm{L}$ [-] | $C_\mathrm{D}$ [-] | $\frac{\mathrm{L}}{\mathrm{D}}$ [-] | $\alpha$ [deg] | $C_\mathrm{D}$ reduction [%] |
|---|---|---|---|---|
| 0.35 | 0.02762 | 12.67 | -2.50 | 19.8 |
| 0.50 | 0.04188 | 11.94 | -1.05 | 17.4 |
| 0.65 | 0.06282 | 10.35 | 0.81 | 11.1 |
| 0.80 | 0.09023 | 8.87 | 2.97 | 6.5 |

The angles of attack, and lift and drag coefficients of the predicted optimal shapes are given by Tab. III. It can be seen that over 6.5 % of drag reduction has been successfully achieved for all the tested target lift coefficients.For the lower lift coefficients, the associated drag is also lower, meaning that the relative error of the drag coefficient prediction becomes larger, and the maximum lift-to-drag ratio may be optimistic. However, with more data, more accurate neural network models could be identified in the future such that more accurate predictions can be made on the highest achievable lift-to-drag ratios. For operational online shape optimization, the exact predicted outputs are of lesser importance as long as the optimal inputs are predicted correctly.

## VI. CONCLUSIONS AND RECOMMENDATIONS

A strategy for online black-box shape optimization for active distributed camber morphing wings has been proposed and tested on a seamless distributed camber morphing wing demonstrator known as SmartX-Alpha in an open jet wind tunnel. The proposed online optimization method combines a global onboard RBFNN model with an evolutionary optimization strategy to find the wing-shape and angle-of-attack combination that maximizes the aerodynamic efficiency at

commanded targeted lift coefficient. Compared to the unmorphed NACA6510 airfoil base shape, a drag reduction of $7.8\%$ was achieved on SmartX-Alpha for a target lift coefficient of 0.65.

To assess more target lift coefficient inputs, additional offline optimization was performed on the experimental data. The analysis showed that under the presence of morphing limit saturation of, the optimizer accepted a deviation from the ideal lift distribution (nearly linear twist distribution towards the tip) and compensated with the remaining servos to meet the higher target lift coefficients. Drag reductions were observed between $6.5$ % and $19.8$ %, with the highest corresponding to lower lift coefficients. Due to relatively larger drag coefficient error, the maximum predicted drag reduction are believed to be optimistic.

A significant benefit of the proposed optimization framework lies in the ability to operate online, eliminating the need for model excitation maneuvers at every trim condition, such as is the case with existing online gray-box methods employing local models. Since the method is inherently black-box, it is independent of the system, easy to adapt to other morphing wing platforms. To further increase the adaptability and accuracy of the framework, it is suggested to expand the input space of the model with additional lift/drag dependent inputs, such as Reynolds number and Mach number.

For the application of the optimization framework to real-world commercial aircraft, no further changes to the optimization architecture are needed. To extract the aerodynamic forces, an alternative must be used for the force-balance. Possible alternatives are strain gauges, fiber optics or a combination thereof [47]. Additionally, model based load estimation from other measurable parameters such as, angle of attack, flight speed, altitude and fuel flow are possible. Lastly, our experimental IR cameras set-up can be converted to a fuselage-mounted in-flight vision tracking system as has been experimentally validated in [48]. With these changes in the sensing equipment, the proposed optimization architecture can be applied to real-world aircraft in the future.

## REFERENCES


[1] T. A. Weisshaar, "Morphing aircraft systems: Historical perspectives and future challenges," *Journal of Aircraft*, vol. 50, no. 2, pp. 337–353, mar 2013.

[2] H. R. Jex and F. E. Culick, "Flight Control Dynamics of the 1903 Wright Flyer." *AIAA Paper*, pp. 534–548, 1985.

[3] D. Li, S. Zhao, A. Da Ronch, J. Xiang, J. Drofelnik, Y. Li, L. Zhang, Y. Wu, M. Kintscher, H. P. Monner, A. Rudenko, S. Guo, W. Yin, J. Kirn, S. Storm, and R. D. Breuker, "A review of modelling and analysis of morphing wings," *Progress in Aerospace Sciences*, vol. 100, no. June, pp. 46–62, 2018.

[4] D. Lentink, U. K. Müller, E. J. Stamhuis, R. De Kat, W. Van Gestel, L. L. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. Van Leeuwen, "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, no. 7139, pp. 1082–1085, 2007.

[5] M. Jacobsen, "Real time drag minimization using redundant control surfaces," *Aerospace Science and Technology*, vol. 10, no. 7, pp. 574–580, 2006.

[6] Y. Ferrier, N. Nguyen, and E. Ting, "Real-time adaptive least-squares drag minimization for performance adaptive aeroelastic wing," in *34th AIAA Applied Aerodynamics Conference*, Washington, D.C., 2016.

[7] N. Nguyen, S. Lebofsky, E. Ting, U. Kaul, D. Chaparro, and J. Urnes, "Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing," *SAE Technical Papers*, vol. 2015-Sep, no. September, 2015.

[8] N. Nguyen, N. B. Cramer, K. E. Hashemi, E. Ting, M. Drew, R. Wise, J. Boskovic, N. Precup, T. Mundt, and E. Livne, "Real-Time Adaptive Drag Minimization Wind Tunnel Investigation of a Flexible Wing with Variable Camber Continuous Trailing Edge Flap System," in *AIAA Aviation 2019 Forum*. Dallas, Texas: AIAA, jun 2019.

[9] N. Nguyen and J. Xiong, "Real-time drag optimization of aspect ratio 13.5 common research model with distributed flap system," in *AIAA Scitech 2021 Forum*. virtual event: AIAA, January 2021.

[10] J. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. 233 Spring Street, New York, NY 10013, USA: (Springer Science+Business Media, Inc., 2005, p. 24.

[11] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[12] W. Gong, A. Zhou, and Z. Cai, "A Multioperator Search Strategy Based on Cheap Surrogate Models for Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, 2015.

[13] J. L. Chávez-Hurtado and J. E. Rayas-Sánchez, "Polynomial-based surrogate modeling of rf and microwave circuits in frequency domain exploiting the multinomial theorem," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 12, pp. 4371–4381, 2016.

[14] M. Tahkola, J. Keränen, D. Sedov, M. F. Far, and J. Kortelainen, "Surrogate modeling of electrical machine torque using artificial neural networks," *IEEE Access*, vol. 8, pp. 220 027–220 045, 2020.

[15] W. Zhang, F. Feng, J. Jin, and Q.-J. Zhang, "Parallel multiphysics optimization for microwave devices exploiting neural network surrogate," *IEEE Microwave and Wireless Components Letters*, vol. 31, no. 4, pp. 341–344, 2021.

[16] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.

[17] B. Liu, Q. Zhang, and G. G. E. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[18] J. Luo, A. Gupta, Y.-S. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1708–1721, 2019.

[19] E. Iuliano and D. Quagliarella, "Efficient aerodynamic optimization of a very light jet aircraft using evolutionary algorithms and rans flow models," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–10.

[20] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[21] C. Kim, "Computational elements for high-fidelity aerodynamic analysis and design optimisation," *Defence Science Journal*, vol. 60, no. 6, pp. 628–638, 2010.

[22] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 1, pp. 66–76, 2007.

[23] J.-W. Park, G. Kumar Venayagamoorthy, and R. G. Harley, "MLP/RBF Neural-Networks-Based Online Global Model Identification of Synchronous Generator," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 6, 2005.

[24] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive real-time clustering method for dynamic visual tracking of very flexible wings," *Journal of Aerospace Information Systems*, vol. 18, no. 2, pp. 58–79, 2021.

[25] E. Pan, X. Liang, and W. Xu, "Development of Vision Stabilizing System for a Large-Scale Flapping-Wing Robotic Bird," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 8017–8028, jul 2020.

[26] S. Tijmons, G. C. De Croon, B. D. Remes, C. De Wagter, and M. Mulder, "Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 858–874, aug 2017.

[27] P. Serra, R. Cunha, T. Hamel, C. Silvestre, and F. Le Bras, "Nonlinear image-based visual servo controller for the flare maneuver of fixed-wing aircraft using optical flow," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 570–583, mar.

[28] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a smart morphing wing using integrated and distributed trailing edge camber morphing," in *Smart Materials, Adaptive Structures and*

*Intelligent Systems*, vol. 84027. American Society of Mechanical Engineers, 2020, p. V001T04A023.
[29] R. Hamming, *Numerical Methods for Scientists and Engineers, second edition*. 1221 Avenue of the Americas, New York,USA: McGraw-Hill, Inc., 1973, p. 470.
[30] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies." *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
[31] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–102.
[32] M. J. D. Powell, *Radial Basis Functions for Multivariable Interpolation: A Review*. USA: Clarendon Press, 1987, p. 143–167.
[33] C. S. K. Dash, A. K. Behera, S. Dehuri, and S. B. Cho, "Radial basis function neural networks: A topical state-of-the-art survey," *Open Computer Science*, vol. 6, no. 1, pp. 33–63, 2016.
[34] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.
[35] M. McCloskey and N. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of Learning and Motivation - Advances in Research and Theory*, vol. 24, no. C, pp. 109–165, Jan. 1989.
[36] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. New Orleans, Louisiana: American Institute of Aeronautics and Astronautics (AIAA), 2018, pp. 3302 – 3309.
[37] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
[38] K. Kanatani, Y. Sugaya, and Y. Kanazawa, "Multiview triangulation," in *Guide to 3D Vision Computation*. Springer, 2016, pp. 133–147.
[39] R. Hartley and F. Kahl, "Optimal algorithms in multiview geometry," in *Asian conference on computer vision*. Springer, 2007, pp. 13–34.
[40] OptiTrack, "OptiTrack - Primex 41 - In Depth." [Online]. Available: https://optitrack.com/cameras/primex-41/
[41] A. M. Aurand, J. S. Dufour, and W. S. Marras, "Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume," *Journal of biomechanics*, vol. 58, pp. 237–240, 2017.
[42] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 9, pp. 1649–1662, sep 2021.
[43] multiSIM, "multiSIM: Distributed simulation D-SIM," 2021. [Online]. Available: https://multisim.nl/d-sim/
[44] C. Runge, "Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten," *Zeitschrift für Mathematik und Physik*, vol. 46, p. 224–243, 1901.
[45] J. D. Anderson Jr., *Fundamentals of Aerodynamics*. 1221 Avenue of the Americas, New York,USA: McGraw-Hill, Inc., 2011, pp. 416,437.
[46] S. J. Huntley, C. B. Allen, and B. K. Woods, "Computational analysis of the aerodynamics of camber morphing," in *AIAA Aviation 2019 Forum*, no. June. Dallas, Texas: AIAA, 2019, pp. 1–20.
[47] J.-H. Kim, Y. Park, Y.-Y. Kim, P. Shrestha, and C.-G. Kim, "Aircraft health and usage monitoring system for in-flight strain measurement of a wing structure," *Smart Materials and Structures*, vol. 24, no. 10, p. 105003, 2015.
[48] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive state estimation and real-time tracking of aeroelastic wings with augmented kalman filter and kernelized correlation filter," in *AIAA Scitech 2021 Forum*, 2021, p. 0666.

# 6

# Wind tunnel experiments

In this chapter a comprehensive description of the wind tunnel experiment is presented. The aim of this chapter is not to repeat the description of the experimental set-up and the findings presented in the IEEE journal paper (chapter 5), but rather to present additional supportive material. The structure of this chapter is as follows. First, the adaptations of the online optimization framework from the configuration described in chapter 3 for use in the wind tunnel experiment are discussed in section 6.1. Next, the measurement routine and testing order are elaborated on in section 6.2. In section 6.3 the results of post turntable actuation bias optimization runs are shown. Finally, a bias correction is performed in post-processing and the corrected data is used to simulate the online wandering part of the experiment in section 6.4.

## 6.1. Framework adaptations

Compared to the simulation experiments described in chapter 3 and 4, a number of changes were made to set up for the wind tunnel experiment. Because of nonlinear backlash effects which were discovered to be present in the morphing mechanism of the SmartX-Alpha demonstrator [66], the actuator angles $\theta_1, \dots, \theta_{12}$ do not relate to the actual wing shape in a one-to-one mapping. Instead, the realized wing shape is determined not only by the actuator positions but also by their positional histories. Since the developed online optimization framework relies on the assumption that a one-to-one mapping between the system inputs and outputs exists, these nonlinear backlash effects would introduce large uncertainties that would be detrimental to the success of the method in the wind tunnel experiment. Therefore, it was decided to use the wing shape as the input to the on-board model instead of the actuator positions. Even though the actuator angles no longer have a one-to-one mapping to the wing shape because of the nonlinear backlash effects, the relationship between the aerodynamic coefficients and the actual wing shape remains unaffected. Instead of the actuator angles, the vertical trailing-edge displacements at the 12 servo locations $z_1, \dots, z_{12}$ were used as the new model inputs in the wind tunnel experiment. However, the following complications remained:

1. The true trailing-edge displacements are unknown.
2. The trailing-edge displacements can only be controlled through the morphing servos
3. The relationship between the actuators and the trailing-edge displacements is complex and nonlinear.

In order to deal with the first of these, a set-up consisting of five Infra-Red (IR) cameras was used to track the locations of 27 markers attached to the morphing and non-morphing parts of the wing. A comprehensive description of the IR camera and marker set-up and the trailing-edge displacement estimation routine is given in chapter 5. In order to compensate for the backlash effects, a nonlinear dynamic inversion controller was implemented. In this new set-up, the inputs to the on-board model, and thus the outputs of the optimization procedure were the commanded trailing-edge displacements. These commanded reference displacements constituted the reference values to which the controller

steered the displacement estimations from the IR vision system. A diagram detailing the integration of and flow of information between the optimization framework and the other software and hardware components of the experimental set-up is shown in chapter 5.

**Optimizer reconfiguration** Because the inputs to the optimization framework black-box system were changed from the actuator angles to the reference vertical trailing-edge displacements, some adaptations to the online optimization framework were required as well. The virtual inputs which used to describe a curve dictating the commanded actuator angles in degrees at the spanwise locations were adapted to describe a curve that dictates the commanded amount of vertical trailing-edge displacement in millimeters. Hence, the units and the domain of the virtual inputs were changed. The actuator angles in degrees were converted to the local vertical trailing-edge displacements in m with the empirically fitted relation in Eq. (3.2).

Table 6.1: CMA-ES scaling of the input variables where the virtual inputs describe the vertical trailing-edge displacements.

| input | scaling | $\sigma$ | $x_{\min}$ | $x_0$ | $x_{\max}$ | unit |
|---|---|---|---|---|---|---|
| $\alpha$ | 1901.85 | 3.1951 | -2.50 | 3.75 | 10.00 | deg |
| $u_1$ | 1.92 | 0.0032 | -0.014 | 0 | 0.014 | m |
| $u_2$ | 2.14 | 0.0036 | -0.014 | 0 | 0.014 | m |
| $u_3$ | 1.00 | 0.0017 | -0.014 | 0 | 0.014 | m |
| $u_4$ | 0.86 | 0.0014 | -0.014 | 0 | 0.014 | m |
| $u_5$ | 0.65 | 0.0011 | -0.014 | 0 | 0.014 | m |

Because these changes have altered the input domain, the parameters of the optimization procedure need to be updated accordingly. The new global CMA-ES step size was selected as $\sigma_0 = 0.00168$, and the specific scaling of the input variables was updated as shown in Tab. 6.1. Not only the units of the virtual input bounds were converted, but a new set of bounds was selected altogether since the input space region of interest was different from simulation cases described thus far. This will be further elaborated in the following section. The lower and upper bounds of the virtual inputs were all set to -0.014 and 0.014 m respectively. This corresponds to the minimum and maximum achievable training edge deflections for symmetric 25-degree actuator actuation according to DIC measurements. However, as the commanded trailing-edge displacement at any location is the result of a combination of the virtual inputs, these bounds do not prevent actuator saturation. Rather, they serve to constrain the optimization algorithm to only operate within the search space domain that is accurately covered by the on-board model. Without these constraints, the optimizer would be able to exploit the poor extrapolation ability of the neural network model to find extremely well-scoring solutions outside the region spanned by the training data of the on-board model. The high scores of these solutions, however, would be the result of a significant underestimation of the drag coefficient by an inaccurate model rather than a real-world optimal wing shape. Hence, these types of solutions are undesirable and the optimizer should be prevented from converging on them.

**On-board model and wandering phase** Due to the black-box nature of the on-board model, very few adaptations were necessary to make the on-board model ready for operation in the wind tunnel. Nevertheless, one important change was made that affected the region of the search space spanned by the model as mentioned in the previous paragraph.

Since artificial neural networks are very effective tools for data interpolation, but are very poor extrapolation tools, the region of the domain on which the model can be trusted in determined by the bounds of the training data. The training data for the online optimization is gathered online, during a one-time wandering phase dedicated to search space exploration and during all subsequent optimization iterations. During the optimization iterations, the sampled points will be directly determined by the most promising solutions found by the evolutionary optimizer. Hence, the points sampled during the optimization iterations are locally concentrated and are unknown beforehand. Therefore, the domain coverage of these points cannot be guaranteed. Consequently, a minimum coverage of the input space region of interest should be achieved during the exploratory wandering phase.

For the simulation experiments, the region of interest of the virtual shape functions was determined from unbounded direct optimization on the aerodynamic model. This type of optimization required

approximately 45 minutes of computational time per target lift coefficient. The measurement times in the wind tunnel setup are approximately 40 times longer than the computational time required for the evaluation of the aerodynamic model. Moreover, the limited time available in the wind tunnel facility is very precious.

Regardless, the region of interest defined for the simulation experiment should not be used for the wind tunnel experiment as that would come with the inherent assumption that the differences between the optimal shapes of the simulation model and the wing demonstrator in the experimental setup are negligible. However, this should not be expected. A notable difference between the simulation and the experimental setup is the flow situation around the wing root. While the aerodynamic model assumes a perfect wing-fuselage interface with no pressure leaks, the situation in the experimental setup is different. A close-up of the wing root in the experimental set-up is shown in Fig. 6.1.

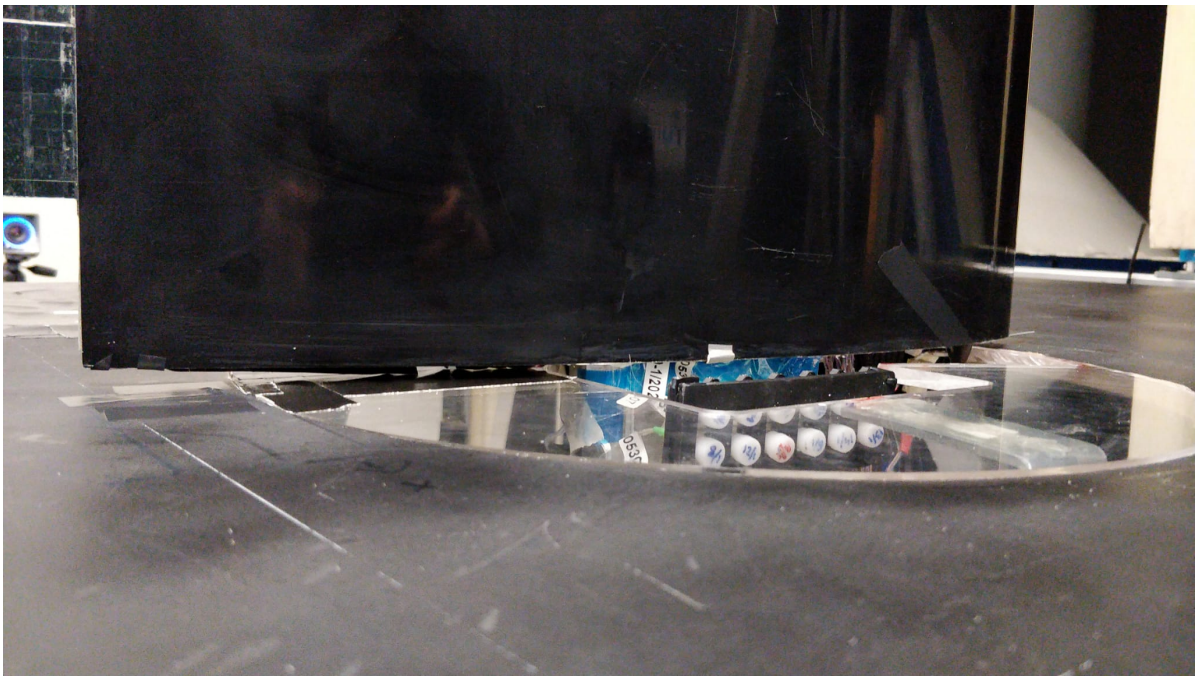


Figure 6.1: Close-up of the wing root and horizontal flow table interface during the wind tunnel experiment.

The morphing wing was mounted vertically, and the aircraft fuselage was mimicked by the horizontal surface of a table (black) with a circular insert (transparent) that rotated freely with the wing. Because the wing's chord length exceeds the diameter of the insert, the wing's leading and trailing edges extend beyond the insert and are directly over the black non-moving part of the "simulated fuselage". In order to allow the wing to rotate freely to make angle-of-attack changes, a clearance gap between the wing root and the simulated fuselage had to be maintained throughout the experiment. As a result, some degree of pressure leakage from the region of higher pressure on the wing's bottom surface to the lower pressure region at the wing's upper surface is to be expected.

In the case that no simulated fuselage would be present the root end of the wing would behave essentially a free, similar to the wing tip end. In that case, the expected optimal wing shape would be symmetrical with maximum camber occurring at the spanwise midpoint of the half wing and would elliptically decrease towards both the wing root and tip ends. The clearance gap definitely allows some degree of pressure leakage but it is not thought to be wide enough to make the flow situation around the wing root completely similar to that around a free wing end. The optimal wing shapes corresponding to this set-up are expected to be somewhere in between the described free end optimal shapes and the ideal root interface optimal shape solutions are shown in Fig. 3.13. However, the exact extent to which the clearance gap will shift the optimal shapes from ideal root end optimal shapes to free root-end optimal shapes is unknown. Therefore, it was decided to select the widest possible bounds on the virtual inputs for the wandering phase such that any shape achievable on the morphing wing platform would be within the scope of the on-board model. While this wider training data set domain extends

the scope of the on-board model, it is at the price of a lower data resolution. Hence, a trade-off is made between model accuracy and model scope.

During the wind tunnel experiments, two wandering phases were performed. The inputs actuated during these wandering phases were sampled with six-dimensional Sobol sequences as described in section 3.4.2. During the first wandering phase, the sampled inputs spanned the domain bounded by the variable bounds shown in Tab. 6.1. The minimum and maximum local vertical trailing-edge displacements were set to -10 mm and 10 mm respectively as it was thought that larger trailing-edge displacements were likely to lead to actuator saturation on the current version of SmartX-Alpha. Although previous DIC measurements on SmartX-alpha showed that training edge displacements of $\pm 14$ mm were achievable with symmetric actuation, these displacements could not be reproduced with the current version of SmartX-alpha. It is supposed that the observed reduction in achievable trailing-edge deflection between the two versions of the wing has been the result of performed maintenance. Manual maintenance activities such as the stiffening of the sliding interface and the re-gluing of the pick-up point of the actuation mechanism may have decreased the maximum realizable vertical trailing-edge displacements of the modules. Since the trailing-edge displacements are directly mattered in terms of actuator saturation and aerodynamic response, the individual virtual inputs are allowed to take greater values than are allowed for the trailing-edge displacements as long as they partially cancel out at the actuator locations such that the commanded trailing-edge displacements are within the set bounds.

During the first wandering phase, it was observed that the actuated shapes often led to actuator saturation. Therefore it was decided to perform a second wandering phase with reduced $z_{\text{ref}}$ limits. Additionally, the points sampled in this second wandering phase were focused onto a narrower region of the domain where to optimal solutions were expected to reside. As was observed in the simulations, the virtual inputs corresponding to the more complex higher-order Chebyshev polynomials have a more narrow range in smooth shapes. As the number of extrema within the fixed domain increases proportionally to the order number of the Chebyshev polynomials, their respective slopes increase as well as can be clearly seen at the last spanwise module in Fig. 3.5. Therefore, higher values for the higher-order virtual inputs decrease the smoothness of the shapes. However, it is expected that the true optimal shapes are relatively smooth. In this second wandering phase, the pseudo-randomly sampled input shapes were limited to $z_{\min} = -8$ mm and $z_{\max} = 8$ mm. The bounds of the individual virtual inputs $u_1, \cdots, u_5$ were narrowed to $\pm 11.2, 11.2, 10.5, 9.8$, and $9.1$ mm respectively.

During the first wandering phase, 150 iterations were performed. During the second wandering phase, with reduced bounds, an additional 57 measurements were performed.

## 6.2. Measurement routine

The start-up procedure of the online shape optimization wind tunnel experiment was as listed below.

1. turntable to zero position;
2. force balance de-biasing;
3. all morphing servo's to zero degrees;
4. vision system calibration (optional);
5. backlash controller de-biasing (optional);
6. wind tunnel start-up;
7. wind speed set to $15$ m/s;
8. online optimization algorithm start.

The first step was the resetting of the turntable to zero degrees of table angle for a consistent baseline. Then the noisy and non-zero mean force balance readings were de-biased such that their means were zero. If required, the IR-camera vision system was calibrated, in which case de-biasing of the backlash controller was also required. In hindsight, the biases observed after calibration of the vision system were likely caused by the fact that morphing servos were always set to zero degrees before calibration. However, due to the nonlinear backlash effects, the wing shape corresponding to uniform

zero actuator inputs was not consistent. Hence, the baseline from which the vision system estimated the trailing-edge displacements changed with each calibration, which manifested as the observed biases for the controller. The next steps were the start-up of the wind tunnel and the configuration of the desired wind speed, which was $15$ m/s. With all sensing equipment calibrated and the wind speed having converged to its set target, the online optimization was finally started.

**Baseline measurements** With the adaptations to the optimization framework and the experimental procedure as described so far, a number of wandering and optimization phase runs were performed as well as a baseline performance measurement routine with the wing jig shape. The full test matrix is shown in chapter 5. This baseline comprised an angle of attack sweep from -18 to 10 degrees. The purpose of the jig shape measurements is twofold. First, as mentioned before, the lift and drag measurements of the jig shape serve as the performance baseline to which the performance of the online optimized wing shapes will be compared. Secondly, the jig shape measurements are used to estimate the constant angular difference between the zero-position of the turntable and the aerodynamic angle of attack. Specifically, a first-order polynomial is fitted to the linear part of the lift curve and the corresponding zero-lift angle of attack is compared to the theoretical zero-lift angle of attack. This theoretical zero-lift angle of attack was determined to be $-6.5570$ degrees, with an evaluation of the Smart-X-Alpha geometry using the 3D panel viscous solver *XFLR5 v 6.48*. Since the transformation of the force balance measurements in the body-fixed reference frame to the lift and drag forces defined in the aerodynamic reference frame also depends on the aerodynamic angle of attack, the angular difference $\varepsilon$ between the commanded turntable angle $\psi$ and the aerodynamic angle of attack $\alpha$, was estimated using the iterative approach outlined in Chapter 5. The computed angular difference was $\varepsilon = -9.5789$.

**Wandering phase** As described in the preceding section, a total of 207 wandering phase iterations were performed using two sets of sampled wandering data points. During the experiment, the algorithm operated continuously and in theory, the wandering phase measurements could have been performed without pausing. However, in practice, the experiment often needed to be halted. In most cases, the controller was observed to suddenly command seemingly arbitrary and extreme deflections of some of the morphing actuators. These were caused by a loss of tracking information from the vision system. This in turn would be caused by the system losing track of one of the IR markers and thereby the rigid body defined by it. However, as the experiment progressed some vision system settings were adapted and these instances became increasingly rare. In the instances where the measurements had to be aborted, the wind tunnel was kept running. After re-acquisition of the target marker by the vision system, the experiment could be restarted within several minutes. For some shapes, the marker loss was observed to repeat. In these cases, the corresponding shape was skipped rather than repeated a third time. The measurement timestamps of groups of wandering phase measurements are shown in Tab. 6.2. The last four rows correspond to runs of continuous operation. The first two rows correspond to groups of 10 measurements each.

Table 6.2: Measurement timestamps of grouped wandering phase measurements.

| iteration | | date | | |
|---|---|---|---|---|
| from | to | | from | to |
| 1 | 57 | 26-Apr | 17:19 | 19:26 |
| 58 | 65 | 28-Apr | 12:40 | 13:13 |
| 65 | 100 | 28-Apr | 13:26 | 14:03 |
| 101 | 110 | 28-Apr | 14:13 | 14:23 |
| 110 | 120 | 28-Apr | 14:27 | 14:37 |
| 130 | 150 | 28-Apr | 15:06 | 15:28 |
| 141 | 207 | 29-Apr | 15:41 | 17:06 |

**Turntable reset** On April $28^{\text{th}}$, at the end of the day, the turntable was not set to zero before the turntable controller was shut down. On start-up on the next day, the zero angle of the turntable was

found to have shifted. By matching the force balance readings of the jig shape to their previously measured values, the difference in the table angle was estimated to be approximately 10.5 degrees. On the last day of the experiments, another jig shape angle of attack sweep was performed so that a more accurate estimate of $\varepsilon$ could be computed in post-processing.

**Optimization** After the wandering phases with the nominal and reduced bounds, the online optimization algorithm was operated in the optimization mode for various target lift coefficients. The target lift coefficients, number of iterations performed and the timestamps of the optimization experiments are shown in Tab. 6.3.

Table 6.3: Measurement properties and timestamps for the optimization phase measurements.

| $C_{L_t}$ | Iteration | Date | Start time | End time |
|---|---|---|---|---|
| 0.65 | 15 | 28-Apr | 18:21 | 18:39 |
| 0.75 | 15 | 29-Apr | 13:15 | 13:32 |
| 0.75 | 30 | 29-Apr | 13:34 | 14:14 |
| 0.40 | 30 | 29-Apr | 14:19 | 14:54 |
| 0.90 | 30 | 30-Apr | 11:54 | 12:30 |
| 0.40 | 30 | 30-Apr | 12:34 | 13:10 |
| 0.75 | 30 | 30-Apr | 13:13 | 13:49 |

In each of the optimization experiments, the on-board model weights and the training set buffers were initialized with the weights and buffer contents that resulted at the end of the latest wandering phase. Even though in operation on real-world aircraft the training samples added to the buffer during operation in optimization mode are of course retained, for the purpose of the wind tunnel experiments the various optimization runs were given the same “starting points” for the sake of comparability. The first online optimization experiment was conducted with a target lift coefficient of 0.65 and was continued for 15 iterations. This experiment took place after the wandering phase measurements with nominal bounds, but before the wandering phase with reduced bounds was performed. Hence, only the wandering phase measurements from iterations 1-150 were in the buffer. Note, that this experiment also took place before the zero position of the turntable was changed through an unprocedural shut-down.

## 6.3. Post turntable bias optimization results

The results from the successful optimization run performed before the angle of attack actuation bias was introduced has been discussed in chapter 5. In this experiment, a $7.8\ \%$ drag reduction, relative to the wing jig shape was achieved on SmartX-alpha. However, during the optimization phase experiments performed after the $28^{\text{th}}$ of April, no drag reductions were achieved relative to the wing jig shape. It, therefore, seems that, although the resetting of the turntable zero position was approximately compensated, a significant bias remained which had a prominent adverse effect on the performance of the online shape optimization algorithm. The true extent of the bias introduced into the data will be further investigated in section 6.4. In this section, the results of an unsuccessful optimization run are shown.

The estimated lift and drag of the wing shape and angle of attack combinations evaluated during online optimization with a target lift coefficient of 0.90 are shown in Fig. 6.2. It takes many iterations before the estimated lift coefficient somewhat converges onto the target lift coefficient. This indicates that the on-board model was initially unable to accurately predict the mapping between the system inputs and the lift coefficient. With the gathering of more data and online training, the on-board model's ability to predict the lift coefficients seems to have improved. Nevertheless, compared to the wing jig shape, at the same lift coefficient, the estimated drag coefficients of all of the evaluated inputs were higher instead of lower. While a slowly decreasing trend in the amount of relative drag increase can be observed, the actuated inputs do not outperform the wing jig shape before the experiment was halted after 30 iterations. In conclusion, there initially seems to be a large mismatch between the system output estimates predicted by the on-board model and the estimated system outputs. However, with more data collection and online learning, the model prediction error does decrease. Although this decreasing trend was observed, the experiment was halted due to time constraints. Since data

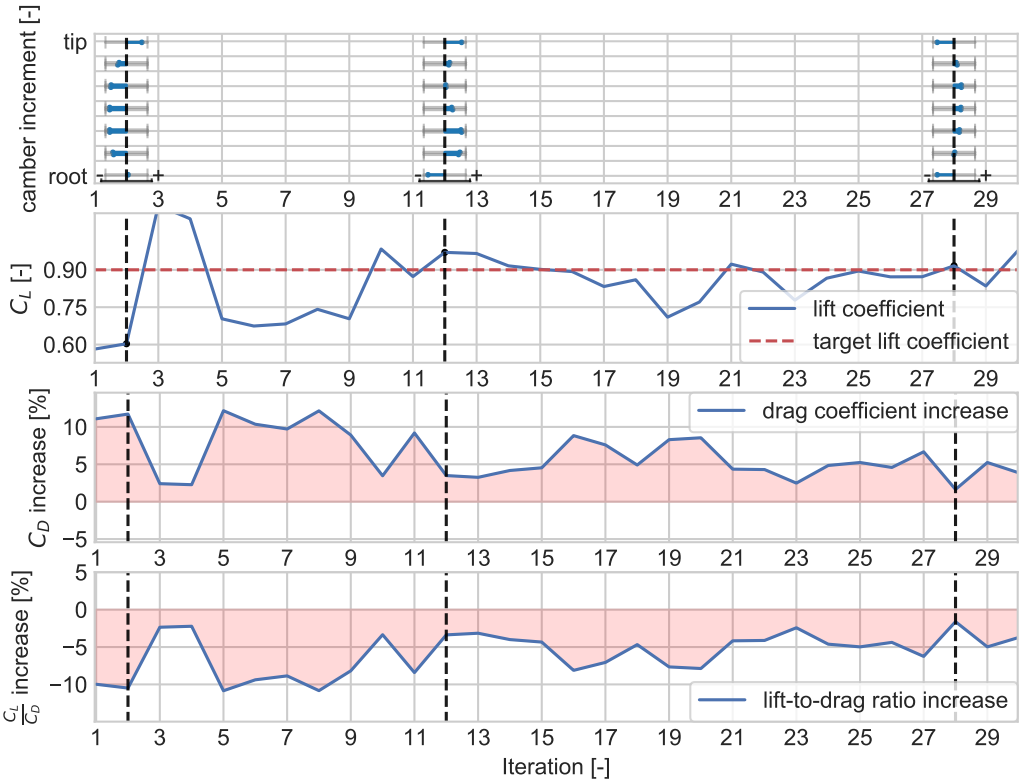



Figure 6.2: Results from online shape optimization after the introduction of the table-bias with a target lift coefficient of 0.90.

collection was rather slow (1 min.) due to the time required for transient aerodynamic effects to settle and the long measurement time required to average out the substantial measurement noise, big data collection could not be afforded. Moreover, in real-world applications in flight, the methodology should yield drag reductions within few iterations, as was observed in the April 28 experiment.

## 6.4. Post processing and simulation

### 6.4.1. Effects of angle of attack bias

In post-experiment processing of the last set of jig shape measurements (post table bias) the angle between the turntable zero position and the aerodynamic angle of attack $\varepsilon$ was estimated using the estimation routine described in chapter 5. With this procedure, the angular difference between the aerodynamic angle of attack and the turntable zero angle was estimated to be $\varepsilon = 8.9019$ degrees. This improved estimation of $\varepsilon$ represents a more accurate estimation than the manual force balance matching performs in the wind tunnel on the day the new bias was discovered. At that time $\varepsilon$ was estimated to be approximately $10.5$ degrees. This means that the initial estimation of $\varepsilon$ was off by approximately $1.6$ degrees. Hence the angles of attack actuated during the last wandering phase with reduced bounds, and all but one of the optimization runs were approximately $1.6$ degrees lower than supposed. On top of that, the supposed angles of attack were also used to transform the measured forces in the body-fixed reference frame to the aerodynamic reference frame using Eq. (6.1). Hence, both the actuated angles of attack and the estimated lift and drag coefficients of the actually actuated inputs were off for these measurements.

$$\begin{bmatrix} \mathrm{L} \\ \mathrm{D} \end{bmatrix} = \mathbf{T}(\alpha) \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} \sin\alpha & \cos\alpha \\ -\cos\alpha & \sin\alpha \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{6.1}$$

Due to the angle-of-attack-dependent nonlinear transformation that is part of the lift and drag estima-

tion method, the effect of the turntable bias on the estimated lift and drag forces is more complex than the effect that an angle-of-attack shift has on the true lift and drag forces. As has been shown for the wing jig shape, changes in angle of attack with a fixed-wing shape follow a curve in the lift-to-drag ratio versus lift coefficient plot. An example can be seen in the green jig shape curve in Fig. 4.17. Positive steps in angle of attack cause greater lift coefficients, and generally result in lower lift-to-drag ratios, except for the lower lift coefficients $C_{\mathrm{L}} < 0.4$. For these lower lift coefficients, increases in angle of attack, and by extension increases in the lift coefficient lead to increases in the lift-to-drag ratio. For these lift coefficients, the amount of change in lift-to-drag ratio increases with smaller lift coefficients. The effect of the introduced turntable bias is different, however, it can be deduced from a set of measurements that were repeated, with the same shapes, after the turntable zero position was reset.

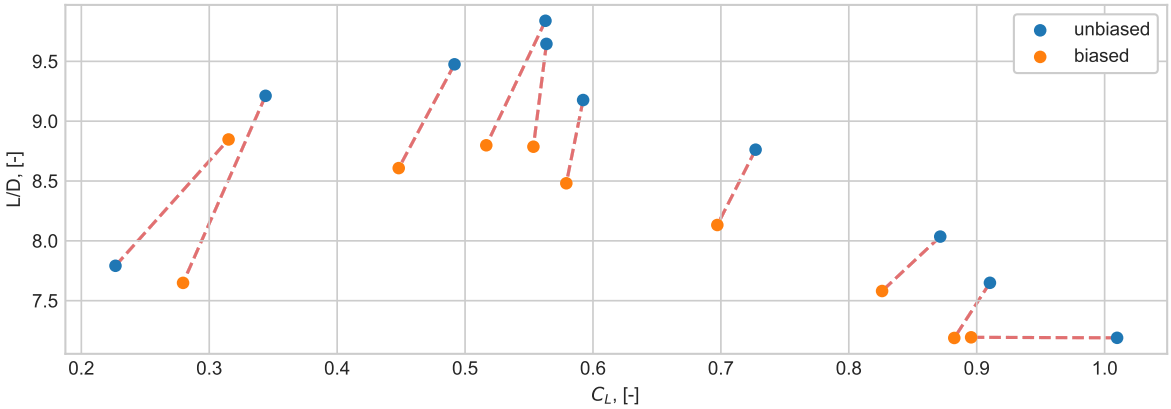



Figure 6.3: Uncorrected estimated lift-to-drag ratios from repeated wandering phase measurements from before (blue) and after (orange) the turntable reset bias.

In Fig. 6.3, the estimated lift-to-drag ratios versus the estimated lift coefficients of both sets of iterations 141 through 150 of the wandering phase are shown. The dashed lines connect the measurements that correspond to the same wing shapes. First of all, the effect of the bias is very pronounced in the vertical axis. The estimated lift-to-drag ratios of the biased measurements are mostly lower than their unbiased counterparts. The apparent shifts in lift-to-drag ratios between the samples with the same wing shapes seem to exceed the lift-to-drag ratio differences between surrounding samples of the same set. In other words, the effect of the turntable bias on the estimated aerodynamic efficiency is stronger than the effects of different wing shapes for similar lift coefficients. Note that the differences between the two sets of measurements are twofold. While the same shapes were actuated, the angles of attack are off by approximately $1.6$ degrees, and the transformation of the measured forces is also affected. Based on the first of these, a more consistent and more dominant leftward horizontal shift would be expected owing to the lift reduction that is caused by the lowered angle of attack. However, this effect seems to be overshadowed by the second effect. While the angle-of-attack-dependent transformation does not affect the total aerodynamic force, it does greatly affect the decomposition of this force into the lift and drag components. The effect of a change in the angle of attack used for the transformation with fixed forces $F_x, F_y$ is shown in Fig. 6.4. During the experiments, the commanded angle of attack was assumed to be equal to the true angle of attack, and hence the commanded angle of attack was used for the transformation of the measured forces. Since, the truly actuated angle of attack was approximately $1.6$ degrees lower, the angle used for the transformation was approximately $1.6$ degrees too high. As shown in the graph, this corresponds to a $1.65$ decrease in the estimated lift-to-drag ratio. In summation, the observed effects of the turntable bias on the estimated lift-to-drag ratios are caused by a combination of a variable, but consistently negative transformation error and a variable error due to the actuation error. This last error has a negative lift-to-drag ratio contribution for most samples, with the exception of the samples that have small lift coefficients.

Because (i) the lift-to-drag ratio estimation error caused by the angle-of-attack bias is larger than the lift-to-drag ratio variation of the pseudo-random wing shapes with similar lift coefficients, and (ii) the biased data made up only $27.5$ % of the training set, it is perhaps unsurprising that in the post turntable bias optimization experiments, the on-board model was not able to accurately predict the

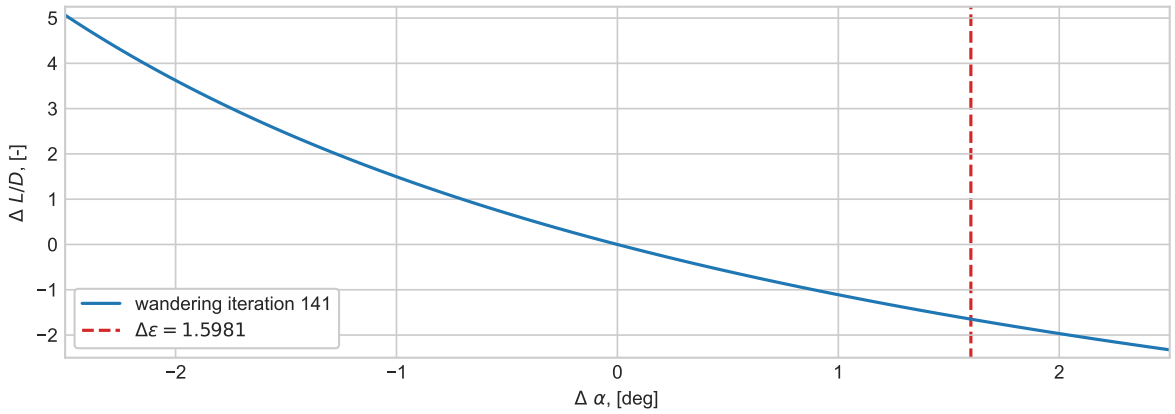



Figure 6.4: Change in estimated lift-to-drag ratio as function of change in angle of attack used for the transformation for wandering phase iteration 141.

best-performing shapes. Looking back at Fig. 6.3, it would be challenging for any method to accurately predict the locations of the orange dots based on a data set which is for $72.5$ % made up by the blue dots.

### 6.4.2. Post measurement table-bias correction

In further post-processing of the measurement data, the samples measured after the zero position of the turntable was reset were corrected. This correction was performed by first correcting the commanded angle of attack $\alpha_c$ to obtain the actually actuated angle of attack $\alpha_{act}$ with $\alpha_{act} = \alpha_c + \Delta\varepsilon$, where $\Delta\varepsilon = -1.5981$ degrees. Next, the original averaged force balance measurements in the body-fixed reference frame $\bar{F}_x$, $\bar{F}_y$ were again transformed to the lift and drag forces defined in the aerodynamic reference frame with the angle of attack dependent transformation shown in Eq. (6.1) using the actuated angle of attack.

The resulting bias-corrected repeated measurements from wandering iterations 141 through 150 are shown in Fig. 6.5. Here, the large and consistent discrepancy in the lift-to-drag ratios observed for the two sets of measurements in Fig. 6.3 is no longer visible. In contrast with Fig. 6.3, without the color-coding, the two sets of measurements would be indiscernible in Fig. 6.5. Nevertheless, the observed differences between the unbiased and bias-corrected measurements are cannot be used to quantify the overall measurement error in the experimental set-up. Although the transformation error has been corrected, the actuation error has not been corrected. Since the effect of the actuation error actually affects the aerodynamics and by extension, the measured signals rather than the post-processing of these signals, this type of error, although of known origin is impossible to correct.

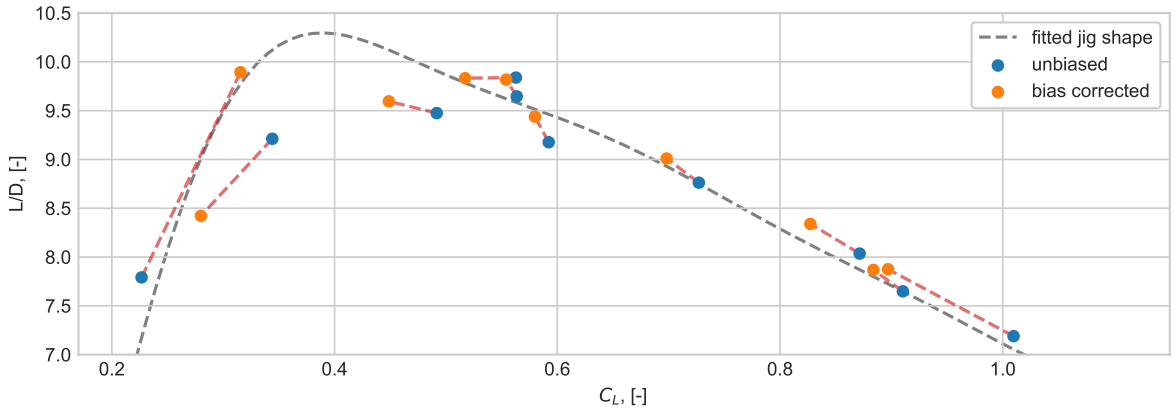



Figure 6.5: Uncorrected estimated lift-to-drag ratios from repeated wandering phase measurements from before (blue) and after (orange) the turntable reset bias.

The differences between the connected samples in Fig. 6.5 are the combined effects of the measurement error and the $-1.5981$ degree error in the angle of attack actuation. The effects of steps in angle of attack for the same shape cause steps along the aerodynamic efficiency curve of that wing shape. The exact shapes of the efficiency curves are different for each of these shapes and are unknown. The unique efficiency curves of various wing shapes attain their maximum values at different lift coefficients, as some shapes are more efficient for producing either low or high amounts of lift. It is precisely by cherry-picking the best sections of the various efficiency curves through using different shapes that active morphing wings are able to achieve high aerodynamic efficiency for any lift coefficient. However, even though the exact efficiency curves are unknown, they can are expected to be similar to the efficiency curve of the jig shape, shown in black. Therefore, it can be verified that the corrected samples represent steps along the approximate direction of the jig shape and no clearly discernible biases remain.

The aforementioned bias corrections were made to all of the post-turn-table-zero-position-reset measured wandering samples. The uncorrected lift-to-drag ratio estimations are shown in Fig. 6.6a. In this graph, the unbiased and the biased sample distributions are clearly separated. The same samples, after correction of the angle-of-attack bias induced transformation error, are shown in Fig. 6.6b. The biased and subsequently corrected data points are overlapping with the earlier measured data points that were not biased. The lowest lift coefficients of the distribution of the corrected samples are somewhat smaller than the lowest lift coefficients of the unbiased samples. Although the commanded angles of attack of both sets of pseudo-random inputs were sampled using the same bounds, the actuated angles of attack of the last 57 iterations were approximately $1.6$ degrees lower than their commanded angles of attack. On average it also seems as though the lower lift coefficient samples of the corrected set have slightly lower lift-to-drag ratios than the unbiased samples. The exact cause of this remaining difference cannot be definitively proven. However, the observed behavior indicates that an increased zero-lift drag coefficient could be the culprit. As lower lift coefficients, in general, have a lower total drag, the lift-to-drag ratio becomes increasingly more sensitive to the zero-lift drag coefficient at low lift coefficients, while at higher lift coefficients the zero-lift drag coefficient makes up a relatively small portion of the total drag. Since, the two sets of measurements were gathered on different days, with other measurements being performed in between, it is not implausible that a small drag increasing change, such as the partial detaching of pieces of tape from the wing surface, may have occurred.

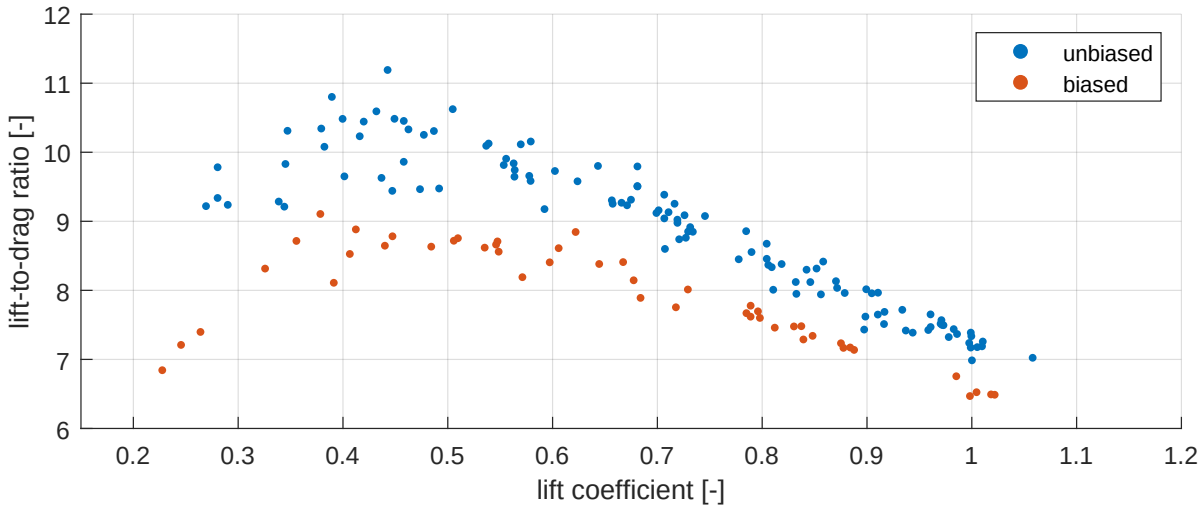
### 6.4.3. Further data set cleaning

In addition to the correction of the angle of attack actuation bias, the tracking signals of the vision system were also critically reviewed to improve the overall quality of the data by filtering out samples for which the tracking of the reference signals was unsatisfactory.

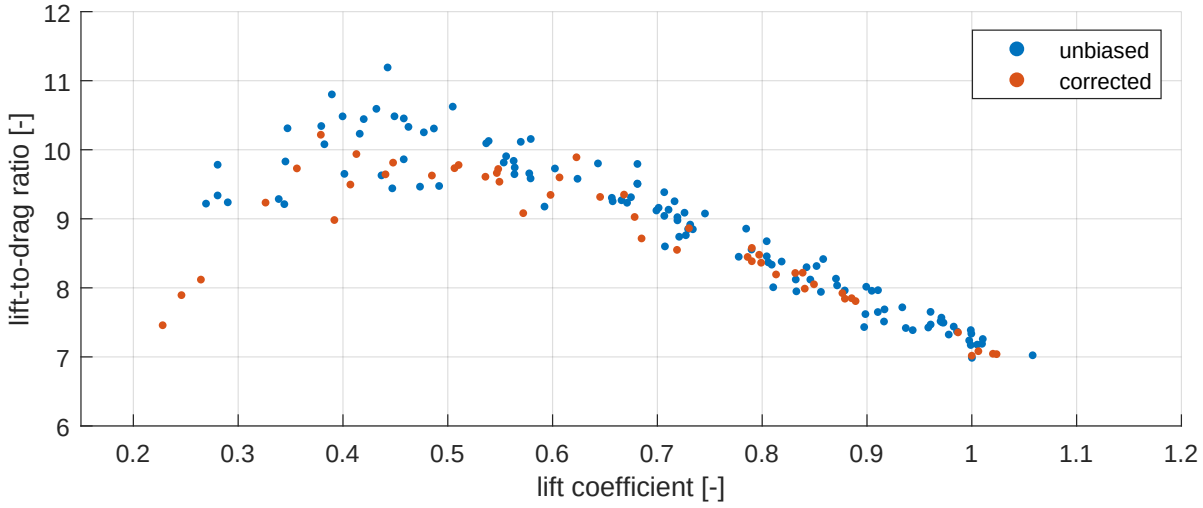
The twelve commanded local vertical trailing-edge deflections serve as the final reference signal values for the controller to steer the 12 measured local vertical trailing-edge displacements from the vision system. Because of the backlash hysteresis phenomenon, the one-minute controller cycles start with a doublet input to consistently shake the wing trailing edge. Then, the reference signals transition to their final constant values through a sigmoidal step. The reference signals reach their final values after approximately 11 seconds into the controller cycle. However, the convergence time of the controller varies per commanded shape as motions of the trailing edge at the 12 spanwise stations are coupled. From 20 seconds into the cycle until the cycle end at 60 seconds, the measured aerodynamic forces resulting from the actuated shape are averaged. During this period, the wing shape should remain constant, i.e., all tracking responses should be stationary. The tracking signals of all wandering phase measurements were visually inspected to ensure their timely convergence.

An example of a tracking signal which did not converge within the measurement window is shown in Fig. 6.7. During the averaging period, the wing shape should not change anymore, as is the case at the first servo location in module two shown in orange. However, the tracking signal of the second servo location (purple) indicates an undesirable movement of the trailing edge. For this reason, this data point was not used for the post-data-driven simulated wandering experiment. Note that the tracking signals should converge to a constant value, but not to the exact reference signals indicated by the blue and yellow lines. Due to the calibration-dependent tracking biases described in section 6.2, a constant offset is observed when in fact, the tracking error has converged to zero. Therefore, the flatness of the tracking signals was inspected rather than their differences from their tracking signals.

In general, it was decided to omit the samples for which the difference between the minimum and




(a) Comparison of the estimated aerodynamic efficiencies of all the wandering phase measurements before correction.




(b) Comparison of the estimated aerodynamic efficiencies of all the wandering phase measurements after correction.

Figure 6.6: Effects of the post experimental turntable bias correction on the estimated lift-to-drag ratio versus lift coefficient distributions for the wandering measurements.

maximum tracking signal values during the last 40 seconds of the controller cycle was greater than 1.5 mm for any of the tracking signals. In total 28 measurements were omitted. For wandering phase iterations 1 through 57 no tracking signal data was saved. Therefore, these samples were also omitted so that the tracking convergence could be guaranteed for the data used.

### 6.4.4. Experimental Data-driven Optimization Predictions

The cleaned and bias-corrected data were used to re-perform the wandering phase part of the wind tunnel experiment in simulation. Because only the transformation error of the angle of attack actuation bias was corrected, the actuated angles of attack were still different from the commanded angles of attack for wandering phase iterations 150 through 207. However, these data points could still be used because their actually actuated angles of attack were known. The purpose of the wandering phase experiment is the exploration of the search space. The used inputs are pseudo-random and their exact values are not of interest, but rather the underlying mapping from wing inputs to aerodynamic performance is. The remaining data was divided into training and validation data sets with a $75\,\%/25\,\%$ split.

During the simulated wandering phase experiment, the optimization algorithm operated similarly to the wind tunnel experiment. However, rather than actuating pseudo-random inputs of the wing and sending the averaged force balance measurements, the estimated lift and drag coefficients from the

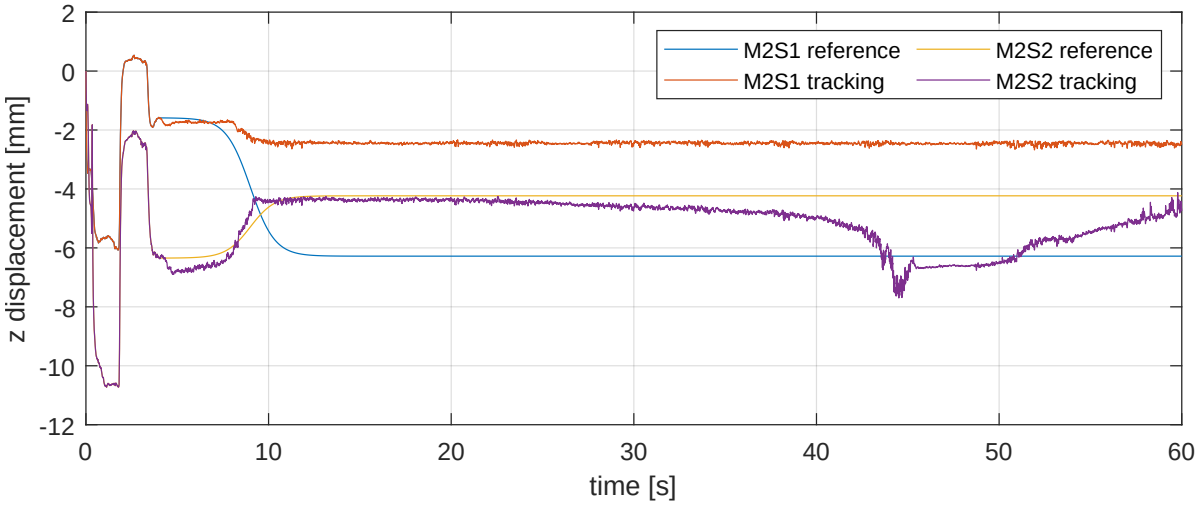



Figure 6.7: Example of a desirable (orange) and a undesirable (purple) tracking signal response in module 2 at wandering phase iteration 176.

inputs are fed to the optimization algorithm directly from the training data set. With this architecture, the online optimization algorithm is unable to evaluate new input as there is no actual wing system or model present. Therefore, the optimization phase experiment cannot be simulated this way. However, this setup is suitable for simulating the wandering phase with the same inputs as used in the wind tunnel experiments. Lastly, in contrast with the wind tunnel experiment, in the wandering simulation, the best inputs were estimated by the optimizer using the on-board model during each iteration.

The Mean Squared Error (MSE) losses of both the lift coefficient and the drag neural networks on the training and validation data sets are shown in Fig. 6.8. The training of the neural networks started from iteration 32, which is when the first iteration where the number of samples in the buffer was greater or equal to the mini-batch size which was also 32. Each iteration of the algorithm includes 25 epochs of learning on the training data set. As expected, the training losses are consistently lower than the validation errors. The addition of new measurements to the training data set every 25 epochs manifests as step increases in the training loss. After each step, the training loss subsequently decreases with some noise due to the network weights being adapted to minimize the loss on the expanded training set. The validation losses on the other hand are not so sensitive to the addition of the new measurements. However, with the slow increase in the amount of training data, the generalizability of the trained model also slowly increases, i.e., the validation loss slowly, but certainly decreases. This trend indicates that with more training data, the on-board model could be improved even further.

The most optimal system inputs for a target lift coefficient of 0.65 as computed by the optimizer on the on-board model during the simulated wandering phase experiment are shown in Fig. 6.9. The marked bounds on the inputs are dictated by the bounds of the training data for the on-board model, which are bounds on the wandering data points. At iteration 32, when the online training starts, large steps in all computed optimal inputs are observed. Subsequently, the optimal inputs slowly transition as the online training of the on-board model continues. The final optimal wing shape was shown in chapter 5. The first virtual input $u_1$ gradually increases to $3.8$mm at the end of the simulation. The positive value for $u_1$ indicates an increase in mean camber across the wing as can be observed in Fig. 3.5. The second virtual input $u_2$ decreases to $-4.4$ mm which results in a spanwise decrease in camber towards the wing tip end. The third virtual input decreases to its negative limit at $-7.5$ mm, which corresponds to an increase in camber in the spanwise center and an equal decrease of the local camber towards both the wing root and tip ends. The fourth virtual input is decreased to $-0.7$ mm, which corresponds to a small decrease in camber in module 2 and at the wingtip, and a small increase in camber at the wing root and in module 5. $u_5$ decreases to $-2.2$ mm, which corresponds to camber decreases at the wing ends and center, and an increase at $\frac{1}{6}$ wingspan from both ends of the wing.

The normalized root mean square errors of both neural networks were shown in chapter 5. However, these metrics only indicate the prediction accuracy of these networks on the entire validation data set. They do not provide insight into the distribution of the prediction errors on this set. The absolute

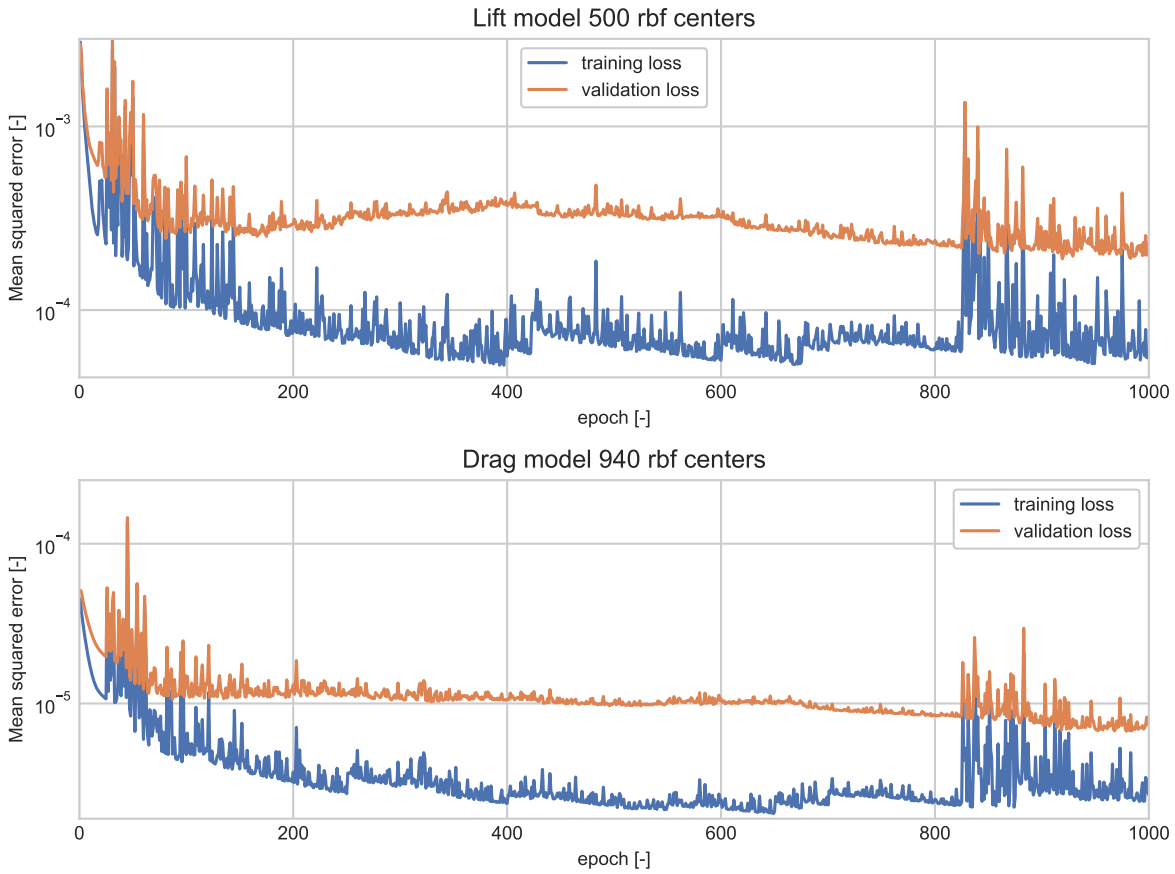



Figure 6.8: Training and validation losses of the lift and drag coefficient neural networks during simulated wandering.

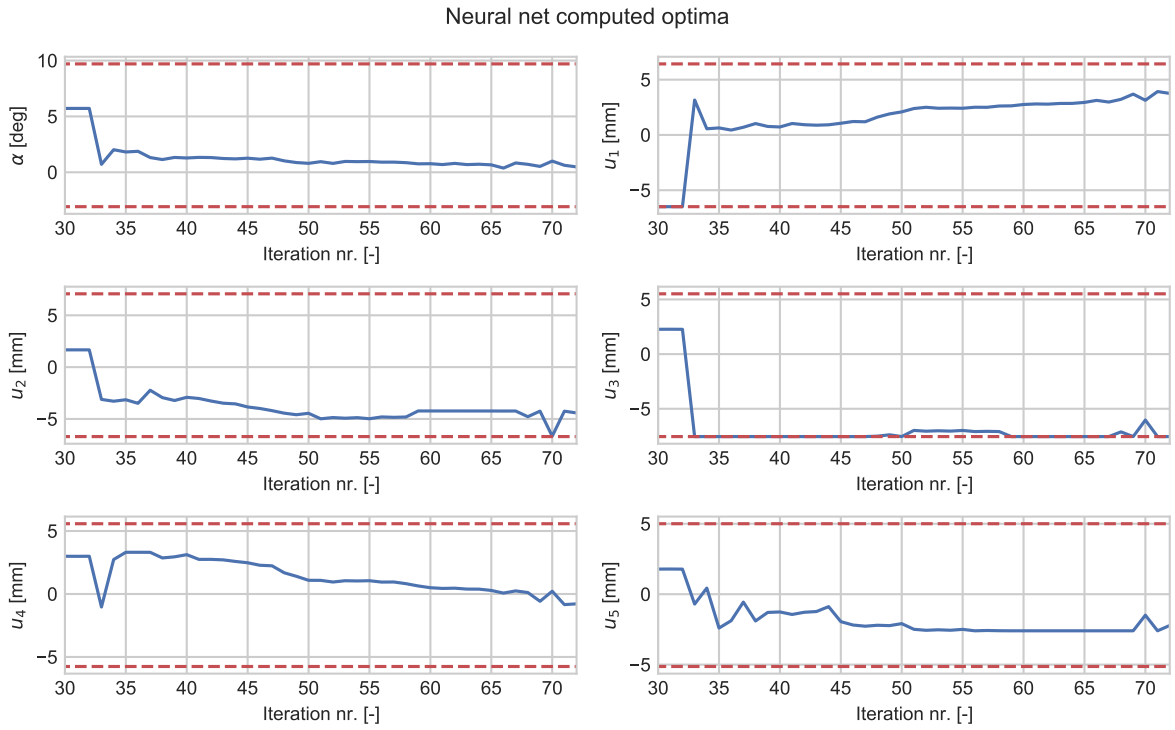



Figure 6.9: Evolution of the computed optimal inputs during simulated wandering, with limits marked by the red dashed lines.

prediction errors of lift and drag coefficient networks are not correlated with the lift and drag estimates. However, for smaller drag coefficients, the drag prediction error becomes larger in the relative sense. Coincidentally, for small drag coefficients, the lift-to-drag ratio becomes more sensitive to the drag prediction error. The variation of the absolute lift-to-drag ratio error on the validation data set with the estimated drag coefficient is shown in Fig. 6.10. A clear trend of increasing absolute lift-to-drag ration prediction error with smaller drag coefficient values can be observed. While the prediction errors are randomly distributed and can have both positive and negative contributions to the predicted lift-to-drag ratios, the optimizer, by its very nature will converge onto the point with the highest lift-to-drag ratio and thus find points where the lift-to-drag ratio is overestimated. Therefore, the predicted lift-to-drag ratios presented in chapter 5 are expected to be optimistic.

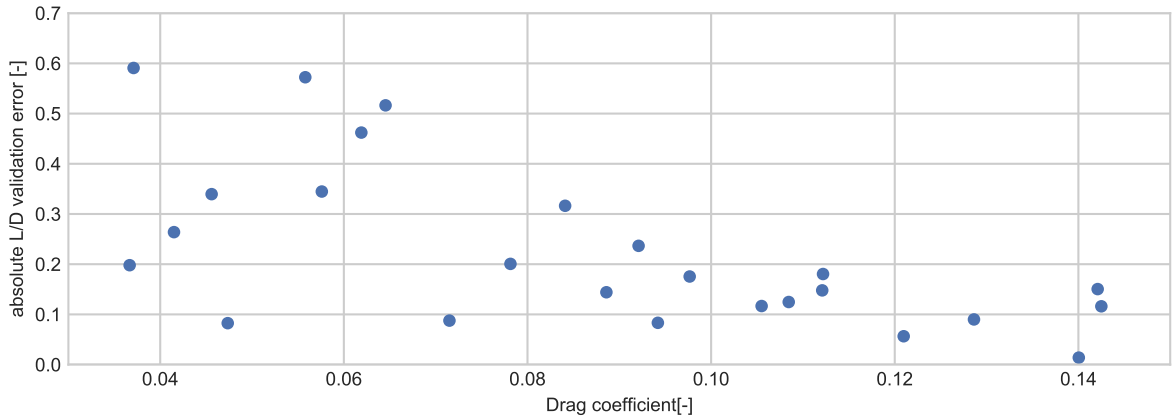



Figure 6.10: Variation of the absolute lift-to-drag ration error on the validation data set with the estimated drag coefficient.

As established earlier, the on-board model can be further improved with more training data. In real-world operation on an aircraft, the training data set would be many times larger than 71 samples. The 71 samples in this case, are the result of the combined effects of the relatively long (1 min.) measurement time, the removal of samples with undesirable tracking signals, and the reservation of $25\%$ of data points for validation. Furthermore, in the optimization mode, the computed optimal inputs are of interest rather than the predicted performance. Since during this phase the predicted optimal inputs are actuated on the system and the resulting performance is measured, the exact predicted performance is of little importance.

# 7 Discussions

In this chapter a comparison is made between the results from the simulation and wind tunnel experiments. First, the optimal wing shapes identified in both experiments are compared, followed by a comparison and explanation of the performance of the optimal shapes. Lastly, a comparison is made to the performance measured in a previous wind tunnel campaign.

## 7.1. Optimal shapes

**spanwise distribution** When comparing the optimal wing shapes found in the optimization of the simulation model and in the optimization performed on the SmartX-Alpha hardware in the wind tunnel experiment, some differences are observed. First of all, the simulation optimal shapes show elliptical distributions where the camber monotonically decreases towards the wingtip (free end) only.

The best performing wind tunnel optimal shapes show camber build-off towards both the wingtip and root. This effect has likely been caused by pressure leakage at the wing root due to an approximately 1 cm wide clearance gap between the wing root and the flow table surface in the wind tunnel set-up. This makes flow around the wing different from that of a perfect wing-fuselage intersection and a little closer to the flow around a free end.

**wing camber versus angle of attack** Since both the wing camber and the angle of attack can be adapted to realize a certain amount of lift, an infinite number of possible angle-of-attack and wing shape combinations exist that will satisfy a given lift requirement. However, the drag associated with these various combinations differs and thus the lowest drag combination is desired. As previously discussed, the ideal spanwise lift distribution is either a single or double elliptical shape depending on the wing root and fuselage intersection. The preferred spanwise lift distribution and the preferred overall amount of camber be conflicting. Hence, the optimal wing shape is ultimately a trade-off between a good spanwise lift distribution and a good angle-of-attack versus overall wing camber trade-off.

In the optimal wing shapes found in the simulation experiments with the developed aerodynamic model, the general trend in the camber versus angle of attack trade-off is that relatively high angles of attack with camber reducing morphing actuation are favored. Although increased airfoil camber is generally a more efficient mechanism of lift induction than angle of attack increments at low Reynolds numbers, the baseline NACA6510 airfoil is already relatively highly cambered. This trade-off is also consistent with the trends shown in the wind tunnel data from previous campaigns on which the augmentation of the aerodynamic model was based as was shown in Sec. 3.1.5. For low target lift coefficients, it was observed that the optimizer favored an almost completely uniform and maximal spanwise reduction of the wing camber. This wing shape was far from elliptical. This indicates that in the case of the aerodynamic model, the benefits of a more appropriate amount of camber for the low lift coefficient outweighed the benefits that more elliptical lift distributions could have had. This makes sense since the elliptical lift distribution reduces the induced drag component. This component is naturally smaller for low lift coefficients. The shapes for higher target lift coefficients were found to be increasingly more elliptical. However, only the highest target lift coefficient shape used the full morphing range. This was needed to meet the target lift coefficient, as the maximum allowable angle of attack had already been

reached. The other observed optimal shapes in the simulations do not use the full morphing range because the drag penalty from increased amounts of camber was found to not outweigh the reduction in drag from an increased spanwise decrease in camber. Hence, the ellipticalness of the camber distributions is sacrificed for lower overall amounts of camber. The word ellipticalness is taken to mean the degree to which a given shape resembles an ellipse.

In the optimal shapes identified from the wind tunnel experimental data, a different trend in the trade-off of overall wing camber and the spanwise lift distribution was observed. In the trade-off between the angle of attack and the overall amount of camber, these optimal solutions on average comprised higher amounts of camber, with the maximum amount of camber being close to the allowable upper limit, and generally lower angles of attack than the simulation-based solutions for the same lift coefficients. These shapes were found to make use of almost the entire morphing range. This indicates that in reality, it is beneficial to actuate as much spanwise decrease of camber as possible within the actuation limits. For increasing lift coefficients, it was observed that increased overall amounts of camber were used by slightly reducing the ellipticalness of the camber distribution. However, the full allowable range of morphing is used in each instance.

From these comparisons, it can be concluded that although the simulation model included similar drag source dynamics such that the ability of the optimization framework to optimize this trade-off could be demonstrated, the exact optimal solutions were different. A probable cause for the observed difference in the exact locations of the optima in the search spaces of the simulation and wind tunnel is the low accuracy with which the viscous effects are modeled in the aerodynamic model. The aerodynamic model is based on the inviscid vortex lattice method and thus viscous effects were neglected by default. However, a low-order estimate of the viscous effects was included through the model augmentation based on wind tunnel data from a previous wind tunnel campaign. Without this augmentation, no camber versus angle of attack trade-off dynamics would be present in the aerodynamic model. The model augmentation introduced an estimate of the effects of the overall wing camber on the zero-lift drag and the Oswald efficiency factor through first- and second-order polynomial functions in the mean equivalent flap angle. This augmentation neglected the intricacies of the spanwise distribution of the camber and other 3D effects. However such information was also not contained in the data set which comprised only measurement with constant spanwise camber morphing settings.

Nevertheless, as discussed in Sec. 3.1, the precise prediction of the optimal wing shape and angle-of-attack combinations was never the intended purpose of the aerodynamic model. Instead, the aerodynamic needed to simulate representative camber versus angle of attack trade-off dynamics such that the effectiveness of the proposed strategy could be demonstrated. This has been achieved. It should also be noted that in this thesis the aerodynamic model has only been used for preliminary design and testing of the proposed online wing shape optimization methodology. In other words, the aerodynamic model has served as a digital playground and learning environment for the proposed framework. However, the final validation of the effectiveness of the method has been evaluated using the SmartX-Alpha demonstrator in a real-world wind tunnel experiment. The results from this experiment have shown that the proposed methodology was able to adapt to this new environment and realize improved aerodynamic performance using new wing shapes. Lastly, it should be mentioned that the difference between the simulation and wind tunnel experimental optima once again highlights the importance of real-world experiments in wind tunnels or test flights for the maturation of promising new technologies.

## 7.2. Estimated performance

The estimated performance of the optimal shapes during offline optimization on the nominal aerodynamic model is lower than the best-performing measurements from the wind tunnel experiments. The reason for this underestimation of the morphing wings' performance can be traced back to the data set which was used to correct the inviscid aerodynamic model.

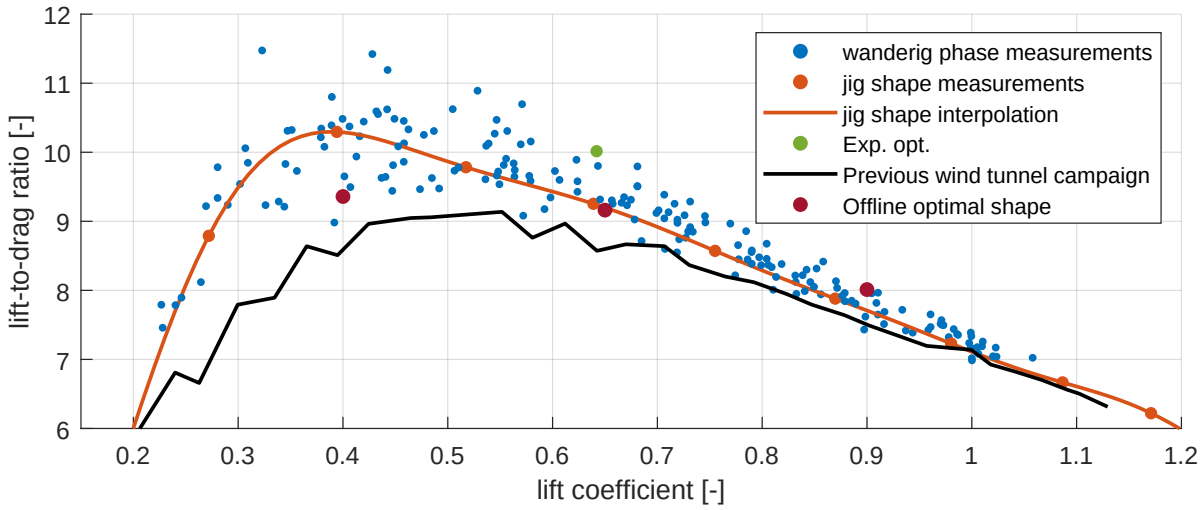



Figure 7.1: Comparison of the lift-to-drag ratios of measured during the wind tunnel campaign, the predicted performance of the offline optimized shapes and the lift-to-drag ratios measured in the previous wind tunnel campaign.

In Fig. 7.1, the predicted performance of three optimal shapes computed on the nominal aerodynamic model are shown (dark red), as well as the performance of the jig shape during the previous wind tunnel campaign (black), and the performances measured during the more recent wind tunnel campaign described in this thesis. From this figure, it is evident that the measurements of the jig shape in the two wind tunnel campaigns are not fully in an agreement. The older lift-to-drag ratio measurements are lower than most of the more recently measured lift-to-drag ratios with the only exceptions near $C_L = 0.2$ and near $C_L = 1.0$. Additionally, the discrepancy between the two measurement sequences varies with the associated lift coefficient, with the largest discrepancy being observed near $C_L = 0.35$. Hence, it is only natural that the prediction based on a model built from the older measurements underestimates the lift-to-drag ratios of the optimal shapes. However, while it is impossible to definitively prove what the causes of these discrepancies are, some differences in the measurement set-ups which have likely contributed to the measured discrepancy are known.

1. Between the wind tunnel campaigns, some changes have been made to SmartX-Alpha during maintenance, which have resulted in a reduced range of achievable trailing-edge displacement.

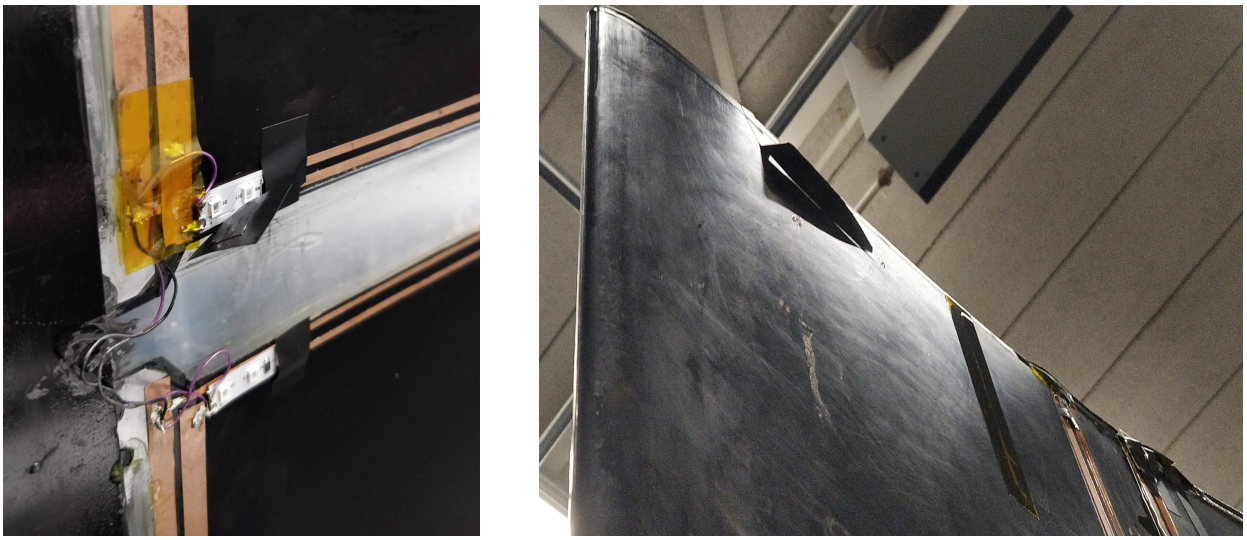
2. Due to the backlash in the morphing mechanism, the true shape of the wing may have been slightly different between the measurements campaigns. If the jig shape shaking maneuver is performed under aerodynamic loading, the shape after the shaking maneuver is also inconsistent.

3. The portion of the drag caused by the cleanliness of the wing surface was different between the wind tunnel campaigns. Pieces of tape were used to secure exposed wiring and IR markers to be used for wing shape estimation by the vision system. Differences in the number, size, and placement of the various pieces of tape on the wing surface and at the wing tip are also a source of drag discrepancy. Especially pieces of tape that partially detach from the wing can have an adverse effect on the total drag measured.

4. The tip end of the wing is open by default. During the most recent wind tunnel campaign, this open end was closed off using flexible tape. However, the applied tape could not fully cover the entire movement range of the TRIC module during morphing. Manual taping is first of all inconsistent and, moreover, the quality of the seal degrades over time under motions of the morphing mechanism.

5. The clearance gap between the horizontal flow table and the wing root was different between the two series of experiments. During the wind tunnel campaign described in this thesis, a considerable amount of time and effort was put into minimizing the width of this gap while still allowing free rotation of the wing. The reduction of this gap at the wing root has resulted in a reduction of pressure leakage at this location compared to previous wind tunnel experiments and has therefore naturally enhanced the efficiency of the wing.

The observed differences likely are the combined result of these differences. However, the prime factor is thought to be the reduced root clearance gap width. The larger increase in difference in lift-to-drag ratio between the two wind tunnel campaigns for lower lift coefficients indicates a reduction in zero-lift drag coefficient in the second wind tunnel campaign. While this is a constant factor, the sensitivity of the lift-to-drag ratio to a fixed “drag penalty” increases with a lower total drag, and thus with lower total lift. The zero-lift drag differences may primarily have been caused by differences in the surface cleanliness in the two campaigns. However, the behavior of partially separated pieces of tape is more dynamic in nature and their effects are harder to predict. However, even if they are to reach a steady-state equilibrium, this equilibrium may be dependent on the angle of attack and even the actuated wing shape.



(a) Loose tape discovered at one of the infra red marker locations

(b) Loose tape discovered on the bottom surface near the wing tip.

Figure 7.2: Two examples of pieces of tape that were found to have been partially detached from the wing during the experiment.

**Morphing limits** The maximum achievable trailing-edge deflections during the wind tunnel campaign were lower than those used in the simulation model. While the morphing limits configured in the simulation model were derived from accurate Digital Image Correlation (DIC) measurements, these measurements were not performed on the most recent version of SmartX-Alpha. During maintenance, the interface between the sliding surfaces was made stiffer and the pick-up point of the morphing mechanism was re-glued. These changes to the SmartX-Alpha hardware have resulted in a reduction of the maximum trailing-edge deflection that can be achieved. Between these different versions of SmartX-Alpha, the actuation ranges of the morphing actuators are unchanged, however, the amount of vertical trailing-edge displacement resulting from these actuator deflections has changed.

This has resulted in a slight discrepancy between the amount of morphing realizable in the simulation model and the wind tunnel experiment. However, other previously discussed factors have greater contributions to the differences between the simulation model and the real SmartX-Alpha demonstrator, and this discrepancy can be tolerated. As discussed in chapter 5, this has also resulted in discrepancies between the commanded and actuated wing shapes during the wind tunnel experiment since the achievable trailing-edge displacements were overestimated. Consequently, saturation of the morphing actuators occurred relatively frequently.

# 8

# Conclusions and recommendations

In this chapter, the conclusions of the thesis research are presented, followed by the recommendations for future work on the online wing shape optimization framework.

## 8.1. Conclusions

In this section, the conclusions of the thesis work are drawn. The research questions from sec. 1.2.1 are revisited and answered. Lastly, the research objective from sec. 1.2.2 is reviewed.

1. **What are suitable modeling methods for trailing-edge camber morphing wings with low computational cost and sufficient accuracy for the purpose of online black-box performance optimization?**

   In this thesis, a Vortex Lattice Method (VLM) was used to model the aerodynamics of a trailing-edge camber morphing wing. The geometry of the morphed wing was represented with a full-span distributed continuous flap wing. This model allows for spanwise twisting of the trailing-edge flaps without discontinuities or gaps between these discrete flaps.

   VLM is a low-order inviscid aerodynamic method, which allows for low-computational cost, and thus fast evaluation. However, the inviscid nature of the method limits the validity of the model to small angles of attack, where no flow separation occurs. However, due to the fact that the target conditions are prolonged flight in steady-state conditions, these lower angles of attack also correspond to the primary region of interest. Additionally, the VLM methods are suitable methods for the prediction of lift and induced drag. The zero-lift drag, which primarily originates from viscous effects rather than the pressure distribution cannot be modeled with a VLM.

   To this end, the VLM aerodynamic model was augmented using functions of the zero-lift drag coefficient and the Oswald efficiency factor based on wind tunnel measurements from a previous campaign. With these corrections, the model predictions are much closer to the real-world measured data for SmartX-Alpha. However, high predictive accuracy is not the primary requirement of the method as for the purpose of testing an online black-box performance optimization framework, it is primarily important that the mechanics of the morphing wing model are similar to those of a real-world trailing-edge camber morphing, rather than that the exact prediction or optimal inputs are close to the real-world counterpart. The inherent trade-offs in the shape optimization of trailing-edge camber morphing wings are the reshaping of the spanwise lift distribution and the choice of camber versus angle of attack increments as the mechanics of lift production. With the distributed flap VLM model and the fitted correction functions respectively, these inherent trade-off mechanics are represented in the proposed model.

2. **What are suitable methodologies for an adaptable online wing shape optimization framework?**

   Due to the high costs of system evaluation, in-flight wing shape optimization methodologies should be very sample-efficient. Furthermore, any such methodology should also be robust to

measurement noise as aerodynamic force measurements are noisy in nature. Additionally, in order to allow for online operation, the computational cost of the method should be small. In other words, the computational time required should be small compared to the time required for measurement. Furthermore, a truly adaptable optimization framework should be able to learn any complex relationship between the system inputs and outputs, and also be able to update its knowledge, and by extension its shape solutions to changes in the system. Lastly, under practical considerations, it is highly desirable that the scope of the method is wide enough such that optimization can continuously take place even as the flight conditions vary. Ideally, there would be no need for model excitation maneuvers in flight.

In this thesis work, the required sample efficiency was realized by using a hybrid model-based evolutionary optimization strategy, thereby decoupling the learning and optimization objectives of the proposed framework. A low-cost surrogate or meta-model was learned online and continuously updated, and simultaneously the optimization procedure operated on this model to estimate the best wing shape within considerably less time than would be required for direct optimization of the system.

This on-board model comprised single-hidden-layer artificial neural networks (ANN). These are universal function approximators that can, with enough parameters, approximate any continuous non-linear function. These ANNs comprised Radial Basis Function (RBF) activation functions. These RBF networks are locally sensitive and more robust to noise than other ANNs. The neural network models are trained online on a global data set comprising both historical and recent measurement data. The Covariance Matrix Adaptation – Evolutionary Strategy (CMA–ES) optimization method was used for its robustness to noise and good global optimization performance. Together with the global on-board model, it allows the framework to predict well-performing wing shape and angle of attack combinations for various flight conditions without renewed input space exploration maneuvers.

3. **Compared to non-morphing wings of similar geometry, what increase in lift-to-drag ratio can be achieved on active trailing-edge camber morphing wings with online black-box wing shape optimization?**

   It has been shown in simulation that online black-box shape optimization can be used to increase the lift-to-drag ratios of rectangular planform active trailing-edge camber morphing wings at low speeds by $2.5\% - 14.6\%$ depending on the operational lift coefficient. The main mechanism of drag reduction for rectangular planform active morphing wings whose jig shapes do not comprise wash-outs is the reshaping of the spanwise lift distribution to more closely approach an elliptical distribution, which is theoretically ideal in terms of induced drag.

   The general trend in the camber versus angle of attack trade-off is that relatively high angles of attack with camber reducing morphing actuation are favored. This is owed to the relatively high degree of camber in the baseline NACA6510 airfoil. The trade-off is also consistent with the trends shown in the wind tunnel data from previous campaigns on which the augmentation of the aerodynamic model was based. Analysis of these measurements with constant spanwise camber morphing of various degrees showed that the maximum achievable camber reduction was aerodynamically most efficient for lift coefficients up to $0.85$ and that the second most camber reduction setting was most efficient for the remainder of the operational range. Nevertheless, the particular computed optimal solutions are not primarily of interest, but rather the ability of the black-box optimization method to identify these optima online regardless of the internal mechanics at play. Herein lies the inherent strength of black-box optimization, which requires no particular knowledge of the complex inner dynamics of the system to be optimized.

   Finally, the performance of the black-box online wing shape optimization strategy for active camber morphing wings has been validated in a wind tunnel experiment. It has been experimentally demonstrated that a lift-to-drag ratio increase of $8.4\%$ can be realized on SmartX-Alpha at $15$ m/s with a lift coefficient of $0.65$ using the proposed method. This constituted a $7.8\%$ decrease in aerodynamic drag compared to the wing jig shape. In contrast to the optimal wing shapes found in simulation, the experiment-based optimal wing shapes comprised a secondary reduction of the local camber near the wing root end. This trend is consistent with the theoretical effects of pressure leakage caused by the existence of a clearance gap between the wing root and the

fuselage surface. Additionally, the optimal shapes for other target lift coefficients have been predicted using the optimization framework in simulation with the most recent wind tunnel data. In the trade-off between the angle of attack and the overall amount of camber, these optimal solutions on average comprised higher amounts of camber, with the maximum amount of camber being close to the allowable upper limit, and generally lower angles of attack than the simulation-based solutions for the same lift coefficients. These trade-offs are consistent with the trends described in the literature for low Reynolds number flows.

4. **What improvements to the methodology are needed to transfer the shape optimization framework to in-flight operation, and what are the challenges?**

   To begin with, a major challenge to any online aerodynamic performance optimization methodology lies in the sensing of the aerodynamic forces. Naturally, no performance optimization is possible if the performance of the system cannot be accurately measured. The main challenge of black-box online global shape optimization for aerodynamic performance is to learn the complex relations between the shape inputs and the aerodynamic forces from few and scattered data in an adaptable way. Due to the significant measurement noise, even in steady-state aerodynamic flow situations, the system is naturally expensive to evaluate. Measurement times of around one minute are common practice in wind tunnel experiments, and it stands to reason that they would be even longer in the less-controlled in-flight environments outside laboratories. Hence it is of the utmost importance to the success of any black-box online wing shape optimization strategy to extract as much information as possible from each precious measurement.

   A proven approach to sample-efficiently generalizing the knowledge from individual data points is through the identification of surrogate models. Local gray-box models typically constrain the model structure to more efficiently estimate the parameters given the assumed model structure. Black-box surrogate models, which are not based on any assumptions in model structure and can generally approximate any non-linear mapping generally need more data points to be functional. On account of their scope global models also tend to need more data than their local scope counterparts. Hence, the primary challenge of the identification of a global black-box model for expensive-to-evaluate systems is the sheer amount of data required.

   In the work described in this thesis, this hurdle is overcome by retaining previously measured samples. This solution does come at a price. By combining new and old data to quench the surrogate model's "hunger for data", some adaptability or rather an adaptation speed is sacrificed. The heavier the reliance on older data in this trade-off, the less sensitive the model identification procedure will be to the latest measurements, and hence the lower the adaptation speed will be. However, the radial basis function neurons used in this thesis work are locally sensitive and are therefore able to locally adapt the model to new measurements. In contrast, with a global basis function surrogate model, each new data point would only be a drop in the proverbial ocean of historical data. In addition, the speed of adaptation was further improved with a targeted data management strategy. This strategy aims to retain historical data in scarce regions of the domain and favors new measurements over historical data in more densely populated areas of the domain.

   In future work, it may be interesting to investigate the merits of a weighted loss function based on sample aged. However, one should be cautious while doing so and guard against adversely affecting the generalizability of the model by assigning too much importance to few samples. Other improvements that should be made to the optimization framework are the inclusion of the Reynolds and Mach numbers as input variables, renewed hyper-parameter optimization using the newly gathered experimental data, and the implementation of trust-region bounds into the optimization constraints. These are discussed in greater detail in the following section.

**Research objective** Finally, the research objective presented in Sec. 1.2.2 is reviewed once more. The goal of the thesis research was to:

> "to realize the most aerodynamically efficient shape on a seamless active trailing-edge camber morphing wing by developing a black-box online-learning shape optimization method and testing the method in a wind tunnel experiment".

During the work described in this thesis, a black-box online shape optimization methodology has been developed and evaluated on both simulation models and on a seamless distributed morphing wing demonstrator in a wind tunnel experiment. The proposed methodology was successful in realizing a 7.8% reduction in the aerodynamic drag of the morphing wing while achieving the targeted lift coefficient in the wind tunnel experiment. With these results, the research objective has been achieved.

## 8.2. Recommendations

Based on the results of the work described in this thesis and the conclusions drawn, The following recommendations for future work are given.

1. As the inviscid aerodynamic model used in the simulation experiments only accurately model lift and lift-induced drag, it is recommended that a higher fidelity aerodynamic model is used in future work. While this type of model, with corrections based on previously gathered wind-tunnel data, was sufficiently accurate for the demonstration of the online optimization framework, the total drag prediction was not accurate enough to precisely predict the real-world optimal trade-off between camber and angle of attack. Especially if the scope of the framework is expanded to include high-speed flight, the aerodynamic model needs to be able to sufficiently accurately model Mach effects.

2. The on-board model is identified under the assumption that an invariant input-output relationship exists within the training data. However, in actuality, the relationship between the angle of attack and wing shape, and the steady-state lift and drag coefficients is also variable with changes in the Mach and Reynolds numbers. Therefore, the validity of the on-board model, in its current form, is limited to the region where the variation of this mapping with respect to the Reynolds and Mach numbers is negligibly small. In order to further expand the scope of the on-board model, and by extension that of the online optimization framework, the Mach and Reynolds numbers should be included as additional inputs to the on-board model in future work so that the scope of the method can encompass the entire flight envelope.

3. Since the strength of the neural network models strictly lies with the interpolation of data, and their ability to make generalizable predictions does not extend to the extrapolation of data, caution should be exercised in the “blind” optimization of their outputs. Parameter sweeps of trained models have shown that indeed the accuracy of the trained models is highest near the center of the domain spanned by the data, and a slight decreasing trend in predictive accuracy was observed towards the edges of the domain. Moreover, the predictive accuracy of the model starts to degrade much faster when crossing the boundaries of the training data set domain. The optimizer should therefore be guarded against exploiting model deficiencies in regions of low model accuracy. In this thesis work, the optimizer was restricted to operating strictly within the bounds of the training data set. However, by determining the trust region of the model online, more informed and adaptive optimization bounds could be maintained in future work. In this case, a low accuracy model would result in a more restricted optimization space, and therefore in less complex, but still improved wing shapes rather than in more complex but bogus shapes that yield undesirable performance.

4. In this thesis, the hyperparameter optimization was performed based on data from the aerodynamic model. However, the final goal of the on-board model was to approximate the input-output relationship of the data obtained during the wind tunnel experiments. Because ideal network parameters are data-set dependent, the hyperparameters used were likely not optimal for the task. However, a data set of various angles of attack and wing shapes, and their resulting steady-state aerodynamic coefficients exists has been gathered during the wind tunnel experiments described in this thesis. Therefore, in future work, this new data set, which is more representative of the real input-output mapping, should be used for hyperparameter optimization so that an even better model may be realized. In addition, it is recommended that a more advanced hyperparameter optimization strategy such as Bayresian Optimization – HyperBand is used to further improve the tuning of the on-board model hyperparameters.

5. The deviation from the target lift coefficient term in the cost function comprises the negative inverse of the error. This peculiar formulation is a remnant from a previous version, which was

designed as a reward function to be maximized. These inverse error relationships are undesirable since such terms can “blow up“ when the error takes on particularly small values. A small number was thus added to the denominator to limit the effect of this phenomenon. However, for the minimization objective, the negative inverse term is no longer needed. Instead, a more elegant form of the cost function is recommended for future work. The negative inverse of the squared lift error can simply be replaced by the squared lift error. This formulation is simpler and needs not to be safeguarded against blowing up.

6. Lastly, it is recommended that some changes in sensing equipment are made in future work to enable the application of the methodology to real-world aircraft. Since an external force-balance as used in the experimental set-up described in this thesis cannot be applied to a free-flying aircraft; these forces will have to be estimated using other on-board sensors. In the steady-state equilibrium conditions for which the optimization framework is proposed, the lift and drag forces balance the weight and thrust forces. These forces can be estimated using the gross aircraft weight, pitch angle, angle of attack, fuel consumption rate, and the engine’s Thrust-Specific Fuel Consumption (TSFC). The gross aircraft weight can be calculated from the aircraft’s gross take-off weight and the fuel burned. These are known quantities on commercial airliners, as are fuel consumption rate and the angles of attack and pitch. The TSFC varies with flight speed and altitude, which are also known. Additionally, the experimental IR cameras set-up used in this work can be converted to a fuselage-mounted in-flight vision tracking system as has been experimentally validated in [67].

# Bibliography


[1] Adrian L.R. Thomas. “The flight of birds that have wings and a tail: Variable geometry expands the envelope of flight performance”. In: *Journal of Theoretical Biology* 183.3 (1996), pp. 237–245. ISSN: 00225193. DOI: 10.1006/jtbi.1996.0217.

[2] Daochun Li et al. “A review of modelling and analysis of morphing wings”. In: *Progress in Aerospace Sciences* 100.June (2018), pp. 46–62. ISSN: 03760421. DOI: 10.1016/j.paerosci.2018.06.002. URL: https://doi.org/10.1016/j.paerosci.2018.06.002.

[3] L Michaelis and ML Menten. “Die kinetik der invertinwirkung Biochem Z 49: 333–369”. In: *Biochemische Zeitschrift* 49 (1913), pp. 333–369.

[4] Tigran Mkhoyan et al. “Design of a smart morphing wing using integrated and distributed trailing edge camber morphing”. In: *Smart Materials, Adaptive Structures and Intelligent Systems*. Vol. 84027. American Society of Mechanical Engineers. 2020, V001T04A023.

[5] Glenn B. Gilyard, Jennifer Georgie, and Joseph S. Bamicki. *Flight test of an adaptive configuration optimization system for transport aircraft*. Tech. rep. 1999. DOI: 10.2514/6.1999-831. URL: http://www.sti.nasa.gov.

[6] Glenn Gilyard. “In-flight transport performance optimization: An experimental flight research program and an operational scenario”. In: *NASA Technical Memorandum* 206229 (1997). ISSN: 04999320. DOI: 10.1109/dasc.1997.637236.

[7] Martin España and Glenn Gilyard. “Direct Adaptive Performance Optimization of Subsonic Transports: A Periodic Perturbation Technique”. In: *Nasa Technical Memorandum* March (1995).

[8] G. Gilyard. “Development of a real-time transport performance optimization methodology”. In: *NASA TM-4730*. 1996.

[9] Jacob Schaefer and Nelson A. Brown. “Peak-seeking optimization of trim for reduced fuel consumption: Architecture and performance predictions”. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. Boston, MA, 2013, pp. 1–15. ISBN: 9781624102240. DOI: 10.2514/6.2013-4856.

[10] Nelson Brown and Jacob Schaefer. “Peak-seeking optimization of trim for reduced fuel consumption: Flight-test results”. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. Boston, MA: American Institute of Aeronautics and Astronautics Inc., 2013. ISBN: 9781624102240. DOI: 10.2514/6.2013-5171.

[11] J.A. Snyman. “Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms”. In: 233 Spring Street, New York, NY 10013, USA: (Springer Science+Business Media, Inc., 2005, p. 24. ISBN: 0-387-24348-8.

[12] Marianne Jacobsen. “Real time drag minimization using redundant control surfaces”. In: *Aerospace Science and Technology* 10.7 (2006), pp. 574–580. ISSN: 12709638. DOI: 10.1016/j.ast.2006.05.002.

[13] Luis Miguel Rios and Nikolaos V. Sahinidis. “Derivative-free optimization: A review of algorithms and comparison of software implementations”. In: *Journal of Global Optimization* 56.3 (2013), pp. 1247–1293. ISSN: 09255001. DOI: 10.1007/s10898-012-9951-y.

[14] Raymond M. Kolonay and Franklin E. Eastep. “Optimal scheduling of control surfaces on flexible wings to reduce induced drag”. In: *Journal of Aircraft* 43.6 (2006), pp. 1655–1661. ISSN: 15333868. DOI: 10.2514/1.14604.

[15] Yvonne Ferrier, Nhan Nguyen, and Eric Ting. “Real-time adaptive least-squares drag minimization for performance adaptive aeroelastic wing”. In: *34th AIAA Applied Aerodynamics Conference*. Washington, D.C.: American Institute of Aeronautics and Astronautics Inc, AIAA, 2016. ISBN: 9781624104374. DOI: 10.2514/6.2016-3567.

[16] N. Nguyen et al. “Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing”. In: *SAE Technical Papers* Sept. 2015.September (2015). DOI: 10.4271/2015-01-2565.

[17] Nhan Nguyen and James Urnes. “Aeroelastic modeling of elastically shaped aircraft concept via wing shaping control for drag reduction”. In: *AIAA Atmospheric Flight Mechanics Conference August 2012*. Minneapolis, Minnesota: American Institute of Aeronautics and Astronautics Inc, AIAA, 2012. ISBN: 9781624101847. DOI: 10.2514/6.2012-4642.

[18] Nhan T. Nguyen et al. “Real-Time Adaptive Drag Minimization Wind Tunnel Investigation of a Flexible Wing with Variable Camber Continuous Trailing Edge Flap System”. In: *AIAA Aviation 2019 Forum*. Dallas, Texas: American Institute of Aeronautics and Astronautics (AIAA), June 2019. DOI: 10.2514/6.2019-3156.

[19] Nhan Nguyen and Juntao Xiong. “Real-time drag optimization of aspect ratio 13.5 common research model with distributed flap system”. In: *AIAA Scitech 2021 Forum*. virtual event: American Institute of Aeronautics and Astronautics (AIAA), Jan. 2021. ISBN: 9781624106095. DOI: 10.2514/6.2021-0069.

[20] John J. Bertin and Michael L. Smith. *Aerodynamics for engineers / John J. Bertin, Michael L. Smith.* eng. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 1998. ISBN: 0135763568.

[21] Peter D. Sharpe. “AeroSandbox: A Differentiable Framework for Aircraft Design Optimization”. MA thesis. Massachusetts Institute of Technology, 2021.

[22] Noud P.M. Werter et al. “Design and experiments of a warp induced camber and twist morphing leading and trailing edge device”. In: *24th AIAA/AHS Adaptive Structures Conference*. San Diego, California: publisher = American Institute of Aeronautics and Astronautics (AIAA), 2016. ISBN: 9781624103964. DOI: 10.2514/6.2016-0315.

[23] Alexander Bolonkin and Glenn B. Gilyard. *Estimated Benefits of Variable-Geometry Wing Camber Control for Transport Aircraft*. Tech. rep. October. 1999. URL: http://www.sti.nasa.gov.

[24] Ishan Bajaj, Akhil Arora, and M. M. Faruque Hasan. “Black-Box Optimization: Methods and Applications”. In: *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*. Ed. by Panos M. Pardalos, Varvara Rasskazova, and Michael N. Vrahatis. Cham: Springer International Publishing, 2021, pp. 35–65. ISBN: 978-3-030-66515-9. DOI: 10.1007/978-3-030-66515-9_2. URL: https://doi.org/10.1007/978-3-030-66515-9_2.

[25] J. Nelder and R. Mead. “A Simplex Method for Function Minimization”. In: *Comput. J.* 7 (1965), pp. 308–313.

[26] A. J. Booker et al. “Research Papers A rigorous framework for optimization of expensive functions by surrogates”. In: *Structural and Multidisciplinary Optimization* (1999), pp. 1–13. DOI: https://doi.org/10.1007/BF01197708DO.

[27] J E Dennis. “Mesh adaptive direct search algorithms for constrained optimization”. In: *SIAM Journal on Optimization* 17.1 (2006), pp. 188–217.

[28] Rommel G Regis et al. “Improved strategies for radial basis function methods for global optimization”. In: *Journal of Global Optimization* 37 (2007), pp. 113–135. DOI: 10.1007/s10898-006-9040-1.

[29] J. Mockus. *Bayesian Approach to Global Optimization: Theory and Applications*. Mathematics and its Applications. Springer Netherlands, 2012. ISBN: 9789400909090.

[30] Ishan Bajaj and M.M. Faruque Hasan. “A Projection-based, Data-Driven Method for High-Dimensional Black-Box Optimization”. In: *13th International Symposium on Process Systems Engineering (PSE 2018)*. Ed. by Mario R. Eden, Marianthi G. Ierapetritou, and Gavin P. Towler. Vol. 44. Computer Aided Chemical Engineering. San Diego, CA: Elsevier, 2018, pp. 973–978. DOI: https://doi.org/10.1016/B978-0-444-64241-7.50157-9. URL: https://www.sciencedirect.com/science/article/pii/B9780444642417501579.

[31] Robert L. Smith. “Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions.” In: *Operations Research* 32.6 (1984), pp. 1296–1308. ISSN: 0030364X. DOI: 10.1287/opre.32.6.1296.

[32] Arnon Boneh, Shahar Boneh, and Richard J Caron. “Constraint classification in mathematical programming”. In: *Mathematical Programming* 61 (1993), p. 61.

[33] C. J. P. Bélise, H. E. Romeijn and R. L. Smith. “Hit-and-Run Algorithms for Generating Multivariate Distributions Author ( s ): Claude J . P . Bélisle , H . Edwin Romeijn and Robert L . Smith Published by : INFORMS Stable URL : https://www.jstor.org/stable/3690278”. In: *Operations Research Letters* 18.2 (1993), pp. 255–266.

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (1983), pp. 671–680. DOI: 10.1126/science.220.4598.671. URL: https://www.science.org/doi/abs/10.1126/science.220.4598.671.

[35] Fabio Romeo and Alberto Sangiovanni-Vincentelli. “A theoretical framework for simulated annealing”. In: *Algorithmica* 6.1-6 (1991), pp. 302–345. ISSN: 01784617. DOI: 10.1007/BF01759049.

[36] J.H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975. ISBN: 9780262082136.

[37] N. Hansen and A. Ostermeier. “Completely derandomized self-adaptation in evolution strategies.” In: *Evolutionary computation* 9.2 (2001), pp. 159–195. ISSN: 10636560. DOI: 10.1162/106365601750190398.

[38] Nikolaus Hansen and Stefan Kern. “Evaluating the CMA Evolution Strategy on Multimodal Test Functions”. In: *Parallel Problem Solving from Nature - PPSN VIII*. Ed. by Xin Yao et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 282–291. ISBN: 978-3-540-30217-9.

[39] James Kennedy and Russell C. Eberhart. “Particle swarm optimization”. In: *Proceedings of the IEEE International Conference on Neural Networks*. Perth, WA, 1995, pp. 1942–1948.

[40] D R Jones, Computer Law, and Computer Law. “Lipschitzian Optimization Without the Lipschitz Constant”. In: *Journal of Optimization Theory and Applications* 79.1 (1993).

[41] M Björkman and K Holmström. “Global optimization using DIRECT algorithm in matlab”. In: *Advanced Modeling and Optimization* 1.2 (1999), pp. 17–37. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.6135.

[42] W. Huyer and A. Neumaier. “Global optimization by multilevel search”. In: *Journal of Optimization Theory and Applications* 75.2 (1992), pp. 423–432. ISSN: 00223239. DOI: 10.1007/BF00941477.

[43] Waltraud Huyer and Arnold Neumaier. “SNOBFIT – Stable Noisy Optimization by Branch and Fit”. In: *ACM Trans. Math. Softw.* 35.2 (July 2008). ISSN: 0098-3500. DOI: 10.1145/1377612.1377613. URL: https://doi.org/10.1145/1377612.1377613.

[44] Hans-Paul Paul Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. USA: John Wiley & Sons, Inc., 1993, p. 165. ISBN: 0471571482. DOI: 10.5555/529401.

[45] Robert Michael Lewis et al. “Direct Search Methods : Then and Now Operated by Universities Space Research Association”. In: *Science* 124.1-2 (2000), pp. 191–207.

[46] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[47] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. *CMA-ES/pycma on Github*. Feb. 2019. DOI: 10.5281/zenodo.2559634.

[48] Nikolaus Hansen. *CMA-ES Source Code*. URL: http://cma.gforge.inria.fr/cmaes_sourcecode_page.html.

[49] Nikolaus Hansen. “The CMA Evolution Strategy: A Comparing Review”. In: *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*. Ed. by Jose A Lozano et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–102. ISBN: 978-3-540-32494-2. DOI: 10.1007/3-540-32494-1_4. URL: https://doi.org/10.1007/3-540-32494-1_4.

[50] J. D. Anderson Jr. “Fundamentals of Aerodynamics”. In: 1221 Avenue of the Americas, New York,USA: McGraw-Hill, Inc., 2011, pp. 273, 416, 437, 462. ISBN: 978-1-259-01028-6.

[51] S. Estrada-Flores et al. “Development and validation of ”grey-box” models for refrigeration applications: A review of key concepts”. In: *International Journal of Refrigeration* 29.6 (2006), pp. 931–946. ISSN: 01407007. DOI: 10.1016/j.ijrefrig.2006.03.018.

[52] Peng Zhang. *Industrial control system simulation routines*. First Edition. Peng Zhang, 2010, pp. 781–810. ISBN: 9781437778076. DOI: 10.1016/b978-1-4377-7807-6.10019-1. URL: http://dx.doi.org/10.1016/B978-1-4377-7807-6.10019-1.

[53] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[54] Abdel-Nasser Sharkawy. “Principle of Neural Network and Its Main Types: Review”. In: *Journal of Advances in Applied & Computational Mathematics* 7.1 (2020), pp. 8–19. DOI: 10.15377/2409-5761.2020.07.2.

[55] Laith Alzubaidi et al. *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. Vol. 8. 1. Springer International Publishing, 2021. ISBN: 4053702100444. DOI: 10.1186/s40537-021-00444-8. URL: https://doi.org/10.1186/s40537-021-00444-8.

[56] Tiantian Xie, Hao Yu, and Bogdan Wilamowski. “Comparison between traditional neural networks and radial basis function networks”. In: *2011 IEEE International Symposium on Industrial Electronics*. Langkawi, Malaysia, 2011, pp. 1194–1199. DOI: 10.1109/ISIE.2011.5984328.

[57] Ch Sanjeev Kumar Dash et al. “Radial basis function neural networks: A topical state-of-the-art survey”. In: *Open Computer Science* 6.1 (2016), pp. 33–63. ISSN: 22991093. DOI: 10.1515/comp-2016-0005.

[58] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. “Randomized smoothing for (parallel) stochastic optimization”. In: *Proceedings of the IEEE Conference on Decision and Control*. Maui, Hawaii: Institute of Electrical and Electronics Engineers (IEEE), 2012, pp. 5442–5444. DOI: 10.1109/CDC.2012.6426698.

[59] Matthew D. Zeiler. “ADADELTA: An Adaptive Learning Rate Method”. In: (2012). arXiv: 1212.5701. URL: http://arxiv.org/abs/1212.5701.

[60] Diederik P. Kingma and Jimmy Lei Ba. “Adam: A method for stochastic optimization”. In: San Diego, CA, 2015, pp. 1–15. arXiv: 1412.6980.

[61] Stefan Falkner, Aaron Klein, and Frank Hutter. “BOHB: Robust and Efficient Hyperparameter Optimization at Scale”. In: *35th International Conference on Machine Learning, ICML 2018*. Vol. 4. Stochholm, Sweden, 2018, pp. 2323–2341. ISBN: 9781510867963. arXiv: 1807.01774.

[62] I. Sobol. “On the distribution of points in a cube and the approximate evaluation of integrals”. In: *Ussr Computational Mathematics and Mathematical Physics* 7 (1967), pp. 86–112.

[63] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 00401706. DOI: 10.2307/1268522.

[64] Sergei Kucherenko, Daniel Albrecht, and Andrea Saltelli. “Exploring multi-dimensional spaces: A comparison of Latin hypercube and quasi Monte Carlo sampling techniques”. In: *arXiv preprint arXiv:1505.02350* (2015).

[65] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[66] Xuerui Wang et al. “Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation”. In: *Journal of Guidance, Control, and Dynamics* 44.9 (Sept. 2021), pp. 1649–1662. ISSN: 1533-3884. DOI: 10.2514/1.G005870. eprint: 2012.14520.

[67] Tigran Mkhoyan, Coen C de Visser, and Roeland De Breuker. “Adaptive state estimation and real-time tracking of aeroelastic wings with augmented kalman filter and kernelized correlation filter”. In: *AIAA Scitech 2021 Forum*. virtual event, 2021, p. 0666.