Development of a deep neural network framework for sailplane cross-country performance optimisation

V le Roux

orcid.org/ 0000-0002-4636-6847

Thesis accepted in fulfilment of the requirements for the degree *Doctor of Philosophy in Engineering with Computer and Electronic Engineering* at the North-West University

Promoter:	Prof MH Davel
Co-Promoter:	Dr JJ Bosman

Graduation: August 2024

Declaration

I, Vincent le Roux, hereby declare that the dissertation entitled "Development of a deep neural network framework for sailplane cross-country performance optimisation" is my own original work and has not already been submitted to any other university or institution for examination.

V. le Roux

Student number: 25023438

Signed on the 21 st day of April 2024 at Midrand.

Acknowledgements

This research was performed within the Multilingual Speech Technologies (MuST) research group of the North-West University, which is a member of the Centre for Artificial Intelligence Research (CAIR) of the Department of Science and Innovation. It was supervised by Professor Marelie Davel and Doctor Johan Bosman.

To Professor Marelie I would like to thank for her constant and unwavering support over the past five years. Without her guidance and expertise in deep learning, optimisation, and technical writing this research would not have been possible.

I would also like to thank Doctor Bosman for his technical guidance and for sharing with me his subject matter expertise throughout this research – without whom the final result would not have been achievable.

To my family, specifically my father, mother, and sister I would like to thank for always believing in me and instilling in me the notion that anything is possible.

To my lovely wife, I would like to thank for your relentless belief and support. You kept me inspired and motivated in times when I needed it the most – I cannot thank you enough for your love and support.

Finally, to my heavenly Father, I would like to honour and give glory for enabling me to carry out this research over the past five years and for the constant and numerous provisions that made all of this possible.

Abstract

Traditional sailplane cross-country performance optimisation is a slow and computationally expensive task as it requires the integration of multiple simulation packages and function evaluations for each span-wise station and flight condition. Traditional methods also prohibit expert input to this optimisation process. Deep neural networks, with their powerful generalisation capability, provide a possible solution for replacing the computationally expensive simulation packages with lower fidelity surrogate models, ultimately reducing the time spent on the preliminary sailplane design and optimisation phase.

In this research, a deep learning framework for sailplane cross-country performance optimisation is developed to address these shortcomings of the traditional model. Various modelling techniques are employed to build a deep learning based system that is accurate, computationally efficient, and which enables expert input upfront to the optimisation process. For the expert input, deep learning modules are developed, enabling airfoil generation from parsimonious shape and structural variables. Constraining this parsimonious variable search space, therefore, enables intuitive constraint settings on the wing and individual airfoil designs. It is shown that the deep learning framework not only outperforms the traditional model in terms of computational efficiency but is also able to find wing/airfoil combinations that outperform defined baseline designs whilst adhering to all constraints imposed on the final design. Specifically, the efficacy and validity of the proposed deep learning framework is shown where it is applied to realise a 0.9% average increase in the cross-country performance of the JS4 sailplane – a State-Of-The-Art (SOTA) standard class sailplane and currently one of the worlds best performing models.

This deep learning framework therefore facilitates rapid, accurate, and targeted wing/airfoil design explorations and ultimately reduces the time spent on design, especially in the preliminary design phases, and has been shown to possess the capability of cross-country performance optimisation of SOTA standard class sailplane designs.

Keywords: *cross-country, optimisation, deep learning, sailplane, parametrisation*

Contents

List of Figures

List of Tables

List of Abbreviations

BLSTM Bidirectional Long Short Term Memory

GAN Generative Adversarial Network

SPM Simple Parametric Method

MAE Mean Absolute Error

MSE Mean Squared Error

MAPE Mean Absolute Percentage Error

GA Genetic Algorithm

AoA Angle of Attack

RE Reynolds

MA Mach

LE Leading Edge

TE Trailing Edge

LLT Lifting Line Theory

GMM Gaussian Mixture Model

BGMM Bayesian Gaussian Mixture Model

LSTM Long Short Term Memory

VAE Variational Auto Encoder

CVAE Conditional Variational Auto Encoder

CGAN Conditional Generative Adversarial Network

WGAN Wasserstein Generative Adversarial Network

ANN Artificial Neural Network

CNN Convolutional Neural Network

MLP Multi-Layer Perceptron

MAPE Mean Absolute Percentage Error

SHAP Shapley Additive Explanations

AR Aspect Ratio

DSM Deep Learning Sailplane Model

TSM Traditional Sailplane Model

HTSR Theory of Heavy Tailed Self-Regularisation

IPDLM Inverse 2D Performance Deep Learning Module

PDLM 2D Performance Deep Learning Module

Cm 2D Pitching Moment Coefficient

Cl 2D Lift Coefficient

CL 3D Lift Coefficient

Cd 2D Drag Coefficient

CD 3D Drag Coefficient

PPS Predictive Power Score

PSALL Per Sample Average Log Likelihood

PSO Particle Swarm Optimisation

t-SNE t-distributed Stochastic Neighbor Embedding

RNN Recurrent Neural Network

UIUC University of Illinois Urbana-Champaign

CST Class-Shape Transformation

SVD Singular Value Decomposition

ML Machine Learning

DL Deep Learning

CP Conformal Prediction

CQR Conformalised Quantile Regression

QR Quantile Regression

RMT Random Matrix Theory

CFD Computational Fluid Dynamics

RANS Reynolds-averaged Navier–Stokes equations

FVM Finite Volume Method

RBF Radial Basis Function

PCA Principal Component Analysis

MAC Mean Aerodynamic Chord

SOTA State-Of-The-Art

DSR Design Science Research

Chapter 1

Introduction

1.1 Problem overview

Sailplane cross-country flight is a specialised form of gliding that involves flying long distances using the lift generated by the strategic use of updrafts (Thomas, 1999). These updrafts, also known as thermals, are columns of rising air that can be found in certain meteorological conditions and are often used by glider pilots to gain altitude and extend the range of their flights. Sailplane pilots gain altitude by flying in circles within the thermal and then using that altitude to glide to the next thermal. By carefully navigating from thermal to thermal, pilots can fly for hundreds of kilometres without needing an engine. An idealised depiction of sailplane cross-country flight can be seen in Figure 1.1.

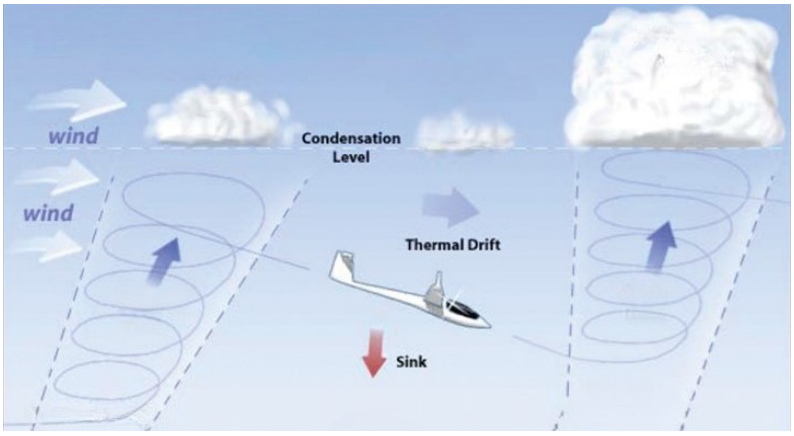


Figure 1.1: Idealised depiction of sailplane cross-country flight by using thermal updrafts. Adapted from (Andersson, 2009)

.

When designing a sailplane for cross-country flight the main objectives are endurance and range to ensure that a sailplane can efficiently extract energy from a given thermal to facilitate the successive thermal-to-thermal gliding action. Competition sailplanes should be optimised for long-distance flight at both high and low flight speeds. That is, the glider should be designed to yield low form, profile, interference, and induced drag over the entire flight envelope. Hence, the ultimate goal in sailplane optimisation is to achieve high lift-to-drag ratios over the entire flight envelope, which facilitates high average cross-country speeds.

Cross-country sailplane design is a multi-objective optimisation problem, which is computationally expensive and typically requires multiple manual interventions. In a typical sailplane wing optimisation process, the wing is divided into a finite number of sections (or stations), each with its own airfoil, length, and chord. This gives rise to different performance attributes at each station which, along with wing properties such as taper ratio, sweep, aspect ratio, and washout angle, results in a varying lift distribution along the span. Finding the correct combination of airfoil parameters and wing attributes for each station is not a trivial task and requires the integration of multiple modules, including parametrisation models and Computational Fluid Dynamics (CFD) and/ or panel code simulations (as will be introduced later in Sections 2.4 and 2.6 respectively).

More specifically, in the traditional approach to sailplane optimisation, there are five distinct modules:

1. A geometric airfoil shape generation function: used for defining an airfoil in terms of either parametric coefficients or x and y coordinates.
2. An airfoil performance analysis module: used to analyse the performance of a given airfoil at a defined operating condition.
3. A sailplane performance analysis module: used to analyse the flight performance of a sailplane for a given operating condition.
4. A cross-country performance analysis module: used to calculate the average cross-country speed of a sailplane for a given cross-country flight task.
5. A search model: used for design space exploration to facilitate multi-objective sailplane cross-country performance optimisation.

In the traditional framework, these five independent modules are integrated and run in conjunction until convergence within the defined constraints. This approach, however, is computationally expensive and prone to having numerical issues due to the underlying architecture of the CFD codes, panel codes, and parametrisation models.

Employing deep learning can help alleviate the computational burden. Deep learning is a form of machine learning that makes use of neural networks to learn from vast amounts of data. Neural networks are models that consist of layered nodes, each with trainable weights combined with an activation function to implicitly account for non-linearity (Goodfellow et al., 2016; Raghu & Schmidt, 2020). Deep learning finds applications in various domains, such as image and speech recognition, natural language processing, and predictive analytics. Its growing popularity is attributable to its capacity to handle large and intricate datasets, and it has proven to be extremely effective in improving the accuracy and speed of many tasks that were previously thought to be beyond the capabilities of machines.

With the recent advances in computational power, modelling techniques, and the general availability of large datasets, deep learning has seen extensive success in the field of aeronautics with surrogate modelling and airfoil optimisation at the core of numerous research papers (Abutunis et al., 2019; Bartoli et al., 2019; X. Du et al., 2021; J. Li et al., 2022; Sekar et al., 2019; Tao & Sun, 2019; Tao, Sun, & Guo, 2020; S. Wang et al., 2018; Yonekura et al., 2022). However, utilising these methods in sailplane cross-country optimisation applications has not, to the best of the authors' knowledge, received wide attention. The two main reasons for this are: (1) the lack of a well-defined framework that unifies the various modules (the parametrisation models, CFD and panel codes, the wing and fuselage design components, and the sailplane performance models), and (2) the scarcity of large datasets that are required to develop a surrogate model (or set of models) fit for multi-objective optimisation (Karali et al., 2020). Deep learning methods, therefore, have the capability to redefine the traditional optimisation pipeline by enabling user-guided, intuitive constraint settings as input to the optimisation pipeline. This has not, however, been explored in the current literature.

In this research, the focus is therefore on the development of a general framework for user guided multi-objective sailplane cross-country performance optimisation that utilises deep learning to replace the airfoil performance analysis module of the traditional model. In order to achieve this, airfoil parametrisation elements of the traditional framework are redesigned, and an additional model is developed to regenerate the airfoil from its parameterised form. This framework and the subsequent machine- and deep-learning models, are developed in such a way as to allow for intuitive constraint setting on the wing and airfoil profiles as input to the optimisation pipeline – further motivating the versatility of the framework.

1.2 Problem statement

The traditional approach to sailplane cross-country performance optimisation is a complex and computationally expensive process. This is because the multi-objective nature of the optimisation problem requires a large number of computationally expensive numerical simulations, typically executed on disparate systems. These simulations can be prohibitively expensive, limiting the options considered during the design process, and are also prone to have convergence issues – further limiting the exploration of a large design space. The traditional implementation of sailplane cross-country optimisation techniques also does not allow for intuitive user constraint setting on the wing geometry and airfoil profiles along the wing span. This therefore limits the use of expert input to the overall optimisation pipeline and possibly leads to unfavourable results through exploration and exploitation of undesirable design combinations.

1.3 Main research components

In this section, details are provided on the two main components of this research i.e. the Traditional Sailplane Model (TSM) and the Deep Learning Sailplane Model (DSM). First, the TSM is introduced with a specific focus on its individual modules. From here an overview of the DSM is given with a focus on the alterations required to the TSM to ensure efficient, robust, and targeted sailplane cross-country optimisation is possible.

1.3.1 Overview of the TSM

As mentioned, a sailplane cross-country model is used to determine the flight performance of a given sailplane configuration for a given weather condition. The flight performance is typically measured as the average cross-country speed over a fixed distance. The sailplane cross-country model is usually an idealised model in which the aim is not to model each minute detail of cross-country flight, but rather to form a solid basis of cross-country

performance comparison grounded in the principles of flight mechanics.

To the best of the authors' knowledge, to date, there is no industry-standard single integrated process available to design and optimise a sailplane for maximum average cross-country speed. Typically, sailplane designers make use of disparate systems to design and optimise each sailplane element in isolation before combining the individual designs to see what the aggregate effect is on the sailplane's cross-country performance. In Figure 1.2 an example is shown of an end-to-end traditional sailplane cross-country design and optimisation process. This figure depicts how each of the sub-modules is integrated and where the outputs of one module are used as input to the next. Table 1.1 gives a brief discussion of each of the modules depicted in Figure 1.2.[1]

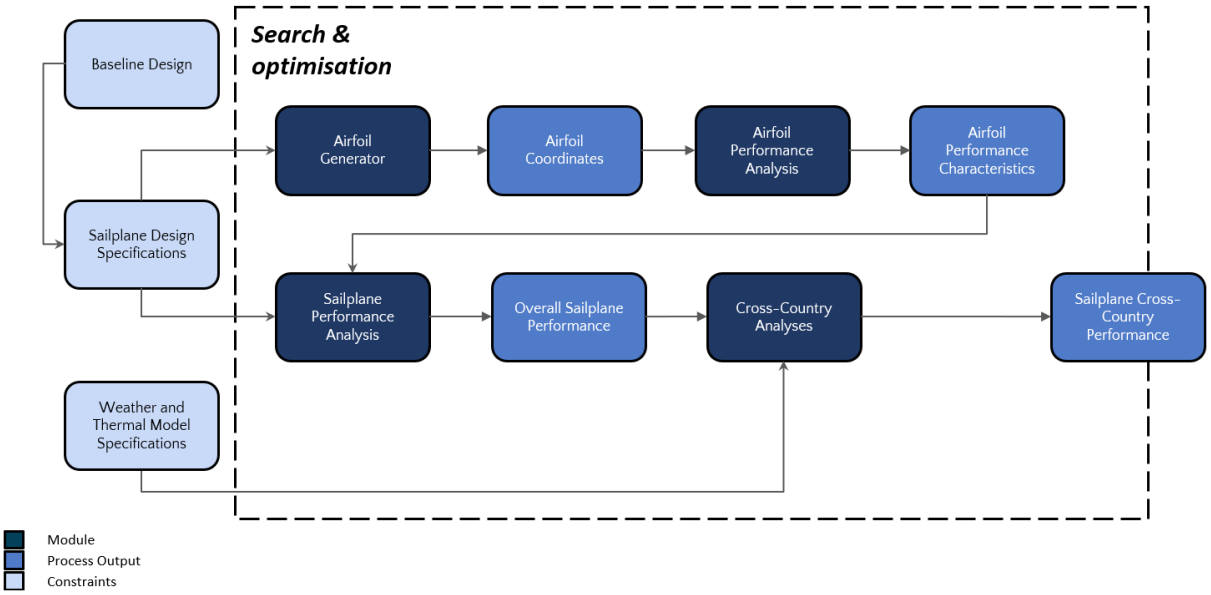


Figure 1.2: Depiction of a traditional sailplane design and optimisation process

[1]In Table 1.1, the baseline design, sailplane designs specifications, and weather and thermal model specifications are not depicted seeing that these are not independent modules but rather inputs to the traditional sailplane cross-country design and optimisation process. Here the baseline design refers to an existing sailplane design that needs to be optimised and is used to set design constraints. The sailplane design specifications refer to the upper and lower constraints for the wing geometry and airfoil profiles along the wing span. Finally, the weather and thermal model specifications refer to the cross-country task definition in terms of thermal types, thermal radii, cross-country distance, and flight distance allocation to each thermal type.

Table 1.1: High-level description of sailplane cross-country design and optimisation modules

Process	Description	Example input	Example output
Airfoil generator	Used to generate an airfoil profile.	Control points, parametric coefficients, latent vector (i.e. noise vector), mode shapes, etc.	Airfoil shapes in the form of x and y coordinate pairs.
Airfoil performance analysis	Used to determine the 2D performance of a given airfoil for a given operating condition.	Airfoil x and y coordinates and operating conditions i.e. the Reynolds number, Mach number, and the Angle of Attack (AoA).	Airfoil performance characteristics i.e. profile drag coefficient, lift coefficient, top and bottom transition, lift-to-drag ratio and moment coefficient.
Sailplane performance analysis	Used to determine the 3D performance of the entire sailplane i.e. the wing, winglet, tail and fuselage.	2D airfoil performance per station, wing design specifications, and fuselage, tail and winglet drag specification.	3D sailplane performance i.e. total sailplane drag and lift at a given operating condition.
Cross-country analysis	Used to determine the average cross-country speed.	3D sailplane performance, thermal profiles, and weather model definition are used as input.	Sink Speed, sailplane climb speed for various thermal models and turning radii, and the average cross-country speed.

1.3.2 Overview of the DSM

Recall Figure 1.2 depicting the individual components of the TSM embedded in a search and optimisation pipeline. As mentioned, this traditional approach has the disadvantage of being computationally expensive and having the known drawback of not allowing for intuitive constraint setting on the wing and airfoil designs as input to the optimisation pipeline. Addressing these shortcomings involves developing the DSM, as depicted in Figure 1.3. The main changes to the traditional framework are the replacement of the airfoil generator and airfoil performance analysis modules with alternative Deep Learning (DL) or Machine Learning (ML) models, as well as the addition of an airfoil shape conversion module.

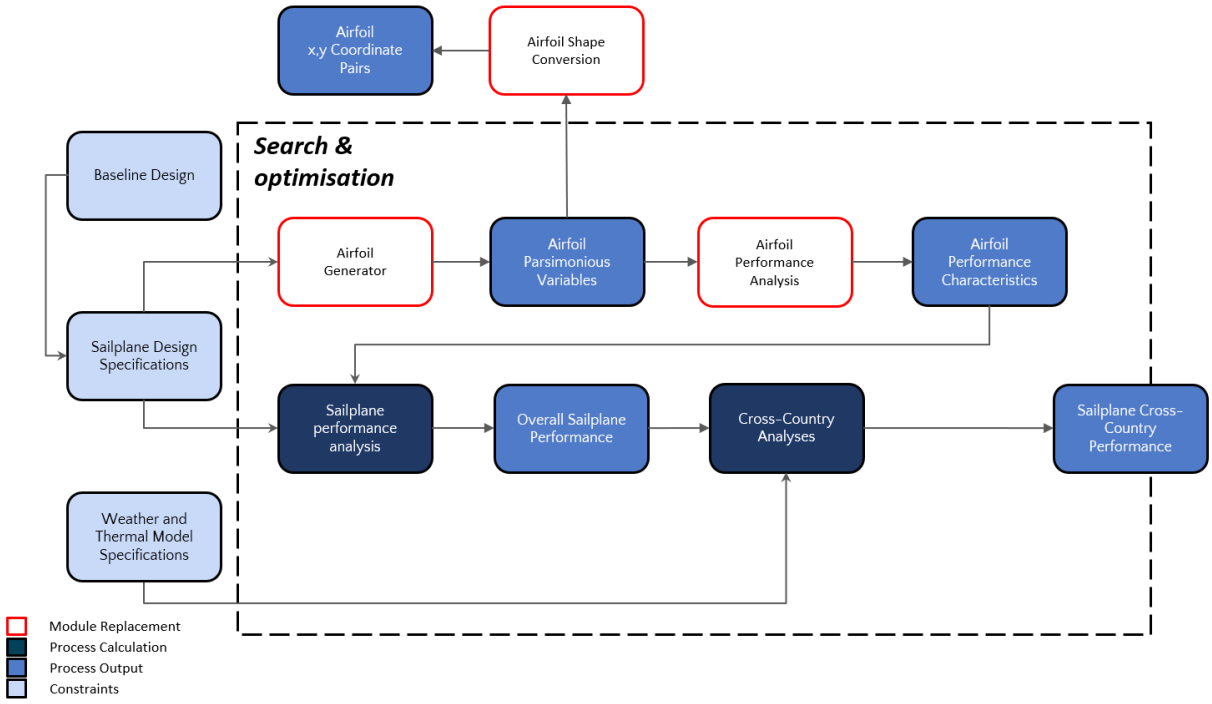


Figure 1.3: Depiction of the proposed deep learning integrated sailplane design and optimisation process with the changes required to the traditional sailplane model highlighted in red

To achieve the end goal of efficient and targeted sailplane cross-country performance optimisation, each of the developed models needs to adhere to a very specific set of criteria. For the airfoil generator module, this criterion is the efficient and accurate generation of airfoil profiles by means of specifying a set of intuitive airfoil characteristics. This allows for the explicit constraint setting on the airfoil characteristics as input to the sailplane cross-country optimisation pipeline. For the airfoil performance analysis module, this criteria is the efficient and accurate prediction of the conventional and non-conventional airfoil performance characteristics. The airfoil performance analysis module should do this by mapping the parsimonious airfoil variables generated by the airfoil generator module to the desired 2D airfoil performance characteristics if supplied with the appropriate operating conditions. Finally, the criteria for the airfoil shape conversion module is the accurate mapping between the raw airfoil x and y coordinate domain to the parsimonious variable domain to allow for accuracy validation of the DSM with the TSM as well as validation of the optimisation pipeline outputs.

1.4 Objectives of the study

The overall objective of this study is to develop a single efficient system that can optimise a SOTA standard class sailplane for cross-country flight by setting explicit constraints on the wing geometry and airfoil profiles. The study identifies five main objectives, which include developing and evaluating sub-components, developing a parsimonious variable extraction method, data mining, developing surrogate models using an Artificial Neural Network (ANN), and defining a framework for DL-based optimisation. This study uses machine learning models such as Bayesian Gaussian Mixture Models (BGMMs) and deep learning models such as ANNs to solve the different problems at hand and focuses on building a system that facilitates intuitive sailplane cross-country performance optimisation. The following sections provide a detailed overview of each objective.

1.4.1 Objective 1: Develop and evaluate required sub-components

In objective one, the focus is on sequentially building the TSM by first developing the required sub-components. As mentioned in Section 1.1 and Section 1.3.1, these sub-components include an airfoil parametrisation method, an airfoil performance analysis model, a sailplane performance analysis model, and a cross-country performance analysis model. The main sub-objectives of objective one can be summarised as follows:

1. Select and implement an efficient airfoil parametrisation method that will be used for airfoil generation.
2. Select a computational flow solver that will be used to predict the 2D performance characteristics of an airfoil profile.
3. Develop a sailplane performance analysis module that will be used to calculate the total performance of any given sailplane configuration.
4. Develop a cross-country performance analysis module that will be used to predict the average cross-country speed for a given flight task and sailplane definition.

5. Integrate the sub-components depicted in items 1-4 to form the TSM model.

Once the traditional model is built, the next focus is on validating the accuracy of this model. This involves comparing the outputs of the traditional model with both experimental and simulation results for a wide range of operating conditions applicable to sailplane cross-country flight.

1.4.2 Objective 2: Extracting intuitive shape and structural variables from airfoil profiles

This objective is concerned with the development and verification of the parsimonious [2] variable extraction framework. It is used to derive the intuitive shape and structural characteristics of an airfoil from its x and y coordinate descriptions. Having an airfoil defined in terms of these parsimonious variables would ultimately allow for the intuitive and explicit constraint settings on the various airfoil profiles along the wingspan – thus allowing for expert input to the design and optimisation process. Therefore, a core part of this objective is to identify which intuitive variables should be used to describe an airfoil profile. This objective is also focused on how these identified parsimonious variables can be extracted numerically from the raw coordinate definition of an airfoil profile, generated with any constructive or deformative airfoil generation method.

1.4.3 Objective 3: Airfoil performance and shape data mining

In objective three a data mining method is developed that enables the acquisition of the required 2D performance and airfoil shape datasets to facilitate the development of the airfoil generator module, the airfoil performance analysis module, and the airfoil shape conversion module. The objective here also includes the development of methods to eliminate impractical airfoil shapes from the final datasets to reduce noise and hence

[2] In this research, airfoil profile characteristics such as shape and structural features with explicit and intrinsic meaning are referred to as parsimonious variables.

further enable efficient model development. A sub-objective here is to gain a deeper understanding of the 2D performance dataset by means of various methods of statistical analysis.

1.4.4 Objective 4: Model development

In objective four, the data mined in objective three is used to develop the three required modules i.e. the airfoil generator module, the 2D Performance Deep Learning Module (PDLM), and the shape conversion module.

For the airfoil generator module, the main goal is to develop a method that is capable of generating an airfoil profile in terms of lower-order parsimonious variables. The parsimonious variable output of this model should be able to capture the local and global sequential structure as present in the raw x and y coordinate forms.

For the PDLM development, the main objective is to have a deep learning-based module that is able to fully replace the functioning of the airfoil performance analysis module as depicted in Figure 1.2. This module should allow for efficient and accurate prediction of 2D airfoil performance characteristics given an airfoil's description in terms of its parsimonious variables and operating condition specification. These 2D airfoil performance characteristics include:

1. 2D Lift Coefficient (C_l),
2. 2D Drag Coefficient (C_d),
3. 2D Pitching Moment Coefficient (C_m),
4. Lift-curve slope*,
5. Zero-lift AoA*,
6. Top surface transition location,
7. Bottom surface transition location, and

8. Maximum $C_l{}^*$.[3]

Finally, for the airfoil shape conversion module, the main goal is to develop a model capable of mapping the lower-order parsimonious variables to the higher-order raw x and y coordinates. Here the main tasks of this objective are model architecture optimisation and accuracy validation.

1.4.5 Objective 5: Define a framework for DL-based optimisation

Objective five can be broken into five distinct sub-objectives. The first sub-objective is centered around the integration of the developed modules into the TSM to ultimately form the DSM. The second sub-objective is concerned with the verification of the accuracy of the DSM to ensure accurate and efficient sailplane cross-country performance prediction results. The third sub-objective is concerned with embedding the DSM into an optimisation module to allow for constrained multi-objective sailplane cross-country performance optimisation of a SOTA standard class glider. Here the efficacy of the optimisation module needs to be verified by means of various case studies to establish if the optimisation module makes the correct sailplane design decisions in the optimisation process and to identify any discrepancies or problem areas of the DSM and ultimately the underlying modules. The fourth objective is to use the optimisation module in an end-to-end cross-country performance optimisation routine. In order to achieve this, all the discrepancies addressed in the DSM throughout its development and evaluation are first addressed in such a way that explicitly constrained cross-country performance optimisation of the JS4 sailplane is possible. Finally, in objective five, the optimised design is subjected to various validation steps to ensure that all upfront explicitly defined constraints are adhered to and to verify that the optimised design indeed has superior performance to the JS4 baseline design over the desired operating range.

[3]In this research, the 2D airfoil performance characteristics marked with * are referred to as non-conventional 2D performance characteristics seeing that they are not traditionally calculated directly by commercially available panel codes or CFD software.

1.5 Research framework

The research framework employed throughout this study is the Design Science Research (DSR) (Ken Peffers & Chatterjee, 2007) framework. DSR is a problem-solving approach that aims to address real-world problems and its outcomes consist of newly designed artifacts and design knowledge. The DSR framework is cyclical in nature such that the artifacts are first designed and developed after which they are continually tested, evaluated, and refined to ensure that the final system solves the identified research problems. In this research, DSR is applied in three distinct cycles: (1) the component design cycle, (2) the system design cycle, and (3) the system refinement cycle. Each cycle has its individual objectives, requirements, implementation, validation, and evaluation criteria.

1.5.1 Component design cycle

The component design cycle is focused on the development of each of the modules as required in the DSM (see Figure 1.3). In Table 1.2 these modules with their respective objectives, requirements, and success criteria can be seen. The success of the component design cycle is the delivery of the required artifacts (modules) that have all of the desired characteristics.

Table 1.2: Overview of objectives, requirements, and success criteria for the respective components developed during the component design cycle

Criteria/ Components	Parameterisation framework	Data mining pipeline	Airfoil generator module	2D Airfoil performance analysis module	Airfoil shape conversion module	TSM
Objectives	Represent an airfoil profile with parsimonious and intuitive variables	Mine airfoil data in terms of raw x and y coordinates, parsimonious variables, and 2D airfoil performance for a large and diverse group of airfoils	Generate a realistic airfoil profile expressed as a set of lower-order parsimonious variables	Map the parsimonious airfoil variables, generated by the airfoil generator module, to a set of 2D airfoil performance characteristics	Map airfoil profiles from the lower-order parsimonious variable definition to the higher-order raw x and y coordinate definition	Calculate the average cross-country speed of a standard class sailplane given the design and flight condition specification
Requirements	Padulo complete airfoil parameterisation such that the method is complete, parsimonious, orthogonal, flawless, and intuitive	Efficient and robust data mining for realistic airfoils generated with any constructive or destructive method over the Reynolds number, Mach number, and angle of attack ranges of interest	Computationally efficient, numerically robust, and accurate generation of diverse airfoil profiles such that the resulting profiles are realistic	Accuracy, numerical stability, and efficiency in 2D airfoil performance prediction for realistic airfoil profiles over a wide range of Reynolds numbers, Mach numbers, and angles of attack	Efficiency and accuracy in airfoil mapping from a low-dimensional to a high-dimensional space such that the same airfoil profile is defined in both data domains	Accurate prediction of the average cross-country speed and performance curve over the operating conditions typically present in standard class sailplane cross-country flight
Validation	Analyse the differences between the parsimonious variable extraction framework calculations and the XFOIL calculations for a diverse group of airfoil profiles	Analyse the resulting data set to ensure realistic airfoil profiles result with the desired shape characteristics; sampled from the operating condition range of interest	Evaluate the generative capacity of the developed model in terms of feature distribution and feature co-variance	Measure the difference between the 2D airfoil performance analysis module predictions and the XFOIL predictions for a large set of out-of-sample airfoils over diverse operating conditions	Analyse the Mean Absolute Error (MAE) between the x and y coordinates of the generated and actual airfoil profiles in the hold-out data set	Compare the average cross-country speed predictions of the TSM and the relevant predicted performance curves with the predictions made by the Jonker Sailplane's performance calculation software i.e. the JSCode and Idaflieg experimental results
Success	Efficient extraction of the identified parsimonious variables from raw x and y coordinates with acceptable agreement with XFOIL calculated values	Large data set results with diverse and realistic airfoil profiles within the desirable upper and lower shape constraints; with 2D performance and operating conditions sampled from the required distributions	Able to efficiently and accurately generate realistic and diverse airfoil profiles in terms of the chosen parsimonious variables which resemble the training data feature distribution and feature covariance	Comparable and highly correlated airfoil performance predictions between the developed 2D airfoil performance analysis module and XFOIL simulations	Efficient and accurate approximation of out-of-sample airfoils i.e. the airfoil shape conversion module can sufficiently predict the raw x and y coordinates of an unseen realistic airfoil profile from a parsimonious variable definition	A high level of agreement and correlation between the TSM predictions, the JSCode predictions, and the Idaflieg experimental results

In each case, the component is designed, developed, evaluated against the success factors, and refined if necessary until the requirements have been met.

1.5.2 System design cycle

The system design cycle has the objective of developing the end-to-end system i.e. the optimisation module. In this research, this objective is achieved in two sequential steps. These two steps include the design, development, and validation of: (1) the DSM and (2) the optimisation module.

For the first objective, the developed modules of Table 1.2 are integrated with the TSM to form the DSM. The DSM has the objective of providing an accurate prediction of a sailplane's average cross-country performance – for a given design specification and a flight condition. The main requirement of the DSM is therefore to be able to specify any conventional standard class sailplane design. This means that the underlying sub-components should allow for the same airfoil and wing definition in both the TSM and DSM. The DSM is validated by comparing its average cross-country performance predictions with the average cross-country performance predictions of the TSM for a diverse range of flight conditions and glider designs. Success here is defined as comparable predictions and a high degree of correlation between the TSM and DSM with clear identification of any underlying causes of inaccuracies due to the assumptions made in the respective sub-components design and development.

In the final objective of the system design cycle, the DSM is embedded into an optimisation framework to form the optimisation module. The objective of the optimisation module is to enable global sailplane cross-country performance optimisation with the requirement of being able to explicitly define constraints as input to the optimisation process. Validating the capability of the optimisation module to make design decisions aligned with known sailplane design patterns involves investigating three different optimisation case studies. These case studies have the following three goals:

1. Determine whether the optimisation module can effectively identify optimal wing designs for different airfoil configurations.

2. Ascertain if optimisation is achieved whilst adhering to all explicitly defined user constraints.

3. Identify any discrepancies in the DSM, the optimisation module, and the underlying sub-modules that would prohibit optimisation of a SOTA standard class sailplane design.

For the system design cycle, success is defined as a final delivered artifact (the optimisation module), capable of constrained global sailplane performance optimisation – with any underlying sub-component problems identified.

1.5.3 System refinement cycle

The system refinement cycle has the main objective of rectifying any undesirable results of the underlying sub-components and/or the optimisation module, that prevents SOTA standard class sailplane optimisation. This refinement cycle, therefore, is mainly focused on iteratively improving the sub-components of Table 1.2 in the identified areas before re-integration into the DSM; after which optimisation module refinement can commence.

Once all of the identified problem areas are corrected in the respective sub-components and the optimisation module, the refined optimisation module is applied to optimise the performance of a SOTA standard class glider i.e. the JS4 sailplane design. The optimised design is then validated in two key areas i.e. the optimised design performance and the optimised design geometry. For the performance validation, the DSM optimised design is simulated with the JSCode to ensure that an actual performance increase is realised. For the geometry validation the optimised design's airfoil profiles and wing geometry are analysed to ensure that all upfront defined constraints are adhered to.

The success of the system refinement cycle is defined as the delivery of an artifact (the refined optimisation module) that is capable of performing SOTA standard class sailplane

cross-country performance optimisation under multiple wing and airfoil profile constraints. This means the refined optimisation module as a whole, as well as its constituent parts, designed, developed, and validated independently, adheres to all the required specifications in such a way that the identified discrepancies of the TSM are addressed and hence global cross-country performance optimisation of a SOTA standard class glider is possible.

1.6 Statement of original contribution

The main contributions of this work can be summarised as follow:

1. The development of a deep learning based, airfoil shape generation framework that enables conditional airfoil shape generation with desired shape and structural characteristics.
2. The development of a deep learning based sailplane cross-country performance optimisation framework that enables efficient and numerically robust optimisation.
3. The application of the developed optimisation framework to optimise the cross-country performance of the current world's best performing standard class sailplane i.e. the JS4.

1.7 Project scope

This section is concerned with the scope and limitations of the proposed study. Specifically, the following items are considered out of the scope of this research:

1. **Simulation domain**: In this research, all airfoil simulations are done with 2D panel codes and hence no wind tunnel experiments or CFD simulations are run for the 2D case. For areas with complex 3D phenomena, such as for the fuselage and

tail area, CFD simulation results are supplied by Jonker Sailplanes[4]

2. **Flight Condition**: Given the operating range of cross-country sailplane flight, this research is focused on incompressible subsonic flight only, with $0 < Mach$ (MA) < 0.3 and $8 \times 10^5 < Reynolds$(RE) $< 4 \times 10^6$

The following items are regarded as out of scope for this research:

1. **Flaps and slats usage**: This study does not include the investigation of flaps and slats.

2. **Supersonic flight**: This study is not concerned with supersonic flight and therefore double wedge airfoils are not considered in the data mining, modelling, or optimisation objectives.

3. **3D surrogate modelling**: The PDLM will be trained on 2D data only and therefore 3D effects will not be accounted for by the deep learning model. The correction for 3D effects will come from the sailplane model.

1.8 Thesis disposition

In this section, a high-level breakdown of the chapters is given.

- Chapter 1: Introduction to the research with a focus on the objectives and methodology.

- Chapter 2: Relevant literature that was reviewed as part of this research is discussed.

- Chapter 3: Development and integration of the required sub-components to form the TSM. Here, validation of the TSMs accuracy is also carried out experimentally and by means of simulations.

[4]Jonker Sailplanes is a leading manufacturer of sailplanes located in South Africa. The head aerodynamicist of Jonker Sailplanes served as a subject matter expert in this research and also supplied simulation data, experimental data, and proprietary in-house developed software which aided in the model development and validation.

- Chapter 4: Identification and derivation of the parsimonious shape and structural variables used to describe an airfoil profile.

- Chapter 5: Discussion of the data mining procedure with special attention given to how parsimonious variables and XFOIL are used to mine the required data for later use in the model development.

- Chapter 6: Use of the mined data to develop and verify the surrogate model that enables mapping between parsimonious airfoil parameters and conventional, as well as non-conventional 2D airfoil performance characteristics.

- Chapter 7: Development of an airfoil shape mapping module in two steps. First, an airfoil generator module is developed with the help of a Bayesian Gaussian Mixture Model (BGMM) capable of generating realistic airfoils in terms of the defined parsimonious variables. Second, an airfoil shape conversion module is developed that is capable of mapping the lower-order parsimonious variable domain to the higher-order raw coordinate domain. It is then shown how these two modules are combined to form the airfoil shape mapping module.

- Chapter 8: Development and accuracy evaluation of the DSM.

- Chapter 9: Development, validation, and discrepancy identification of the optimisation module that enables efficient and intuitively constrained sailplane cross-country performance optimisation.

- Chapter 10: Refinement of the optimisation module given previously identified discrepancies such that global cross-country performance optimisation of the JS4 sailplane design is possible.

- Chapter 11: Conclusion of the study, with a focus on the key findings, main contributions, and finally the recommendations for future research on this topic.

Chapter 2

Background

.

2.1 Introduction

In this chapter, a brief overview of the relevant research constituents is given. First, attention is given to the derivation of the sailplane cross-country model. The model discussed and derived here is responsible for the prediction of the average cross-country speed for a given sailplane design and a defined flight task. This model will later be integrated with the TSM in Section 3.5. This is followed by a discussion of the Lifting Line Theory (LLT) model (Section 2.3), which forms the basis of the 3D drag calculation module as implemented in the TSM in Section 3.4.3.

Next, in Section 2.8 an overview is given of the three main model architectures utilised later in this research. Various airfoil parametrisation methods are then discussed in Section 2.4 from the viewpoint of traditional i.e. constructive or deformative methods, versus deep learning-based methods.

The next section (Section 2.5) is dedicated to an overview of surrogate models and how they are applied in the context of airfoil performance analysis. Traditional methods for

2D airfoil performance analysis are discussed in Section 2.6, alongside deep learning-based surrogate approaches. Finally, in Section 9.3.2, two optimisation methods i.e. Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) as applied to 2D airfoil optimisation problems are discussed. In the first place, this is done with a focus on the theory at a conceptual level before moving to the results of the applications of these models to 2D airfoil optimisation problems.

2.2 Sailplane cross-country modelling

Apart from regulatory and class requirements, the primary objective of sailplane design is to achieve the highest possible average ground speed throughout cross-country flight. As mentioned in Section 1.1, cross-country flight involves a succession of climbs and glides from thermal-to-thermal. The area within a thermal is limited, therefore limited lift can be achieved in a given thermal and the climb within a thermal takes place in a continuous turn. To optimise for a high lift-to-drag-ratio, and ultimately the best possible average cross-country speed V_{avg}, we need to design the sailplane with a minimal sink rate while circling V_{sc}, a minimal sink rate whilst gliding from thermal-to-thermal V_s, and a high interthermal glide speed V_g.

In the following sub-sections the sailplane cross-country model is derived. In this derivation, focus is given to the assumptions made - specifically pertaining to the thermal types, the weather model, and the simplifications made to the cross-country thermal-to-thermal glide profile. For this derivation, it is assumed that the total sailplane lift and sailplane drag for a given design and flight condition is known and hence no attention is given to these elements here. Rather, the discussion of these calculations is dealt with in Section 3.4 respectively.

2.2.1 An idealised cross-country flight model

To derive a simple cross-country model, we make certain assumptions based on previous works (Bramesfeld & Krebs, 2016; Thomas, 1999). First, we assume that the strength within a thermal remains constant throughout the flight. Secondly, we assume that the strength of the thermal does not vary with altitude. Finally, we assume that the transition from glide to climb is immediate, meaning that we do not account for the search for climb. A depiction of the idealised cross-country glide from thermal to thermal, as discussed above, can be seen in Figure 2.1

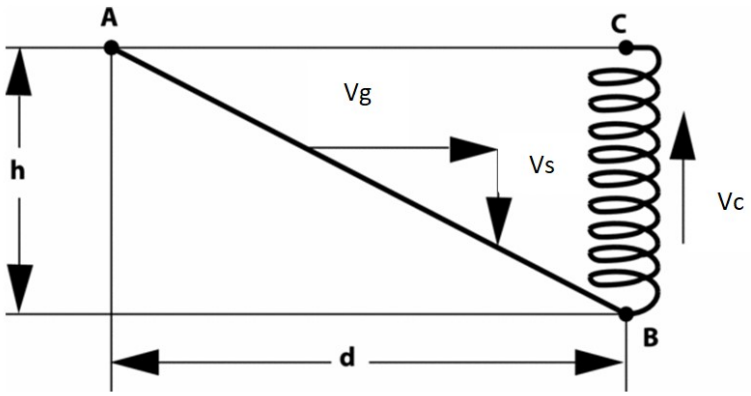


Figure 2.1: Idealised segment of cross-country flight (Bramesfeld & Krebs, 2016)

The observations from Figure 2.1 are as follows: First, there is a glide from point A to point B, where the glider moves from one thermal to another. This glide is characterised by a horizontal flight component V_g and a vertical flight component V_s. V_g is the glide speed component whereas V_s is the sink speed component. Secondly, there is a climb within the thermal from point B to point C where the altitude gained in thermalling is denoted by h. Finally, the distance covered in the glide is represented by d. This idealised depiction forms the basis which is used to derive the cross-country model – as used in this research.

2.2.2 Deriving the average cross-country speed model

From the idealised model in Figure 2.1 we can define the average cross-country speed V_{avg} as:

$$V_{avg} = \frac{d}{t} = \frac{d}{t_c + t_g} \tag{2.1}$$

With t_c and t_g the time spent climbing and the gliding time respectively and defined as:

$$t_c = \frac{h}{V_c} \tag{2.2}$$

and:

$$t_g = \frac{h}{V_s} \tag{2.3}$$

The distance covered in the glide can be defined as:

$$d = \frac{V_g}{V_s} h \tag{2.4}$$

The rate of climb V_c is a function of the thermal strength V_t, which is the rate at which the thermal itself is climbing, and the sink rate of the circling sailplane V_{sc}:

$$V_c = V_t - V_{sc} \tag{2.5}$$

The average cross-country speed can thus be expressed as:

$$V_{avg} = V_g \left(\frac{V_c}{V_c + V_s} \right) \tag{2.6}$$

When a sailplane is in level flight the entire lift component is available to support the sailplane's weight. However when a sailplane is in a turn, flying at airspeed V_k with banking angle φ, only a component of lift is available to support the sailplane's weight. Given an aircraft of weight W, lift coefficient C_L and density ρ the sailplane's airspeed is given by:

$$V_k = \sqrt{\frac{2}{\rho} \frac{W}{S} \frac{1}{C_L \cos(\varphi)}} \tag{2.7}$$

The sink rate while circling can therefore be expressed as:

$$V_{sc} = \left(\frac{C_D}{C_L^{\frac{3}{2}} \cos^{\frac{3}{2}}(\varphi)} \right) . \left(\sqrt{\frac{2}{\rho} \frac{W}{S}} \right) \tag{2.8}$$

Using Newton's first and second laws of motion we define the centrifugal force F_{CF} and the weight F_m as:

$$F_{CF} = m(\frac{V_k^2}{r}) \tag{2.9}$$

and:

$$F_m = mg \tag{2.10}$$

Given these identities we can define:

$$tan\varphi = \frac{F_{CF}}{F_m} = \frac{V_k^2}{rg} \tag{2.11}$$

$$V_k = \sqrt{rg \tan \varphi} \tag{2.12}$$

$$r = \frac{V_K^2}{g \tan \varphi} \tag{2.13}$$

Substituting Equation 2.7 into Equation 2.12 yields:

$$\sin \phi = \frac{2}{\rho} \frac{W}{S} \frac{1}{\mathrm{rgC}_L} \tag{2.14}$$

Applying a trigonometric identity leads to:

$$\cos \phi = \sqrt{1 - \sin^2 \phi} = \sqrt{1 - \left(\frac{2}{\rho} \frac{W}{S} \frac{1}{\mathrm{rg}C_L} \right)^2} \tag{2.15}$$

By substitution of Equation 2.15 into Equation 2.8 we arrive at:

$$V_{sc} = C_D C_L^{-\frac{3}{2}} \sqrt{\frac{2}{\rho} . \frac{W}{S}} \left[1 - (\frac{2}{\rho} \frac{W}{S} \frac{1}{rgC_L}) \right]^{-\frac{3}{4}} \tag{2.16}$$

Breaking down the drag into its main components, i.e. induced drag, profile drag (C_{D0}) and parasite drag (C_{D_p}), yields the following expression for the sink rate while circling:

$$V_{sc} = \left(\frac{kC_L^2}{\pi A.R} + C_{D_0}(C_L) + C_{D_p}(C_L) \right) \left(C_L^{-\frac{3}{2}} \sqrt{\frac{2}{\rho} . \frac{W}{S}} \left[1 - \left(\frac{2}{\rho} \frac{W}{S} \frac{1}{rgC_L} \right) \right]^{-\frac{3}{4}} \right) \tag{2.17}$$

The expression for average cross-country speed V_{avg} can now be written as:

$$V_{\text{avg}} = \frac{\mathrm{V_c C_L^{-1/2}}}{\left(\mathrm{C_{D_0} + C_{D_p}}\right) \mathrm{C_L^{-3/2}} + \frac{\mathrm{k}}{\pi A.R} \mathrm{C_L^{1/2}} + \frac{\mathrm{V_c}}{\sqrt{\frac{2}{\rho} \frac{\mathrm{W}}{\mathrm{S}}}}} \tag{2.18}$$

Substituting Equation 2.5 therefore yields:

$$V_{\text{avg}} = \frac{(\mathrm{V_t} - V_{sc}) \mathrm{C_L^{-1/2}}}{\left(\mathrm{C_{D_0} + C_{D_p}}\right) \mathrm{C_L^{-3/2}} + \frac{\mathrm{k}}{\pi A.R} \mathrm{C_L^{1/2}} + \frac{\mathrm{V_t} - V_{sc}}{\sqrt{\frac{2}{\rho} \frac{\mathrm{W}}{\mathrm{S}}}}} \tag{2.19}$$

The calculation of the two unknown quantities i.e. thermalling speed V_t and the sink rate of the circling sailplane V_{sc} is discussed in the subsequent chapters.

2.2.3 Thermal modelling

To solve and ultimately optimise for the average cross-country speed expressed in Equation 2.19 we need to be able to model the thermalling speed V_t. To do this effectively, an accurate model for a thermal, or collection of thermals is required. Having such a model also allows for the calculation of the sink rate of the circling sailplane by means of Equation 2.16.

Throughout the years there have been various researchers who have developed models to mathematically express thermals. One such model is that of Carmichael et al. (Carmichael, 1954), where thermals are classified as narrow-strong, narrow-weak and wide. This was further developed by Horstmann et al. (Horstmann, 1976) who performed a systematic analysis of numerous in-flight experiments. This resulted in an improved set of standard thermal models that are more realistic in representing European weather conditions, and which have seen wide adoption.

This model of Horstmann et al. classifies thermal into four distinct categories varying in thermal strength and width. These four distinct categories representing four standard thermal profiles are: the combination of strong (2), weak (1), wide (B), and narrow (A) thermals; see Figure 2.2 (Kubrynski, 2006). Using this notation, a strong-narrow thermal would therefore be denoted as an A2 thermal type.

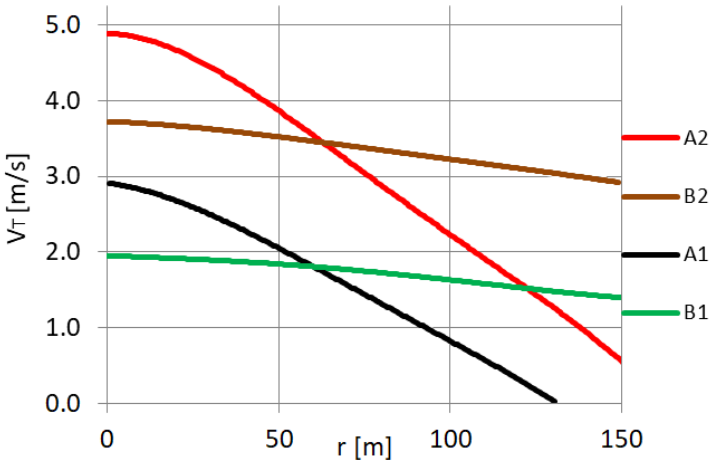


Figure 2.2: Standard thermal profiles as defined by (Horstmann, 1976) specified in terms of thermal strength and gradient

The specification of thermal characteristics is the most challenging problem as this depends on a great number of factors including, but not limited to, weather conditions, geographical region, ground features, altitude and time (Kubrynski, 2006). Horstmann's model assumes a linear variation of the vertical velocity distribution outside a 60-meter radius thermal core. The thermal strength V_t and gradients associated with the standard Horstmann profiles at a radius of 60 m is defined as follow:

Table 2.1: Standard thermal profiles as defined by (Horstmann, 1976) specified in terms of thermal strength and radius from the core of the thermal

	A1	A2	B1	B2
V_t (m/s)	1.750	3.500	1.750	3.500
Gradient $(1/s)$	0.025	0.032	0.0045	0.006

2.2.4 Calculating the maximum climb speed

After defining the thermal model, the next challenge in calculating the average cross-country speed is to determine the maximum climbing speed while circling within a given thermal. This quantity is required as it is ultimately used to calculate the optimum speed

at which a sailplane should be flown between thermals.

During cross-country flights, pilots circle within a thermal in order to gain altitude and extract maximum energy from the thermal. This allows them to glide to the next thermal. The circling is done at a specific angle of attack and banking angle. To determine the relationship between the average speed during cross-country flights and the sink speed while circling, it is important to consider the circling angle and banking angle. Equation 2.17 can be used to calculate the sink speed while circling under different operating conditions, such as a range of lift coefficients and circling radii. By using this equation for different radii and operating conditions, a matrix can be constructed that relates the flight speed, thermal radius, total aircraft drag, and the thermalling angles (banking angles) for each wing lift and turning radius case. Using this matrix, along with specific thermal strength information (as seen in Table 2.1), the climb speed while circling for a particular turning radius and operating condition can be calculated. From this information, the lift and turning radius location can be determined where the maximum climb speed is achieved for each specified thermal type.

2.2.5 Weather model definition

The final element required to solve Equation 2.19 is a weather model. In this research, a weather model refers to a collection of thermal models in which the distance that the sailplane travels in a particular thermal type, is specified. For instance, if we have a 300 km cross-country task the Quast weather model, as seen in Table 2.2, states the glide-time allocation per thermal type (Thomas, 1999). As seen in the following section, the lift-to-drag ratio, time spent climbing, and time spent gliding is calculated for each thermal definition in the weather model. These quantities are then used to arrive at the average cross-country speed for the given weather model.

Table 2.2: Quast's weather model for a 300 km cross-country flight task

Thermal type	Percentage allocation (%)	Total flight time (km)
A1	8	24
A2	42	126
B1	8	24
B2	42	126

In this research, the Quast weather model as well as custom-defined weather models that make use of either Horstmann or custom thermal types are used. The reason for using both standard weather and custom weather models is to ensure that adequate diversity is present in both the development and validation of the TSM and the DSM.

2.2.6 Glider lift-to-drag ratio and average cross-country speed calculation

With a weather model defined, the sailplane performance characteristics quantified, and the flight conditions profiled, the final step is to calculate the lift-to-drag ratio for the maximum climb speed for each thermal type present in the weather model. Using the fact that the interthermal glide should be flown to maximise the average cross-country speed (see Equation 2.20), the optimal lift coefficient $C^2_{L_{Opt}}$ can be found by iteratively solving Equation 2.21:

$$-\frac{dV_{avg}}{dC_L} = 0 \tag{2.20}$$

$$C_{D_p} + C_{D_o} - \frac{kC^2_{L_{Opt}}}{\pi AR} - \frac{V_c C^2_{L_{Opt}}}{2\sqrt{\frac{2W}{\rho S}}} = 0 \tag{2.21}$$

With $C^2_{L_{Opt}}$ and the corresponding total sailplane drag known, for the specific operating condition, the lift-to-drag ratio can be calculated and Equations 2.1 to 2.3 can then be used to calculate the average cross-country speed.

2.3 Lifting Line Theory

An important component of the sailplane performance model is to determine the 3D lift based on the 2D sectional properties of each airfoil along the wingspan. This requires the use of the LLT model.

LLT is based on the assumption that, although the flow around an aircraft's wing is 3D, it can be approximated as a linear summation of the 2D flow around the elemental airfoils (Karali et al., 2019; Prandtl, 1921; Prandtl, 1918). In LLT, the assumption is that the wing is a flat plate lying in the x-y plane – the theory therefore does not take thickness distribution into account. The classical implementation of the LLT is also unable to handle dihedral or sweepback angles but is however able to model tapered wings with geometrical and aerodynamic twists.

The LLT aims to approximate the lift per unit span length distribution, the wing lift coefficient, and the wing induced drag coefficient. The Kutta-Joukowski Lift Theorem (Joukowski, 1906; Kutta, 1902) states that lift is directly proportional to vortex strength. The LLT therefore must be capable of predicting the span-wise bound vortex strength per unit length distribution. If we approximate the vortex strength distribution, $\Gamma(y)$, by a Fourier series, then the vortex strength distribution can be expressed as:

$$\Gamma(\theta) = 2bV_\infty \sum_{n=1}^{N} A_n sin(n\theta) \tag{2.22}$$

with the positional coordinate, y, of the wing defined as:

$$y = -\frac{b}{2} cos(\theta) \tag{2.23}$$

or:

$$\theta = cos^{-1}\left(\frac{-2y}{b}\right) \tag{2.24}$$

where $\theta \, \epsilon \, (0, 2\pi)$ and b the total wing span.

The problem, therefore, reduces to the calculation of the unknown Fourier series coefficients of 2.22 - amplitudes A_n. The Fourier series approximation becomes more accurate

as the terms of N increase. To make use of Prandtl's governing equations the derivative of $\Gamma(\theta)$ is required and is defined as:

$$\frac{d\Gamma}{dy} = \frac{d\Gamma}{d\theta}\frac{d\theta}{dy} = 2bV_\infty \sum_1^N nA_n \cos(n\theta)\frac{d\theta}{dy} \tag{2.25}$$

Substituting Γ and $\frac{d\Gamma}{dy}$ into the governing equations yields:

$$\begin{aligned}\alpha_{geo}(\theta_0) = \frac{2b}{\pi c(y_0)}\sum_1^N A_n \sin n\theta + \alpha_{L=0}(\theta_0) \\ +\frac{1}{\pi}\int_0^\pi \frac{\sum_1^N nA_n \cos n\theta}{(\cos\theta - \cos\theta_0)}d\theta\end{aligned} \tag{2.26}$$

where α_{geo} is the angle of attack, $c(y_0)$ is the chord defined at the arbitrary segment y_0, similarly θ_0 is the θ at the arbitrary defined segment y_0, and $\alpha_{L=0}$ is the zero-lift angle of attack. It is shown that Equation 2.26 has an elegant closed-form solution (Boyd, 1967) expressed as:

$$\begin{aligned}\tfrac{1}{\pi}\textstyle\int_0^\pi \frac{\sum_1^N nA_n \cos n\theta}{(\cos\theta-\cos\theta_0)}d\theta = \sum_1^N \frac{nA_n}{\pi}\int_0^\pi \frac{\cos n\theta}{(\cos\theta-\cos\theta_0)}d\theta \\ = \sum_1^N \frac{nA_n}{\pi}\left(\frac{\pi\sin n\theta_0}{\sin\theta_0}\right) = \sum_1^N nA_n\frac{\sin n\theta_0}{\sin\theta_0}\end{aligned} \tag{2.27}$$

The closed-form solutions simplifies Equation 2.26 to:

$$\alpha_{geo}(\theta_0) = \frac{2b}{\pi c(y_0)}\sum_1^N A_n \sin n\theta + \alpha_{L=0}(\theta_0) + \sum_1^N nA_n\frac{\sin n\theta_0}{\sin\theta_0} \tag{2.28}$$

Presently, the equation appears in a more simplified manner; however, certain variables remain unknown. To determine these variables and consequently deduce the circulation distribution, a series of linear equations is devised utilising points scattered along the bound vortex. By solving these equations simultaneously, the values of A_n's are obtained, leading to the derivation of the circulation distribution. Equating the definition of the lift coefficient to the Kutta-Joukowski Lift Theorem yields:

$$L = \frac{1}{2}\rho V_\infty c(y_0)C_l = \rho V_\infty \Gamma \tag{2.29}$$

Rearranging for C_l, and substitution of Equation 2.28 into Equation 2.29 whilst integrating over the wing yields:

$$C_L = \frac{2b^2}{S}\sum_1^N A_n \int_0^\pi \sin n\theta \sin\theta d\theta = A_1\pi AR \tag{2.30}$$

Equation 2.30 therefore indicates that the lift coefficient is solely dependent on the first coefficient of the Fourier series.

The downwash flow leads to a rotational adjustment of the lift vectors for each section, causing them to be inclined backward relative to the free-stream direction. Consequently, a lift-induced drag emerges due to this effect. It's important to note that the governing equations are based on potential flow, implying the absence of friction or pressure drag. Nevertheless, the lift-induced drag becomes a significant constituent of the overall wing drag. To determine the downwash velocity induced at any span location, one needs to ascertain the strength of the wing loading. Once this is known, the variation in local flow angles can be computed. By integrating the component of the section lift coefficient parallel to the free-stream direction across the span, the induced drag coefficient can be evaluated.

$$D_i = \rho V_\infty \int_{-b/2}^{+b/2} \Gamma \sin(\alpha_i)\, dy \tag{2.31}$$

If small angles of attack are assumed then the induced drag $D_i = L_i sin(\alpha_i)$ is approximated by $D_i = L_i sin(\alpha_i)$. Which, as shown in (Boyd, 1967), simplifies to:

$$C_{Di} = \pi \cdot AR \cdot \sum_{n=1}^{\infty} n A_n^2 \tag{2.32}$$

2.4 Airfoil parameterisation

In this section a comparison is drawn between traditional and deeplearning-based airfoil parametrisation methods. For the traditional methods, specific attention is given to the five most popular airfoil parametrisation methods, i.e. the B-Spline, Hicks-Henne, Singular Value Decomposition (SVD), PARSEC, and Class Shape Transformations (CST). Comments are given on the usability and effectiveness of these aforementioned traditional methods with focus on their Padulo-completeness. An overview of the current state-of-the-art in DL-based parametrisation methods is also given and their effectiveness in both parametrisation and flexibility in inverse design is highlighted. Finally, the drawbacks of both the traditional and DL-based methods are discussed and specific mention is made of

how these discrepancies can be overcome by developing a novel DL-based parametrisation method.

2.4.1 Traditional airfoil parametrisation techniques

Airfoil parametrisation can be defined as the process of representing an airfoil surface with parametric equation coefficients and/or control points. That is, mapping the airfoil surface from the high dimensional raw coordinate space to low dimensional parametric space. The two categories of airfoil parametrisation methods are constructive and deformative methods (Masters et al., 2017; Xiaoqiang et al., 2018):

1. Deformative methods: methods take an existing airfoil then deform it to create the new shape.
2. Constructive methods: represent an airfoil shape based purely on a series of specified parameters.

Effective airfoil parametrisation methods are characterised by their efficiency to describe a large design space with a restricted set of parsimonious variables (Masters et al., 2017). This means that an airfoil parametrisation method should be able to capture complex critical airfoil features inherent in the airfoil shape with a small set of interpretable equation coefficients. Whether deformative or constructive, Padulo et al. (Padulo et al., 2009) define the desirable properties of an airfoil parametrisation method as:

1. *Completeness*: Assesses whether the method can describe any airfoil, up to a specific degree of accuracy.
2. *Orthogonality*: Guarantees that each airfoil shape generated with the method corresponds to a unique set of input parameters.
3. *Flawlessness*: Ensures that the parametrisation method does not generate impractical airfoil shapes, such as intersecting or degenerated airfoils.

4. *Parsimony*: Limits the number of parameters required to induce significant changes on the main geometric features of the airfoil profile.

5. *Intuitiveness*: Relates to the physical meaning of the parameters, which simplifies the choice of input bounds or design judgments.

The five most popular airfoil parametrisation methods are the B-Spline, Hicks-Henne, SVD, PARSEC and CST. The function proposed by Hicks and Henne (Hicks & Henne, 1978) applies a bump function to a baseline airfoil shape to modify the baseline airfoil shape. The Singular Value Decomposition or SVD method uses orthogonal decomposition to derive a set of ordered orthogonal basis modes from a set of predetermined training airfoils (Masters et al., 2017). The B-spline method describes an airfoil shape with a piece-wise polynomial function expressed as a linear combination of the product of basis functions and control points (Weihong et al., 2017). The CST method comprises a curve function by a class function, a shape function and a trailing edge term (Kulfan, 2007). The PARSEC method was originally proposed by Sobieczky in 1999 (Sobieczky, 1999). The method makes use of 11 descriptive parameters which have physical meaning with respect to the airfoil geometry. Another constructive parametrisation method that is worth a mention is the Simple Parametric Method (SPM) (Ziemkiewicz, 2017). This method is simple yet efficient in its approach to airfoil profile generation where only six parametric variables are required. These variables were derived in such a manner that local and global [1] control results and to ensure that intuitiveness is inherent in the generation process.

Although widely used, each of the above-mentioned methods fails to comply with all five categories of Padulo et al. Specifically, Hicks-Henne functions, B-splines, CST, and SVD lack parsimony, intuitiveness, and flawlessness. Although PARSEC-11 has intuitiveness and parsimony implicit in the parameter definitions, it lacks the capability to represent any airfoil (completeness) and is also prone to generating impractical or non-airfoil shapes (flawlessness). To address the completeness issue, Nemati and Jahangirian extended the PARSEC-11 method by adding the capability of explicit leading and trailing edge control

[1] Global control refers to the overall airfoil profile shape whereas local control refers to the control of various airfoil characteristics such as Trailing Edge (TE) thickness, nose radius, etc.

(Nemati & Jahangirian, 2020). This means that the leading edge can be described with six parameters and the trailing edge with 11 parameters. Although their solution means more flexibility in shape representation, it does not account for additional intuitiveness or flawlessness.

Masters et al. (Masters et al., 2017) analysed the efficiency of the aforementioned airfoil parametrisation methods with the inclusion of the radial basis function parametrisation method by evaluating the shape recovery of 2 000 airfoils. In this, the respective methods' efficiency of design space coverage within geometric tolerance was evaluated. The study highlighted that between 20 to 25 design variables are needed for all methods to cover the full design space within geometric tolerance; the SVD method was shown to have the highest precision. The study also shows that to achieve aerodynamic convergence within one drag and one lift count, between 33 to 66 design variables are required, thus highlighting the strong correlation between geometric error and aerodynamic convergence.

2.4.2 Deep learning-based airfoil parametrisation techniques

With the advent of big data and deep learning, more sophisticated parametrisation methods were developed in an attempt to create a Padulo-complete airfoil generation method, that is, an airfoil generation method that strictly adheres to all five criteria set out by Padulo et al. (J. Li et al., 2022; Padulo et al., 2009). The majority of these methods employ either Generative Adversarial Networks (GANs), Convolutional Neural Networks (CNNs), or ANNs in an attempt to arrive at novel parametrisation frameworks. These novel parametrisation frameworks are then typically used in an airfoil surrogate model which in turn is embedded in an optimisation routine.

To date, the DL-based parametrisation methods that have seen the most success are those that made use of the GAN architecture. One such recent example is the framework developed by WU et al., 2022. They showed that Hicks-Henne parametrisation coupled with a GAN can be used to accurately predict the performance of a 2D airfoil in a low-resource environment. Here they used 14 control points (7 points for the upper surface

and 4 points for the lower surface) to parameterise the airfoil. The 14 dimensional vector was then coupled with a Mach number to form a 15 dimensional vector to be used in the training of the GAN. The authors showed that the developed model exhibits accurate flow-field prediction in a certain range of Mach numbers. Chen et al. (W. Chen & Fuge, 2018) proposed a new method for parameterising aerodynamic designs using a deep generative model called Bézier-GAN. This method allows for a more compact representation of the design space, which can accelerate the optimisation process by reducing the number of costly computational fluid dynamics simulations required. The Bézier-GAN is specifically designed for smooth geometries and allows for a two-level shape parametrisation that separately controls major and minor shape deformation. The article compares the Bézier-GAN to several other parametrisation methods and finds that it outperforms them in terms of both representation compactness and optimisation convergence. The work of (X. Du et al., 2021) proposes a fast and interactive airfoil aerodynamic design framework that can complete an airfoil optimisation within a few seconds. The framework uses a B-spline-based generative adversarial network model for shape parametrisation, a Gaussian copula dependent sampling to reduce the input space, and a combination of multilayer perceptron, recurrent neural networks, and mixture of experts for surrogate modelling to enable scalar and vector response predictions. The results show that the optimal designs and aerodynamic quantities obtained by the proposed framework agree well with the ones computed by direct CFD-based optimisations and evaluations.

However successful, GANs have not only been applied to direct airfoil parametrisation but also to the inverse design problem. Traditionally, in airfoil inverse design, a target pressure distribution is supplied and the corresponding airfoil profile is predicted – usually in the form of x and y coordinates. Hence, traditionally with airfoil inverse design, rather than starting with airfoil parameters and acquiring the corresponding airfoil performance, the performance in terms of pressure distribution is specified and the airfoil shape solved for. Deep learning, however, extends this traditional inverse design approach where practically any performance coefficient can be specified as input and the corresponding profile can be solved for. One such example is the work of Wang et al. (J. Wang et al., 2022) showed that generative models such as Wasserstein Generative Adversarial Networks (WGANs)

in combination with Conditional Variational Auto Encoders (CVAEs) can be used to generate target wall Mach distributions for the inverse design that matches specified features, such as the location of suction peaks, shocks and aft loading. Similarly, the work of Achour et al. (Achour et al., 2020) saw the development of a Conditional Generative Adversarial Network (CGAN) capable of predicting an airfoil profile given the maximum lift-to-drag ratio and total airfoil area requirements.

Another architecture that has seen success in the field of 2D airfoil parametrisation is the CNN (Catalani et al., 2023). Recently, Qiuwan et al. (Q. Du et al., 2022) proposed a CNN-based framework named DPCNN for airfoil design and performance prediction. The framework achieves airfoil profile parametrisation, physical field prediction, and performance prediction with only three geometric parameters, providing advantages such as good robustness, fast computation speed, and high prediction accuracy compared to conventional machine learning methods. The results show that the DPCNN framework can generate substantial airfoil profiles, and the predicted results can be obtained within 5ms with low prediction errors for physical field and aerodynamic performance parameters. The framework is shown to be effective for optimisation attempts of operating parameters, with good stability and convergence. Overall, the DPCNN framework is shown to provide outstanding advantages in time cost and prediction accuracy for airfoil design and optimisation. Another study that utilised CNNs to enable flow-field prediction was the work of Chen et al (H. Chen et al., 2020). They developed a new approach for predicting multiple aerodynamic coefficients of airfoils using a CNN. Rather than using airfoil parametrisation techniques, a CNN was employed to extract the important geometrical characteristics from airfoil images. More specifically, their method involves creating a composite airfoil image that combines an original airfoil image with a transformed image created by convolving the image with flow conditions. The CNN model predicts pitch-moment, drag, and lift coefficients simultaneously with high accuracy and minimal time consumption. Although there were a few instances with significant deviations from the optimum, most of the predicted values were close to the actual values.

CNNs have also been shown to be effective in the inverse design problem (Lei et al., 2021). One such notable study is that by Sekar et al. (Sekar et al., 2019). Sekar et al. showed

how this deep learning architecture can be used effectively to obtain the airfoil shape from the coefficient of pressure distribution. The CNN is applied due to its ability to handle any airfoil geometry without the need for complex parametrisation. In the testing phase, a new pressure coefficient distribution is given to the CNN model, generating an airfoil shape that is very close to the associated airfoil with an $L2$ error of less than 2% for most of the cases. The results show that the CNN method is very efficient in terms of computational time and demonstrate a competitive prediction accuracy. Similar efforts by Kim et al. (Kim & Yoon, 2022) showed that CNNs are effective to solve for airfoil profiles from a given C_p distribution. Here, the surface pressure coefficient distributions are used as input to the CNN, which is used to predict the corresponding airfoil profile. Numerical simulations of viscous and steady 2D flow over the airfoil are used to obtain the surface pressure coefficient distributions as the dataset. Three modification approaches are performed to achieve more favourable distributions, resulting in predicted airfoils with higher lift-to-drag ratios and more preferable pressure fields than the corresponding baseline airfoils. The study concludes that the approach based on deep CNN shows promise for developing geometric disturbances as a flow control method.

2.4.3 Traditional vs deep learning parametrisation approaches

When commenting on the limitations of the traditional airfoil parametrisation methods discussed, other than the failure to adhere to some of Padulo's criteria, is the lack of explicit structural control – such as control over the airfoil area, bending inertia about the x and y axes, x and y centroids, and the airfoil perimeter. All the previously listed traditional airfoil parametrisation techniques make use of mathematical techniques to control only the shape of the generated airfoil, and are not able to model airfoil profiles as a function of both intuitive shape and structural parameters. Optimising for desirable airfoil properties, in terms of structural limitations, are equally as important as optimising for airfoil shape properties, as it has a direct consequence on manufacturability, wing stiffness, fuel storage, etc. Currently, structural characteristics are either implicitly accounted for in the optimisation process by constraining the bounds of the parametric coefficients similar

to that of a proven seed or baseline airfoil, or accounted for in an additional calculation step after the candidate airfoil was generated (De Gaspari & Ricci, 2010; Yuan et al., 2019). Although both are appropriate methods, taking either one of these two approaches limits the search space of possible solutions and therefore restrains the novelty of the newly produced airfoil(s).

When commenting on the limitations of the current literature on deep learning applied to airfoil generation by means of parsimonious variable specification, it is clear that current solutions either generate airfoils by: 1) mapping airfoil profiles to uninterpretable latent dimensions (W. Chen & Fuge, 2018), or 2) inversely solving for isolated performance, shape, or structural characteristics (Achour et al., 2020; G. Sun et al., 2015; J. Wang et al., 2022). The existing state-of-the-art also mainly makes use of GANs, CNNs, or ANNs and the application of architectures such as transformers and Long Short Term Memorys (LSTMs) have yet to be fully explored.

This therefore motivates the development of a deep learning-based airfoil parametrisation method capable of generating airfoils in terms of both intuitive shape and structural characteristics that is Padulo-complete.

2.5 Surrogate modelling

Surrogate models (SM) are computationally inexpensive mathematical approximations that mimic the deterministic computationally expensive behaviour of an original system over the complete or part of the design space (Fang et al., 2006). Once assembled, a SM is used as a replacement for the expensive full-order model. SMs can be used to predict the values of the objective functions at locations that do not belong to the set of points used during the fitting process or training phase (Yondo et al., 2018). To date, there are numerous surrogate modelling techniques employed in aircraft design. Amongst these, data-fit models (DFMs), reduced-order models (ROMs), multi-fidelity models (MFMs) and hybrid models (HMs) are widely used. Moreover, the most popular surrogate models include Radial Basis Functions Neural Networks (RBF-NN), Multilayer

Perceptrons (MLPs), Response Surface Models (RSMs), and Kriging Models (Z. Li & Zheng, 2017).

The most computationally inexpensive SM is the polynomial regression RSM. These models approximate the modelled function, globally or locally, with polynomials of any order (Yondo et al., 2018). Advantages of RSMs are their ease of interpretability, implementation and visualisation. RSMs are also useful when there is a need to empirically develop a model or in modelling low fidelities in MFM frameworks. Although RSMs carry several advantages, they have seen limited adoption due to their poor prediction power in highly nonlinear and multi-modal problems. Furthermore, high-order polynomials require large training sets and are prone to instabilities (Barton, 1992).

A Kriging model is a weighted linear combination of the trained samples of the modelled functions. This method makes use of the spatial correlation among the data samples (Barton, 1992). Closer samples are likely to have highly correlated response values (i.e. bigger weights), whereas samples taken far apart are unlikely to have highly correlated response values (Forrester et al., 2008). Kriging models hold the advantage of alleviating the effects of data clustering by accounting for directional influences. This means that these models are suited for noisy experiments. One of the drawbacks of Kriging models is that the generated surface does not pass through any of the point values which causes interpolated values to be higher or lower than real values. In theory, Kriging models could be applied to high-dimensional problems (50 design variables or parameters). However, for problems of this nature, building a Kriging surrogate model becomes tedious and time-consuming.

The Multi-Layer Perceptron (MLP) and RBF-NN are two common neural architectures used for surrogate modelling purposes in aerodynamic problems. For MLP-based SMs, multiple feed-forward densely connected networks coupled with back propagation are used to update training weights and ultimately minimise the training error. Although these networks can approximate any continuous functions in theory, the convergence of the learning process does not guarantee that the network correctly predicts the actual function (Skinner & Zare-Behtash, 2018). MLPs have attractive characteristics for use as SMs.

These characteristics include:

- implicit dimensionality reduction,
- implicit sensitivity analysis (ability to analyse effect with varying input parameters), and
- the ability to model complex feature interactions to a high degree of accuracy.

Disadvantages of using these methods as SMs include:

- high computational burden,
- tedious and iterative optimisation to avoid overfitting, and
- the need for external tools to interpret them.

Surrogate models and approximation-based optimisation methods have attracted much attention and received recognition for their ability to condense and expedite the design and optimisation process (Skinner & Zare-Behtash, 2018). These models have also been criticised for their lack of generalisation capability (out-of-sample performance), a problem arising from model and training data discrepancies and model limitations. Multiple surrogate models, however, constructed by weighting individual surrogate models, in general, gives lower errors with better robustness than single surrogate models. Models that have been combined in this manner include the polynomial RSM, the Kriging model, RBF-NN and MLPs (Goel et al., 2007; Jin et al., 2001).

2.6 2D airfoil performance analysis

As mentioned earlier, 2D airfoil performance analysis is the calculation of the 2D performance characteristics of a given airfoil profile. Accurate 2D airfoil performance predictions are required seeing that wind tunnel experiments are not always feasible due to

either budget constraints, time constraints or both. It is therefore desirable to have a method that is able to efficiently and accurately calculate the 2D airfoil performance of any given (practical) airfoil profile. This can be done in one of two ways: simulation methods and surrogate models. With simulation methods, mathematical models, governed by the laws of fluid dynamics, are numerically solved to arrive at the appropriate performance characteristics. With surrogate models (see Section 2.5), an approximation model is developed that can be used to predict the flow field characteristics for a given airfoil configuration.

In this section both simulation methods and surrogate models are discussed from the viewpoint of traditional versus deep learning-based approaches to 2D airfoil performance analysis. Seeing that simulation methods have been around longer and are still the industry standard when it comes to aerodynamic design and optimisation, these methods are discussed under the traditional approaches. With deep learning becoming an ever more present figure in the field of aerodynamic design and optimisation, especially in the preliminary design phases, the surrogate models are discussed under the deep learning-based approaches.

2.6.1 Traditional 2D performance analysis techniques

When it comes to 2D airfoil performance analysis by means of simulation, there are two main techniques to be employed: CFD methods, and Vortex panel methods.

CFD methods are those most commonly used for performance evaluation in airfoil design and optimisation (Weia et al., 2020). They use numerical algorithms and models to solve the equations that describe the flow of fluids, such as the Navier-Stokes equations, which govern the behaviour of fluids in motion. CFD methods allow engineers and researchers to analyse and predict the behaviour of fluids in complex geometries and flow regimes, which would be difficult or impossible to observe and measure experimentally. These methods are not limited in the geometries they can simulate and performance for practically any solid body can be predicted.

Alternatively, panel code methods are a class of numerical methods used to analyse fluid flows around solid bodies, such as airfoils, wings, and fuselage/tail configurations. These methods are based on the principle of dividing the body into a number of flat or curved panels, each of which is assumed to produce a constant pressure and velocity distribution. The panel code method solves for the potential flow around the panels by considering the boundary conditions of the fluid flow, such as no-flow or free-flow conditions, and uses integral equations to calculate the fluid velocity and pressure at each point around the body. Panel code methods are particularly useful for analysing low-speed flows around two-dimensional airfoils, where the assumption of potential flow is valid, and the flow can be described by the two-dimensional Laplace equation. The method is computationally efficient and relatively easy to implement, but it has limitations, such as its inability to account for turbulence and other complex fluid phenomena.

Both panel code methods and CFD simulations have been used extensively in the design and optimisation of airfoils, wings, and other aerodynamic shapes for a wide range of applications, including aircraft, wind turbines, and automobiles.

In the present work the vortex panel code, XFOIL (Drela, 1989) is used as an airfoil performance analysis module. XFOIL analyses the flow around the shape of a two-dimensional airfoil via a potential flow panel method coupled with an integral boundary layer formulation. XFOIL can also perform corrections for viscosity and compressibility and was designed for low Reynolds number applications. In XFOIL the solution is iteratively achieved by numerically solving for the inner and outer flow solutions on the boundary layer displacement thickness. Thus, the XFOIL code calculates the viscous pressure distribution and captures the influence of limited trailing edge separation and laminar separation bubbles (Morgado et al., 2016).

Compared to CFD codes that implement Reynolds-averaged Navier–Stokes equations (RANS) and Finite Volume Method (FVM) methods with state-of-the-art transition models, it has been shown that XFOIL produces good accuracy in the calculation of low Reynolds number airfoils, and can be adopted in the rapid design process (J et al., 2016). Molland (AF et al., 2004) showed that the XFOIL-calculated lift-to-drag ratio and cavita-

tion performance results for four different kinds of NACA profiles were in good agreement with experimental results. Wei et al. (Weia et al., 2020) showed that, for low Reynolds and Mach numbers, XFOIL is accurate in predicting airfoil performance parameters when evaluated against experimental results. The high accuracy, efficiency, and robustness of XFOIL in 2D airfoil performance analysis, especially in the linear subsonic regime, have also been confirmed by numerous researchers (Pires et al., 2016; M. Selig, 2003; Z. Sun et al., 2021; Timmer & Bak, 2013; Van Treuren, 2015; W. J. Zhu et al., 2014)

2.6.2 Deep learning-based 2D airfoil performance analysis techniques

The application of deep learning to the field of aeronautics has seen widespread application, with the focus on using trained deep learning models as low-fidelity surrogate models. DL-based surrogate model architectures that have seen success over the years include CNNs, Radial Basis Functions (RBFs)-based neural networks, ANNs, and GANs (Andres et al., 2012; Catalani et al., 2023; Hui et al., 2020; Kwon & Choi, 2019; Mao et al., 2017; Z. Wang et al., 2023).

One such recent study that makes use of both Kriging and ANN-based surrogate models to calculate the performance prediction of a twin vertical axis wind turbine (VAWT) is the work of Chen et al. (Y. Chen et al., 2023). The study aimed to reduce the time consumption required for high-fidelity unsteady simulations using CFD during the design stages of the twin-VAWT. The turbines' pitch angles and their averaged torques at the best tip speed ratio were taken as the input and output, respectively. The study found that both Kriging and ANN models can provide satisfactory predictions using only 22.45% of CFD observations as a training set. Additionally, the proposed data sampling strategy was tested to be modest and robust through sensitivity analysis. The study was, however, limited to the operating conditions and geometries simulated seeing that both were fixed in the data mining phase. Similarly, Zhu et al. (G. Zhu et al., 2023) showed that a deep belief network is effective in establishing the response relationship between hydrofoil

geometry and performance. Specifically, it was shown how a deep belief network can accurately predict the performance of a given hydrofoil, and parametrised with a 7-th order Bezier curve. Similar to (Y. Chen et al., 2023), the generalisation of the work of Zhu et al. is limited in that data was mined for a fixed Reynolds number and the airfoil geometry was fixed as the NACA63-815 hydrofoil. Another use of deep belief networks as a surrogate model for 2D airfoil performance prediction came from the work of Tao et al. (Tao, Sun, Guo, et al., 2020) combined a deep belief network with Principal Component Analysis (PCA) to train a robust and efficient surrogate model capable of optimising the shape of transonic laminar flow airfoils for a given operating condition. In this study, the CST method was used to parametrise the airfoil shape which resulted in 14 design variables. Using PCA, these 14 variables were reduced to 8. Although the network was extremely accurate in predicting performance characteristics of a 2D airfoil, the training data was limited to a fixed Mach number, Reynolds number and AoA. This means the trained model could only be used to optimise the airfoil shape for these operating conditions.

A common theme across the aforementioned works is that the surrogate models' applicability to multi-objective optimisation over multiple flight conditions is limited to only a narrow set of operating conditions and airfoil profiles. This is a result of the restrictions set in the data mining phase on the operating conditions i.e. Reynolds number, Mach number, and angle of attack, as well as the variation in airfoil geometry. A partial attempt to address this issue was made by Karali et al (Karali et al., 2020). Karali et al. focused on utilising neural networks to construct a one-to-one mapping of airfoil shape parameters to aerodynamic properties. To achieve this, the authors mined training data from an in-house developed non-linear lifting line code (Karali et al., 2019). The database included thousands of wing-tail configuration geometry parameters and their corresponding performance coefficients. It was shown that the developed deep learning model accurately calculates the aerodynamic performance of multiple wing-tail configurations including 3D maximum lift coefficient and pre- and post-stall aerodynamic behaviour from two-dimensional airfoil performance data and geometric information. Although the issue of airfoil geometry variation was partially addressed here in the data mining phase,

the Mach number and Reynolds number were fixed. This limited the applicability of the trained model to general optimisation over the entire flight envelope. The airfoil geometry problem was partially addressed because Karali et al. opted to use the NACA 4 digit series as airfoil parametrisation method. This meant the descriptive features used to parametrise the airfoil were limited to airfoil thickness ratio, airfoil camber ratio and airfoil maximum camber location. Also, the use of the NACA 4 digit series limited the profile diversity of airfoils in the training set as this method can only generate simple airfoil profiles. This limited application of surrogate models to general airfoil performance optimisation was however addressed by Zuo et al (Zuo et al., 2022). They proposed a combination of a CNN and a multi-head ANN for fast sparse flow field prediction around airfoils. First, a CNN is used to extract the geometry parameters of an airfoil from the corresponding input gray scale image. Secondly, the extracted geometric parameters together with Reynolds number, angle of attack and flow field coordinates are used as the input of the multi-layer perceptron and the multi-head perceptron. This second model is used to predict the incompressible laminar steady sparse flow field around the airfoil. To overcome the recurring issue of airfoil diversity, Zuo et al. used the entire University of Illinois Urbana-Champaign (UIUC) database to fit the CNN-based parametrisation model. As for the operating condition issue, data is mined via CFD simulations for a wide range of Reynolds numbers, Mach numbers, and angles of attack.

When training surrogate models the generalisation capacity of the model needs to be maximised whilst the model prediction variance need to be minimised. This is often difficult to achieve in low resource environments. A common method used to reduce the variance and improve the generalisation capacity of a surrogate model is the application of heterogeneous or homogeneous ensembling. In heterogeneous ensembling the output of models with different architectures is combined in a linear or non-linear way as opposed to the linear or non-linear combination of models with the same architectures in homogeneous ensembling. A notable example of homogeneous ensembling is the works of Lee et al (Lee et al., 2023). Lee et al. introduced a deep learning ensemble framework using a deep residual neural network and K-fold cross-validation for accurate and fast prediction of aerodynamic coefficients of a cone with shape changes due to ablation. The model was

trained to learn the underlying relationship between shape transformations and changes in aerodynamic coefficients. The ensemble model showed improved performance for aerodynamic coefficients compared to the MLP and Gaussian process regression models.

2.7 Model search

In this section a brief overview is given of the theory behind the GA and the PSO techniques. From here examples are given on the success these methods have respectively received in the field of airfoil shape optimisation under multiple constraints.

2.7.1 Optimisation algorithms used in aerodynamic optimisation

In aerodynamic optimisation problems, two families of search models receive preference; these are GAs and PSO.

The GA (Bozorg-Haddad et al., 2017) is a heuristic search technique that is inspired by the process of natural selection and genetic evolution. In a GA, a population of potential solutions to the problem is generated randomly. These solutions are represented as chromosomes, which are essentially strings of genetic information. Each chromosome contains a set of parameters that define a particular solution to the problem. The GA proceeds through a series of iterations, known as generations. In each generation, the fitness of each chromosome in the population is evaluated, which measures how well the solution represented by that chromosome solves the problem. Chromosomes with higher fitness values are more likely to be selected for further processing. The GA then applies a set of genetic operators, such as crossover and mutation, to the selected chromosomes. Crossover involves exchanging genetic information between two chromosomes to create new offspring, while mutation involves randomly changing a portion of a chromosome. These operators are designed to simulate the natural process of genetic recombination and mutation. The offspring produced by the genetic operators are then added to the

population, replacing some of the existing chromosomes. The process of fitness evaluation, selection, and genetic operators is repeated over multiple generations, with the hope that the population will evolve towards better and better solutions to the problem. Eventually, the GA will converge on a population of high-quality solutions to the problem, from which the best solution can be selected. The genetic algorithm can be a powerful tool for optimisation problems, particularly when the search space is large and complex, and traditional optimisation techniques are not effective.

PSO (Bozorg-Haddad et al., 2017) is a population-based stochastic optimisation algorithm that was developed based on the social behaviour of birds flocking or fish schooling. The algorithm starts by randomly generating an initial population of candidate solutions to the optimisation problem, which are represented as particles in the search space. Each particle is evaluated according to an objective function, and then updated based on its own best-known position (p_{best}) and the best-known position among all the particles in the population (g_{best}). The particles adjust their positions in the search space by updating their velocity based on their own position, their p_{best}, and the g_{best}. The PSO algorithm has two key parameters: the cognitive parameter and the social parameter. The cognitive parameter determines the influence of the particle's own best-known position on its movement, while the social parameter determines the influence of the best-known position among all the particles in the population. The algorithm iteratively updates the position and velocity of the particles until a stopping criterion is met. The stopping criterion can be a maximum number of iterations, a threshold for the objective function, or other user-defined criteria. PSO has several advantages, such as its simplicity, ease of implementation, and ability to handle both continuous and discrete variables. It can also find the global optimum in some cases, although this is not guaranteed. However, PSO can suffer from either premature convergence or slow convergence, and the optimal values of the cognitive and social parameters can be problem-dependent.

2.7.2 Applications of optimisation algorithms in airfoil performance optimisation

In the study of wind turbines, a GA was successfully applied in an airfoil optimisation procedure which yielded airfoils with performance greater than those of the original or baseline airfoils (Bedon et al., 2016; Daróczy et al., 2018). This motivates the use of vanilla GA methods for complex multivariate optimisation problems. The vanilla GA is, however, slow to converge and a global optimum is not guaranteed. In an attempt to mitigate this, an improved version of the GA called the NSGA-II (Non-Dominated Sorting Genetic Algorithm II) was developed and used to improve the lift-drag-ratio and cavitation performance of marine current turbine hydrofoils (Luo et al., 2014).

Other notable studies conducted, highlighting the applicability of GA for multi-objective airfoil optimisation across diverse airfoil profile shapes, can be seen below. For thick airfoils with application in wind turbines, the GA was used to improve the lift-drag ratio by 30-40% compared to the original airfoils (Jeong & Kim, 2018). To counter the soiling effect, the GA was used to design and optimise airfoil sections for a 20 kW wind turbine. This resulted in a 2.3% improvement in the turbine's yearly energy production with consistent performance in soiled conditions (K. Ram et al., 2019). In (Saleem & Kim, 2020), Saleem and Kim optimised the shell of an airborne wind turbine using a genetic algorithm, which integrated an airfoil parametrisation method along with Xfoil into the optimisation code. It was shown that the optimised shell configuration resulted in a power augmentation, demonstrating the effectiveness of the GA-based optimisation framework. In (K. R. Ram et al., 2019), Ram et al. showcased the use of a GA in the design of blades for a 20 kW wind turbine. The objective was maximising the lift-to-drag ratio under both clean and soiled conditions. The resulting optimised airfoils resulted in higher annual energy production and were more resistant to soiling ensuring consistent annual energy production. This highlights the capacity of the GA to find optimal designs in highly non-linear optimisation problems.

A variation of the particle swarm algorithm called the slap swarm algorithm (SSA) was

used to optimise two-dimensional airfoil lift-to-drag ratio. The findings demonstrated that the SSA is able to approximate the global optima of real-world single-objective problems with challenging and unknown search spaces (Mirjalili et al., 2017). Khurana et al. (M. Khurana & Massey, 2015) showed that a modification to the traditional particle swarm algorithm results in reduced computational requirements for two-dimensional airfoil optimisation, compared to the optimisation method documented in the literature. Huan et al. (Huan et al., 2017) used a PS optimisation framework to optimise a base airfoil by setting three cases, each with different constraints and objectives. Results indicate that for all cases investigated the optimised airfoil outperformed the base airfoil in terms of drag counts, laminar flow length and moment coefficient.

2.8 Machine learning and deep learning model architectures

This section gives an overview of the three main machine learning and deep learning architectures used in this research, i.e. the ANN, the Bidirectional Long Short Term Memory (BLSTM), and the BGMM.

The first architecture discussed is the ANN, which is inspired by the biological neural networks in the human brain. The focus is specifically on an MLP model. An MLP, which is a type of ANN with multiple layers of interconnected neurons, is widely used in image classification, speech recognition, and natural language processing. In this research this model is used in the development of the surrogate model for the 2D airfoil analysis.

The second architecture discussed is the BLSTM, which is a type of Recurrent Neural Network (RNN) that can process the input sequence in both forward and backward directions. In this research, a BLSTM model is used in the development of the airfoil shape mapping module which enables mapping from the low dimensional parsimonious variable domain to the high dimensional raw x and y coordinate domain.

The third architecture discussed is the BGMM, which is a probabilistic model that rep-

resents a probability distribution as a mixture of several Gaussian distributions. In this research, this model is used to develop the airfoil generator module that is capable of generating an airfoil in terms of its low order parsimonious variable representation.

2.8.1 Artificial neural network model architecture overview

ANNs are a type of machine learning model that are inspired by the structure and function of biological neural networks in the brain. ANNs consist of interconnected nodes, or neurons, that process information through weighted connections and activation functions.

The MLP is a type of ANN that consists of multiple layers of interconnected neurons, including an input layer, one or more hidden layers, and an output layer. The input layer receives the input data, and the output layer produces the final predictions or outputs. The hidden layers process the input data through nonlinear transformations, allowing the network to learn complex patterns and relationships in the data. Mathematically, the hidden neuron of an MLP can be expressed as:

$$Z_i = \left(\sum_{k=1}^{N_{j-1}} X_k^{j-1} W_{k,i} - b_k \right) \tag{2.33}$$

where X_k^{j-1} represents the $k-th$ node input, $W_{k,i}$ is the weight of the link between node k and all the nodes in the previous layers, and b_k is the bias for the node. The output of Equation 2.33 is then passed to the activation function in order to produce the final output for the specific node. This can be represented as:

$$Y_i = f(Z_i) \tag{2.34}$$

where $f(Z_i)$ is the activation function that can take any arbitrarily defined form - the most popular of which the sigmoidal, tanh, or ReLU functions (Schmidt-Hieber, 2020).

To find the best performing MLP model, an optimisation routine is implemented. This optimisation of an MLP, involves adjusting the weights and biases of the neurons in the network to minimise a loss function, which measures the difference between the predicted outputs and the true outputs. This is typically done using an optimisation algorithm such

as gradient descent, which iteratively adjusts the weights and biases to move towards the minimum of the loss function.

The universal approximation theorem (Hornik et al., 1989) states that a feedforward neural network with a single hidden layer can approximate any continuous function on a compact subset of Euclidean space, under mild assumptions on the activation function. This means that theoretically, an ANN can fit any function. However, in practice, it can be challenging to optimise ANNs for complex functions, and overfitting can occur if the model becomes too complex. Additionally, the curse of dimensionality can make it difficult for ANNs to generalise to high-dimensional data. Therefore, although ANNs have the potential to fit any function, it may not always be feasible or practical to do so in practice.

In Table 2.3 the strengths and weaknesses of MLPs are summarised. As can be seen from this table, the main advantages of MLPs are that they can model highly non-linear functions with multiple inputs and multiple outputs, each with different dimensions and data modalities. The main drawback of using MLPs, in context of this research, is the computational intensity seeing that only a limited computational budget is available. The problems of overfitting and hyperparameter sensitivity are easily overcome by following best practice in the model optimisation phase, as seen in Section 6.4. It is also important to note that in this research the issue of data size is not a limiting factor, seeing that a custom data mining pipeline is developed that enables mining as many data points as required, see Section 5.2.

Table 2.3: Strengths and Weaknesses of MLPs

Strengths	Weaknesses
Non-linearity: MLPs can model complex non-linear relationships in data, making them suitable for regression problems with intricate patterns.	**Overfitting:** MLPs are prone to overfitting, especially when the model is too complex or when the training dataset is small. Regularization techniques such as dropout and weight decay are often needed to mitigate this issue.
Feature Learning: MLPs can automatically learn and extract relevant features from the input data, reducing the need for manual feature engineering.	**Computational Intensity:** Training deep MLPs can be computationally intensive, requiring significant computing resources, especially for large datasets.
Universal Approximator: According to the Universal Approximation Theorem, MLPs can approximate any continuous function given enough hidden units, allowing them to capture a wide range of relationships in the data.	**Hyperparameter Sensitivity:** MLPs have various hyperparameters (e.g., learning rate, number of layers, number of neurons per layer) that need careful tuning. The performance of the model is sensitive to these hyperparameters, and finding the optimal set can be challenging.
Parallel Processing: MLPs can process multiple inputs simultaneously, making them efficient for parallel computations and speeding up training on hardware like GPUs.	**Data Requirements:** MLPs often require large amounts of data to generalize well. In cases where the dataset is limited, the model may struggle to capture the underlying patterns accurately.
Flexibility: The architecture of MLPs can be customized by adjusting the number of layers and neurons, allowing for flexibility in adapting to different regression problem complexities.	**Interpretability:** The complex nature of MLPs can make them less interpretable compared to simpler regression models. Understanding the relationships learned by the model may be challenging, which can be a drawback in some applications.

2.8.2 Bi-directional long short term memory model architecture overview

In contrast to Vanilla LSTM models that have a forward architecture to model sequential data with long-term dependencies, BLSTMs have both forward and backward architectures, allowing the model to process the input sequence in both directions simultaneously (Graves & Schmidhuber, 2005). Specifically, BLSTMs consist of two LSTMs layers, one processing the input sequence in the forward direction, and the other processing it in the backward direction (see Figure 2.3).

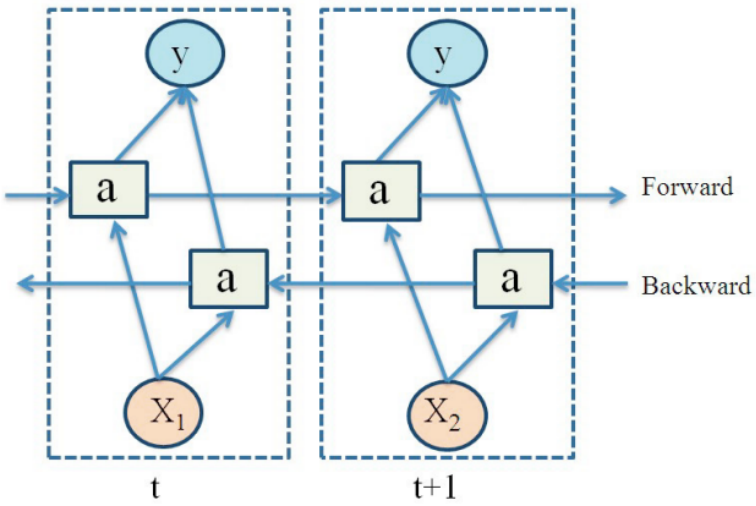


Figure 2.3: Bidirectional LSTM network architecture (Schmidt-Hieber, 2020)

As seen in Figure 2.3, each layer has its own set of weights for the forward pass and the backward pass. The outputs of both layers are concatenated at each time step to obtain the final output. As mentioned, this forward and backward processing capability means that a BLSTM can capture information from both past and future contexts, which is a significant advantage over a vanilla LSTM which can only process information with past context (Graves & Schmidhuber, 2005). This can be particularly useful for tasks where the context of a word is important, such as in natural language processing applications like sentiment analysis and named entity recognition. By processing the input sequence in both directions, BLSTMs can better capture the meaning of a word based on its surrounding words, regardless of their relative positions in the sequence. The same can be said for parsimonious variables where the higher order dimensions can more accurately be mapped when not only a single parsimonious variable is considered but also its surrounding variables.

However, BLSTMs also have some disadvantages compared to vanilla LSTMs. First, they are more computationally expensive due to the need to process the input sequence in both directions. Additionally, the model's output at each time step depends on both past and future inputs, which can make it harder to interpret the model's predictions. Finally, BLSTMs may not be suitable for tasks where the future context is not available

or not relevant, such as online prediction or anomaly detection. These disadvantages of a BLSTM do however not limit their application in this research seeing that: (1) no future context is required in the sequence predictions of an airfoil's raw x and y coordinate pairs, (2) there is no requirement to understand why the model make certain predictions and hence interpretability of the outputs can be disregarded, and (3) given the relative simplicity of predicting an airfoil's profile from parsimonious variables, in comparison with other natural language processing tasks, and considering that the data sets for this task will not be significant in size, the problem of computational complexity can be disregard.

Other than the aforementioned advantages of using BLSTM models for sequence-to-sequence predictions, below follows a list highlighting several additional properties of these models that further motivates their use in this research:

- **Memory retention:** BLSTMs can retain information over long periods, aiding in capturing context from distant points in the sequence for accurate prediction.
- **Robust to noisy data:** The memory cells and gating mechanisms in LSTM units allow the network to filter out noise and focus on relevant information. This makes BLSTMs robust to noisy data, enhancing their performance in regression problems where the input may contain uncertainties or irrelevant details.
- **Variable input length:** BLSTM networks handle sequences of varying lengths, providing flexibility for regression problems with variable-length input sequences.
- **Non-linearity:** The non-linear activation functions within the LSTM units enable the network to learn complex relationships in the data.
- **Adaptability to irregular sampling rates:** BLSTMs can effectively handle irregularly sampled sequence data. In regression problems where the input features are not uniformly spaced, the bidirectional processing can adapt to the varying intervals between observations, providing flexibility in real-world scenarios.

2.8.3 Bayesian Gaussian mixture model architecture overview

BGMMs are an extension of the Gaussian Mixture Model (GMM) which is a probabilistic model that represents a probability distribution as a mixture of several Gaussian (normal) distributions (Reynolds et al., 2009). GMMs are often used in unsupervised learning tasks such as clustering and density estimation, where the goal is to identify the underlying clusters or probability density function of the observed data.

In a GMM, the number of mixture components need to be specified upfront – which directly influences the quality of the clusters formed and hence the accuracy of the model. With BGMMs, Bayesian inference is incorporated into the framework, allowing for the automatic selection of the number of mixture components in the model. Hence in BGMMs, it is assumed that the number of mixture components is itself a random variable, and a prior distribution is assigned to this variable (Roberts et al., 1998). The prior distribution reflects the prior belief about the number of mixture components in the model. During model training, the posterior distribution over the number of mixture components is updated based on the observed data, using Bayes' rule. Specifically, the posterior distribution is proportional to the prior distribution multiplied by the likelihood of the observed data given the model parameters and the number of mixture components.

In this research, the BGMM architecture is used in the airfoil generator module that is tasked with generating an airfoil sample in terms of its lower order parsimonious variable representation. It is therefore of great importance that this sampled airfoil represent the parsimonious variable characteristics of a realistic airfoil. That is, the sample should exhibit the same covariance and hence global and local structure present in the training set. BGMMs are well-suited for modeling covariance and capturing both global and local structures in data due to their probabilistic nature and the incorporation of uncertainty through a Bayesian framework. For this and the below listed reasons, the BGMM is selected in this research as the architecture of choice for the airfoil generator module:

- **Probabilistic modeling of data:**
  - BGMMs treat each data point as a probabilistic distribution, capturing inher-

ent uncertainty in the data.

  - The framework's probabilistic nature allows representation of complex structures and dependencies between variables.

- **Flexibility in capturing covariance:**

  - BGMMs can model various shapes of clusters and accommodate different degrees of covariance.
  - Covariance matrices in Gaussian components allow modeling both diagonal and off-diagonal elements, capturing correlations.

- **Hierarchical structure:**

  - BGMMs naturally model hierarchical structures, with mixture components representing different levels of granularity.
  - Hierarchical structure captures both global patterns (overall mixture) and local structures within each component.

- **Adaptability to complex distributions:**

  - The framework's Bayesian approach allows adaptation to the complexity of the underlying data distribution.
  - Automatic determination of complexity and number of components helps avoid overfitting or underfitting.

- **Incorporation of prior knowledge:**

  - Bayesian models allow incorporation of prior knowledge through the choice of prior distributions.
  - This is useful in scenarios where some information about the data distribution is known a priori.

2.9 Conclusion

In this chapter a literature overview is given of all the relevant components used in this thesis. This includes background on:

1. The derivation of the sailplane cross-country model.
2. The LLT model used to calculate the 3D lift of a wing.
3. The various ML and DL architectures used in this research.
4. The various airfoil parametrisation techniques (both traditional and deep learning-based).
5. The methods used to analyse the performance of a 2D airfoil, both from a traditional and deep learning-based viewpoint.
6. Background on surrogate modelling.
7. Model search and optimisation methods employed in 2D airfoil performance optimisation.

In the following chapters, the literature discussed here will serve as the foundation for various design choices and component implementations. Specifically it will be shown how the cross-country model of Section 2.2 and the LLT model of Section 2.3 are combined to form the TSM as detailed in Section 1.3.1 and Figure 1.2. Furthermore it will be shown how the SPM discussed in Section 2.4.1 is used to develop a parsimonious variable extraction framework, as well as a data mining pipeline to enable the extraction of various airfoil profile and 2D airfoil performance data information. The ANN discussed in Section 2.8.1 will be leveraged to build the required deep learning based airfoil performance analysis module. Also, the models discussed in Sections 2.8.2 and 2.8.3 will be used to develop the airfoil generator module and the airfoil shape conversion module as depicted in Figure 1.2. Finally it will be shown how the optimisation algorithms, specifically the GA, discussed

in Section 2.7.1 are used to facilitate end-to-end sailplane cross-country performance optimisation with intuitive constraints imposed on both the airfoil- and wing geometry of the final sailplane design.

Chapter 3

Building the required sub-components

3.1 Introduction

In this chapter, the development, validation, and integration of the sub-components required in the TSM are discussed. This model serves as a baseline which will later be transformed into the DSM by replacing some of the modules in the traditional model with specific redesigned DL-based modules.

The TSM and its components are introduced in Section 1.3.1, as illustrated in Figure 1.2 and listed in Table 1.1. In the following sections, the development of these modules are discussed individually. Specifically, the airfoil generator module (Section 3.2), the airfoil performance analysis module (Section 3.3), the sailplane performance analysis module (Section 3.4), and the integration of these modules to form the TSM capable of predicting the average cross-country performance for a given cross-country task (Section 3.5) are described. Finally, in Section 3.6, the accuracy of the TSM is verified with both experimental and simulation results.

3.2 Airfoil generator module

The main objective of the (traditional) airfoil generator module is to produce practical airfoil profiles. It does so by generating a set of x and y coordinate pairs that describe the 2D geometry of the airfoil. There are numerous airfoil generation methods available, as seen in Sections 2.4.1 and 2.4.2, and the choice of generation method is typically a result of the particular use case. For this study, the airfoil generation technique used is chosen to satisfy a set of self-defined criteria. In the following sections, we describe the criteria that the airfoil generation technique should fulfil, the selected method, and the extent to which the method adheres to the required criteria.

3.2.1 Selection criteria

The criteria for selecting an airfoil generation technique are defined so that the chosen method allows for efficient and effective data mining. The chosen criteria can be summarised as in Tbale 3.1:

Table 3.1: Criteria for selection of an effective and efficient airfoil generation technique

Criteria	Description
Ease of use	The chosen method should be platform agnostic. That is, the method should be easily implementable in a modern scientific programming language.
Flexibility	The method should be flexible in diverse airfoil profile representation.
Coordinate control	The method should allow the user to adjust the airfoil features on a global, and to an extent, local scale when its inputs are adjusted.
Parsimonious variables	The parameterised variables should be as few as possible, have an easily understandable meaning, and affect a physical characteristic in the airfoil geometry.
Camber and thickness description	The chosen method should provide a means of computing the mean camber and thickness distribution.
Computational complexity	The method should allow for (near) real-time airfoil generation when the function arguments are adjusted; i.e. it should be computationally efficient w.r.t airfoil generation.

Having an airfoil generation technique that adheres to the above-mentioned criteria allows for efficient and diverse airfoil generation, which facilitates the airfoil performance data mining, and therefore enables the training of a deep learning model with good generalisation capacity.

3.2.2 Selected method

Given the defined criteria, the SPM proposed by Ziemkiewicz (Ziemkiewicz, 2017) is selected from the methods described in Section 2.4.1. Inspired by the PARSEC method of Sobieczky (Sobieczky, 1999), Ziemkiewicz proposed a simple parametric model which models not only the airfoil upper and lower surface coordinates but also the thickness distribution and the camber line. This method generates the x-coordinates, y-coordinates, thickness distribution, and camber line as a function of six parameters. The six global parameters include nose shape, thickness, camber and reflex. According to this method, the x and y coordinates can be obtained by applying the following two equations:

$$X(\theta) = \frac{1}{2} + \frac{1}{2}\frac{|cos\theta|^B}{cos\theta} \tag{3.1}$$

$$Y(\theta) = \frac{T}{2}\frac{|sin\theta|^B}{sin\theta}\left(1 - X^P\right) + Csin\left(X^E\pi\right) + Rsin(X2\pi) \tag{3.2}$$

where θ varies between 0 and 2π. The design parameters of the SPM along with a brief description and its corresponding usage properties (for most conventional airfoil descriptions) can be seen in Table 3.2.

Table 3.2: SPM's parameter definition and brief descriptions (Ziemkiewicz, 2017)

Parameter	Description	Property
B	Base shape coefficient	For B=2, the base shape of the airfoil is an ellipse. In the limit of B approaching one, the shape becomes a rectangle. This parameter affects mostly the leading edge. This parameter has a lower limit of 1.6 and an upper limit of 2.4
T	Thickness coefficient	This parameter affects thickness as a fraction of the chord.
P	Taper exponent	For P=1, when going to the trailing edge, the thickness approaches 0 in a linear manner. For higher values, the airfoil tapers more suddenly near the trailing edge.
E	Camber exponent	Defining the location of the maximum camber point, where $E = 1$ describes 50 % camber point, and a smaller value shifts it towards the leading edge, and vice versa.
C	Camber	This parameter affects camber as a fraction of the chord.
R	Reflex parameter	Positive value generates reflexed trailing edge, while negative value emulates flaps.

3.2.3 Method evaluation

The SPM's adherence to the criteria set out in Section 3.2.1 is demonstrated to motivate its choice as the constructive airfoil generation technique to be used in this research.

The first criterion to be evaluated is the **ease of use**. As seen in Equations 3.1 and 3.2, there are six design parameters, namely B, T, P, C, E and R. These design parameters represent the base shape coefficient, the thickness as a fraction of the chord, the taper exponent, the camber, the camber exponent and the reflex parameter respectively. With the SPM, modifying the airfoil shape is as simple as adjusting the values of these six design parameters, within the bounds set when the SPM was derived (see Table 3.2). This simplicity in airfoil generation of the SPM shows that the method extends itself to be easily implementable in any general purpose or scientific programming language and therefore highlights its adherence to the **ease of use** criterion.

The second criterion that the SPM is shown to adhere to is the **flexibility** criterion. For

this analysis, attention is given to Figure 3.1. Here, the profiles of 30 different airfoils are displayed. These airfoils are generated by fixing five of the SPM design variables and then perturbing the remaining design variable incrementally. From this figure, it can be observed that a slight change in one of the SPMs global design parameters results in a significant variation in an airfoil shape. From this figure, it is also noted that certain combinations of these six design parameters result in airfoils with non-conventional or impractical shapes. Examples of these impractical airfoils are those that have highly curved trailing or leading edges compared to the mean thickness distribution of the airfoil, see Figure 3.1 for two examples of these types of airfoils. We address these impractical shapes in Section 5.3.

Although not all the airfoils of Figure 3.1 are practical, their diversity in profiles highlights the versatility and flexibility of the SPM method in terms of the complexity of airfoil shapes that this method can produce. This characteristic is also further motivated in (Ziemkiewicz, 2017) where the researchers show how the SPM, when used in an optimisation pipeline, has the advantage of a well-behaved design space containing diverse, practical airfoil shapes. This, therefore, indicates that the SPM method adheres to both the **flexibility** and **coordinate control** criteria. That is, the SPM method is able to represent a diverse range of airfoils by varying one or all six of the function's input parameters.

Figure 3.1 also indicates that changing one of the SPMs function inputs whilst keeping the remaining five inputs constant, results in a predictable shape alteration. This is because the SPM was derived with parsimony in mind. The six global design parameters thus have (fairly) intuitive meanings. This property of the SPM indicates that the method adheres to the **parsimonious variables** criterion set for our chosen airfoil generation method.

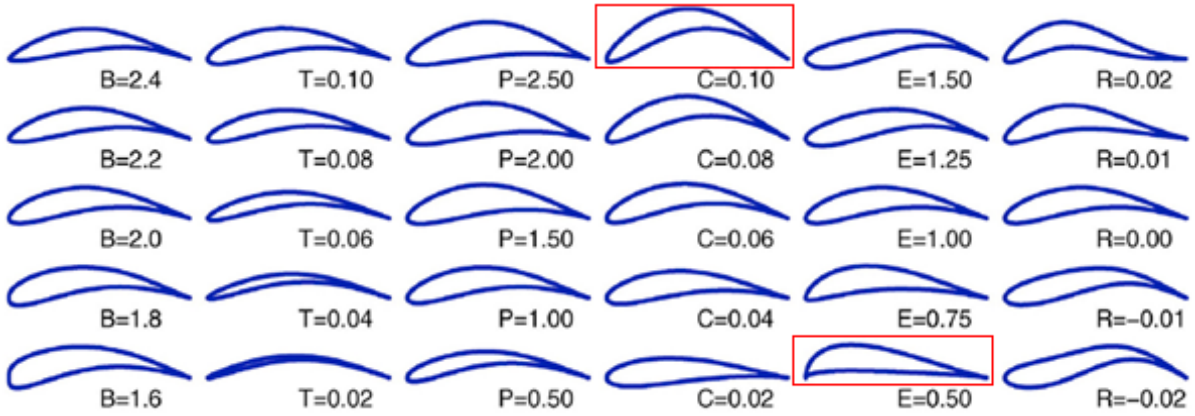


Figure 3.1: SPM shape manipulation with B=2, T=0.1, C=0.05, P=1, E=1, and R=0 as the base curve, the shape changes according to parameters' adjustment (Ziemkiewicz, 2017). Red-boxed airfoils indicate examples of unconventional airfoil profiles.

For the evaluation of the **camber**- and **thickness** description criterion, it is first required that Equations 3.3 and 3.4 be discussed. These two equations highlight the SPM's additional capability of modelling the camber distribution C_M and airfoil thickness distribution T_h separately.

$$T_h(\theta) = T\frac{|sin(\theta)|^B}{sin(\theta)}(1 - X^B) \tag{3.3}$$

$$C_M(\theta) = Csin(X^E\pi) + Rsin(X2\pi) \tag{3.4}$$

Where:

$$\theta \ \in [0, \ \pi] \tag{3.5}$$

The fact that the SPM allows for the separate calculation of the thickness and camber line distribution, means that it complies with the **camber**- and **thickness** description criterion.

Finally, seeing that the SPM makes use of only simple parametric functions to calculate the thickness distribution, camber line, and the x and y-coordinates, it can be concluded that the method is lightweight i.e. computationally inexpensive when it comes to airfoil

profile generation. With this inexpensive airfoil generation property, the SPM method, therefore, complies with the **computational complexity** criterion.

This compliance of the SPM to all of the six defined criteria means that it is a well-suited method for airfoil generation in the context of this specific study.

3.3 Airfoil performance analysis module

The airfoil performance analysis module plays an instrumental role in this research. This module is not only used in the TSM to facilitate the calculation of the cross-country performance but it is also used to mine performance data for the development of the PDLM, and therefore the Inverse 2D Performance Deep Learning Module (IPDLM) (see Section 6.8). The chosen airfoil performance analysis module, therefore, is required to have the following characteristics:

1. It must be computationally inexpensive given the number of simulations that need to be run for the data mining objectives, and average cross-country performance calculations.
2. The module should be flexible in the shapes and operating conditions it can simulate to ensure a diverse airfoil population is present in the final surrogate modelling datasets.
3. It should be accurate in the calculation of 2D performance coefficients to ensure there is an accurate and strong signal for the surrogate model to learn.

As mentioned in Section 2.6.1, in this research, the vortex panel code of XFOIL (Drela, 1989) is used as the airfoil analysis software of choice. This is because XFOIL adheres to all the specified criteria – as motivated in the following sections of this chapter.

3.3.1 Simulation computational complexity

This section is concerned with the simulation computational complexity of XFOIL. Simulation complexity refers to the efficiency, or speed, at which the 2D airfoil performance coefficients for a given airfoil with a set of operating conditions can be calculated. Hence the more efficient the method the less time to get the results for a given simulation.

In the prediction of the aerodynamic characteristics of a 2D airfoil, XFOIL makes use of a panel-based method, the potential-flow solution that is coupled with an integral boundary layer. The panel-based formulation allows for XFOIL to rapidly calculate the polars of a given airfoil and flight condition (J.Morgadoa et al., 2016; Zhang et al., 2019). Calculating a single polar with XFOIL, where the airfoil is discretised in 250 panels, typically takes between 1-10 seconds on a machine with 2-16 Gb of RAM operating at a processing speed between 1.0 and 4.0 GHz. This computational efficiency of the XFOIL panel code, therefore, motivates its use for data mining, and expensive cross-country simulations.

3.3.2 Simulation flexibility

This section is concerned with the simulation flexibility verification of XFOIL. In this research, simulation flexibility refers to the diversity in airfoil profiles and operating conditions that can be simulated by a given software solution.

Emphasising the flexibility in XFOIL simulation use cases, readers can refer to the work of Wei et al. (Xuesong et al., 2020). Wei et al. used Bezier curves and a GA to optimise the performance of the E387 low Reynolds number airfoil. In this, the authors used the Bezier method and XFOIL to respectively generate and evaluate 3 601 airfoils within the upper and lower bounds as shown in Table 3.3. These wide boundaries set on the Bezier control points resulted in diverse airfoils being generated in the optimisation pipeline. XFOIL successfully converged in the simulation of each of these airfoils, attesting to XFOIL's characteristic of simulation flexibility. This same property of XFOIL's simulation flexibility can be seen in the works of (Anitha et al., 2018; S. Khurana et al., 2008;

Xiaoqiang et al., 2018; Zhang et al., 2019; Ziemkiewicz, 2017). These authors clearly highlight how XFOIL could be applied to calculate the performance of a wide range of different airfoil geometries. This, therefore, motivates XFOIL's use for 2D performance data mining where a diverse airfoil population is to be simulated to ensure adequate variability is present in the final 2D airfoil performance dataset.

Table 3.3: Bezier curve upper and lower bounds used to generate the 3 601 airfoils used in (Xuesong et al., 2020).

Control points – upper curve	Lower limit	Upper limit	Control points – lower curve	Lower limit	Upper limit
$P1_x$	0.674	0.874	$P6_x$	-0.001	0.099
$P1_y$	0.026	0.046	$P6_y$	-0.035	-0.015
$P2_x$	0.363	0.563	$P7_x$	0.087	0.287
$P2_y$	0.073	0.093	$P7_y$	-0.029	-0.009
$P3_x$	0.294	0.494	$P8_x$	0.227	0.427
$P3_y$	0.116	0.136	$P8_y$	-0.012	0.008
$P4_x$	0.014	0.114	$P9_x$	0.699	0.899
$P4_y$	0.054	0.074	$P9_y$	0.001	0.021

3.3.3 Simulation accuracy

This section is concerned with the accuracy validation of XFOIL under 2D airfoil performance coefficient prediction. In this research, simulation accuracy simply refers to how accurate the predicted performance coefficients of the final converged simulation are when compared to ground truth, i.e. experimental results.

The validation of XFOIL for low Reynolds number use cases has been done in numerous studies. One such notable study that is worth discussing is the work of J.Morgadoa et al., 2016. This study is of particular importance for three specific reasons:

1. The study is concerned with the E387 and S1223 airfoils. These airfoils represent profiles that are typically used in sailplane applications.

2. The study made use of wind tunnel experiments as well as CFD simulations with 3

different transition models to validate XFOIL's accuracy.

3. The accuracy validation was done over an AoA range of -5° to 18° with a Reynolds number of 2 000 000 – which is typically where sailplanes operate in cross-country flight.

The comparison between the XFOIL simulation, the CFD simulation, and the wind tunnel experimental results for the E387 airfoil can be seen in Figure 3.2. From this figure, it is observed that XFOIL accurately predicts the lift and drag coefficient of the airfoil for all the AoA cases simulated. Here it can also be seen that, in the linear region, XFOIL predictions are highly correlated with the experimental results. Finally, it can be observed that both XFOIL and the k-ω SST turbulence model with low Reynolds corrections (k-ω SST Low Re) accurately predict the corners of the low drag region, a region of complex flow phenomena.

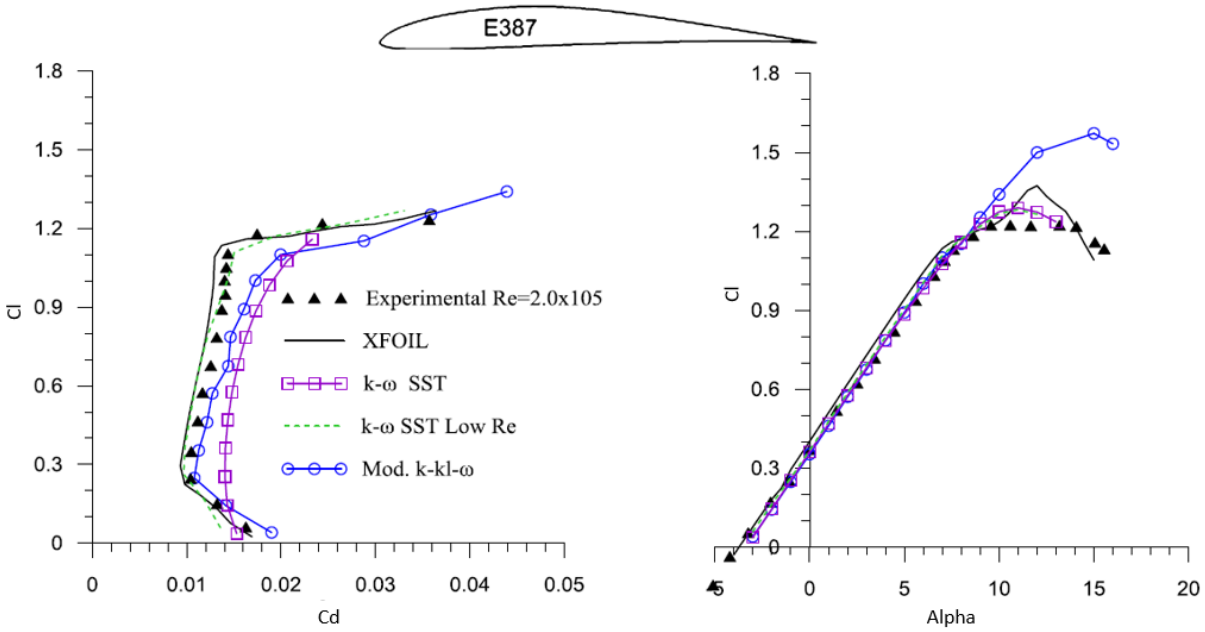


Figure 3.2: Aerodynamic characteristics of the E387 airfoil measured at Pennsylvania State University wind tunnel (J.Morgadoa et al., 2016; S. Selig & Guglielmo, 1997) compared to the numerical simulation results

The comparison between the XFOIL simulation, the CFD simulation, and the wind tunnel

experimental results for the S1223 airfoil can be seen in Figure 3.3. Figure 3.3 shows that, although there is good agreement between the simulation results and the wind tunnel results, the correlation is weaker than that of the E387 case. This is because of the higher degree of flow separation present in the S1223 simulation case. It is a known tendency of XFOIL's accuracy to degrade as the degree of flow separation increases (Coder & Maughmer, 2014; Drela, 1989). Typically, high flow separation conditions are present at high angles of attack, which is the region where we typically find the maximum lift coefficient. As a result, XFOIL tends to overpredict the maximum lift when compared to experimental and RANS-CFD results. This is also observed in Figure 3.2. This known XFOIL tendency does not, however, influence the current work seeing that sailplanes do not traditionally operate near stall conditions during cross-country flight and hence this research's 2D simulations are rarely in areas of high separation. Other than this limitation, XFOIL has proven to be accurate in performance coefficient prediction as further highlighted in Figures 3.2 and 3.3 and the works of (D'Alessandro et al., 2017; El-Shahat et al., 2020; J.Morgadoa et al., 2016; X. Li et al., 2020; Mohammadi et al., 2020; Xuesong et al., 2020; Zhang et al., 2019).

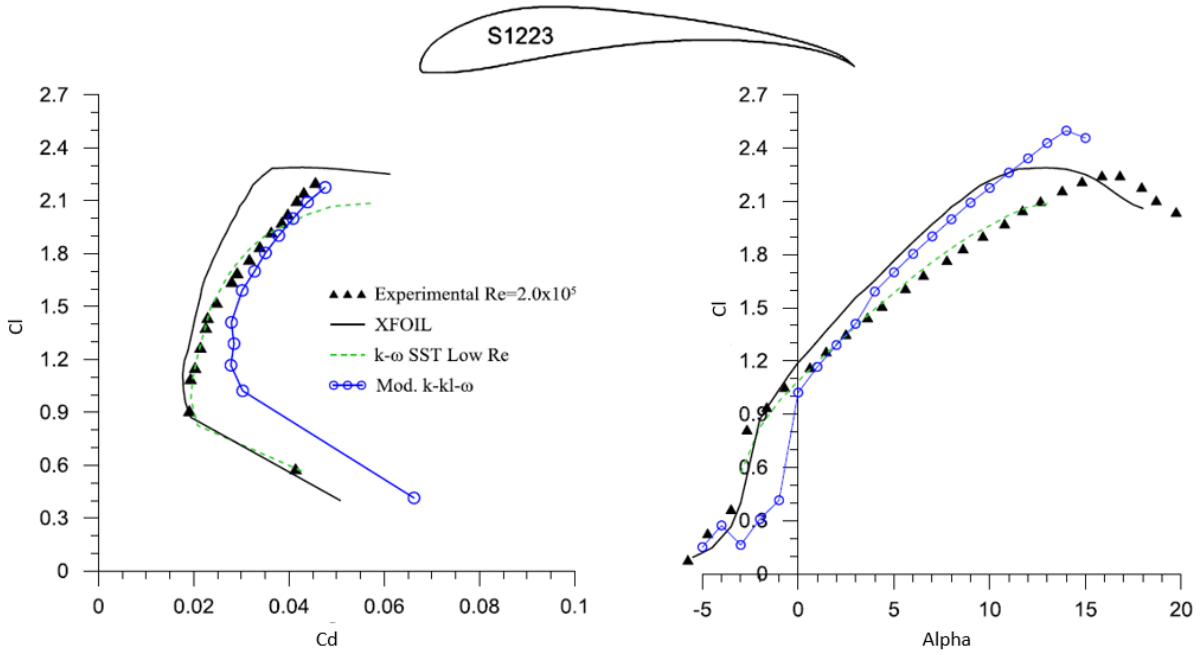


Figure 3.3: Aerodynamic characteristics of the S1223 airfoil measured at Pennsylvania State University wind tunnel (J.Morgadoa et al., 2016; S. Selig & Guglielmo, 1997) compared to the numerical simulation results

With XFOIL accurate in 2D airfoil performance prediction, being computationally efficient, and having the capability of simulating a diverse population of airfoils, its use as the airfoil performance analysis module in both the data mining objective and the TSM development is therefore motivated.

3.4 Sailplane performance analysis module

The sailplane performance analysis module is used to obtain the aerodynamic properties of a given sailplane design. This module accounts for the 3D flow effects of the wing, winglet, fuselage and tail (see Figure 3.4 for a graphical depiction of these elements).

In an attempt to simplify the simulation, the effects of flaps and slats on the cross-country performance of a sailplane are excluded from this research.

It is also important to note that, seeing that there are complex flow phenomena present in the fuselage and tail configuration, the design of these elements is fixed for this research. The performance of these elements was modelled separately with CFD simulations at multiple flight speeds and generously made available by Jonker Sailplanes for use in this research.

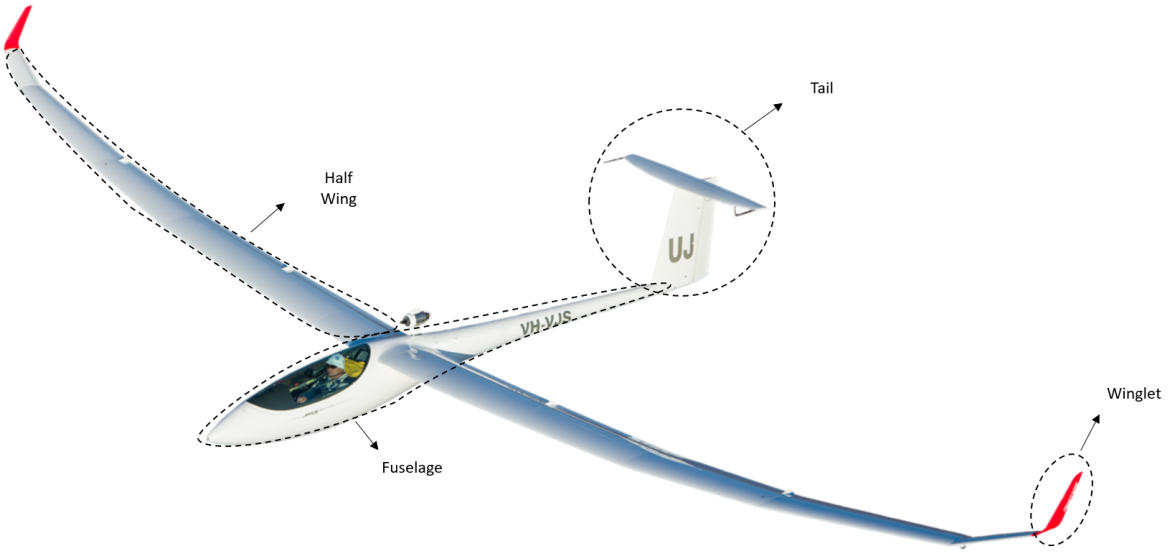


Figure 3.4: Sailplane design elements that will be included in the sailplane performance analysis procedure

At a high level, the sailplane performance analysis module consists of two main elements: (1) a 3D lift calculation module, and (2) a 3D drag calculation module. The 3D lift calculation module makes use of a modified LLT model to calculate the sailplane's 3D lift, wing efficiency, and induced drag (drag resulting from the production of lift). The 3D drag calculation module makes use of the outputs of the modified LLT model together with the fuselage/empennage configuration performance assumptions, wing geometry, and 2D section-wise airfoil performance, to calculate the overall sailplane 3D drag performance.

The following sections in this chapter detail the design of these two elements and how they are integrated with a cross-country model (derived in Section 2.2) to form the final TSM.

3.4.1 3D lift calculation

The 3D lift calculation sub-module is responsible mainly for calculating the total 3D lift of the entire sailplane wing. In this research, the classical LLT model as discussed in Section 2.3 is used to calculate the sailplane's 3D lift, lift distribution along the wing span, planform efficiency, and induced drag. The use of the classic lifting line theory of Prandtl (Prandtl, 1921; Prandtl, 1918) is adequate for this research as the method pertains to high-aspect-ratio wings in steady, incompressible flows with relatively small angles of attack (Reid & Hunsaker, 2021) – which is suitable for the operational range of a sailplane during cross-country flights. The problem with the classic implementation of the LLT is that only basic wing geometries can be described by means of varying the taper parameter. This results in stations with linearly varying chords and wingspan increments. In practice, such a wing description is very restrictive as the aerodynamicist wishes to adjust the individual station and chord lengths, along with other design parameters such as wing twist, station wise airfoils, etc. to maximise the wings' performance for a given operating condition. Hence a modified LLT model is used in this research to address this discrepancy of the classical model. This variation allows for the specification of wings with nonlinearly varying wingspan increments and chords, enabling the specification of realistic wing geometries.

The LLT model developed and used in this research, referred to as the dynamic station LLT model, requires the arguments of Table 3.4 as model inputs to calculate the lift-related performance characteristics and uses the model derived in Section 2.3.

Table 3.4: Inputs as required by the dynamic station LLT to determine the sailplane's 3D lift, lift distribution along the wing span, planform efficiency, and induced drag.

Parameter	Typical range
Number of stations	[1, 5] $\in \mathbb{Z}^+$
Number of numerical wing subdivisions	[200 - 1 000] $\in \mathbb{Z}^+$
Wingspan increments	[0.1 m, 3 m]
Chord lengths	[0.1 m, 1.25 m]
Zero-lift AoA per station	[-10 °, 8 °]
Lift-curve slope per station	[5.5 1/rad, 8 1/rad]

The process followed to specify the individual wingspan increments and chords such that more complex planforms are defined in the dynamic station LLT is as follows:

1. **Calculate wing half span locations:**

   - Given wingspan increments ws_i, the half-span locations s_i can be determined by:

$$s_i = \sum_{j=0}^{i} ws_i[j], \quad \text{for } i = 0, 1, 2, \ldots, n-1 \tag{3.6}$$

2. **Calculate numerical half-wing sub-divisions:**

   - Given the total number of wing subdivisions, N, the numerical wing can be calculated by:

$$w_{ni} = \frac{b}{2} cos(\Theta_i) \tag{3.7}$$

     where b the wing span, and Θ_i the i^{th} positional coordinate on the half span ranging form $[0, \frac{\pi}{2}]$.

3. **Calculate nearest match:**

   - Given the numerical wing subdivisions w_{ni} and the defined actual half-span locations s_i, the nearest match between the defined wing and the generated

wing can be calculated. It is important to note that the more wing subdivisions used, the closer the match between the numerical wing properties and the generated wing properties.

$$w_{mi} = min(abs(s_i - w_{ni})) \tag{3.8}$$

Here the indices of the locations of the nearest matches are also stored for later use.

With the nearest match indices known, and the root chords specified, Algorithm 1 can be used to determine the Mean Aerodynamic Chord (MAC) of each station.

Algorithm 1 Calculate the MACs along the wing span

Require: Idx: an array containing the indices where the numerical wing closely approximated the defined wing.

Require: $Chords$: an array defining the chord lengths of each station.

Require: N: the total number of numerical wing subdivisions.

Ensure: MAC: an array of modified chord lengths.

$MAC \leftarrow []$
for j **in** range(len($Chords$) - 1) **do**
    $MAC_i \leftarrow Chords[j]$
    for i **in** range(0, N) **do**
        if Idx$[j+1] \leq i <$ Idx$[j]$ **then**
            $MAC_i \leftarrow MAC_i - \frac{Chords[j]-Chords[j+1]}{\text{Idx}[j]-\text{Idx}[j+1]}$
            MAC.append(MAC_i)
        end if
    end for
end for
MAC.prepend($Chords$[0])
$MAC \leftarrow$ reverse(MAC)
return MAC

3.4.2 Verification of the 3D lift calculation sub-module

This section is concerned with the verification of the developed dynamic station LLT element of the TSM. To verify the accuracy of this sub-module for a given wing design, a set of free-stream speeds and lift coefficients are fixed for a diverse range of operating conditions and the relevant module outputs are compared with those of other simulation packages. The free-stream speeds and lift coefficients are chosen such that the verification is done over a flight envelope typically present in sailplane cross-country flights. For every lift coefficient and free-stream speed, the wing's induced drag as calculated by the TSM is compared to those of the XFLR5[1] model and an in-house model developed by the Jonker Sailplane (JS)[2] group referred to as the JSCode.

XFLR5 is an analysis tool for airfoils, wings and planes operating at low Reynolds numbers and includes linear LLT, nonlinear LLT, and 3D panel capabilities for estimating a wing's performance. Seeing that the TSM and JSCode make use of a linear LLT as opposed to the XFLR5-model and that the TSM employs techniques to guarantee XFOIL convergence near a given operating condition, the outputs of the 3 models are not expected to be identical. However, because the induced drag verification is done mostly in the linear region the results are expected to be within an acceptable error margin, i.e. less than 2%.

Having the same conditions in all three environments is necessary for verifying the dynamic station LLT element of the TSM with XFLR5 and the JSCode. That is, we need to specify exactly the same wing design parameters and execute the simulation by varying the operating conditions in the same way. For the verification process, the wing design parameters, as shown in Table 3.5, are used to set up the simulation environment in the TSM's simulation model, XFLR5-model and the JSCode. The simulations are run for free-stream speeds ranging from 25 m/s (90 km/h) to 68 m/s (244 km/h) with the wing lift varying between 0.18 to 1.34.

[1] XFLR5
[2] Jonker-Sailplanes

Table 3.5: Wing design parameters used in the 3D lift and 3D drag sub-module accuracy verification

Station	Span (m)	Chord (m)	Dihedral (deg)	Twist (deg)	airfoil
1	0	0.820	0	0	Opt-110
2	2.200	0.800	0	0	Opt-110
3	4.180	0.700	0	0	Opt-110
4	5.970	0.535	0	0	Opt-110
5	6.850	0.419	0	0	Opt-110
6	7.520	0.210	0	0	Opt-110

The results from this analysis can be seen in Table 3.6. From this table the following is observed:

1. For the case of the TSM induced drag results compared to those of the XFLR5 model, the Mean Absolute Percentage Error (MAPE) is 1.56%.

2. The largest errors for XFLR5 versus the TSM is for the operating points 55 m/s and 25 m/s with a value of 3.01% and 6.44% respectively. These are also the two points where the XFOIL calculations in the XFLR5 model struggled to converge and therefore led to slightly decreased induced drag values. Another cause of the discrepancies seen here is attributed to the fact that XFLR5 uses a nonlinear lifting line method to solve for the induced drag whereas the TSM implements a linear lifting line method (see Section 3.4.1).

3. Excluding the non-convergent cases for XFLR5, the TSM error rate results in a MAPE of 0.98%.

4. When comparing the TSM induced drag results with those of the JSCode, more favourable results are seen with a MAPE of 1.18%. These results are aligned with the XFLR5 results when excluding the non-converged cases.

5. The minor differences between the TSM and JSCode can be attributed to the different ways that XFOIL is utilised. In the JSCode, XFOIL is used to manually

calculate the 2D performance of a given airfoil section upfront and stored in a matrix. This matrix is later used in an interpolation routine to find the appropriate 2D performance coefficients. In the TSM XFOIL is called directly at the point of calculation with the appropriate simulation parameters and hence no interpolation is required.

Table 3.6: Verification results of the TSM's induced drag prediction for multiple operating conditions compared to the JSCode and XFLR5 model outputs. Cases highlighted with an * indicate the operating points where the XFOIL simulations did not/ or struggled to converge.

Operating Conditions		TSM	XFLR5		JSCode	
C_L (dim)	V (m/s)	C_{D_i} (dim)	C_{D_i} (dim)	MAPE	C_{D_i} (dim)	MAPE
0.180720	68.00	0.000467	0.000461	1.30	0.000461	1.30
0.276900	55.00	0.001096	0.001064	3.01*	0.001083	1.20
0.373000	47.00	0.001989	0.001987	0.10	0.001966	1.17
0.469300	42.00	0.003148	0.003147	0.02	0.003112	1.16
0.565000	39.00	0.004571	0.004546	0.54	0.004510	1.35
0.661700	36.00	0.006258	0.006185	1.18	0.006190	1.10
0.757900	33.00	0.008210	0.008076	1.65	0.008116	1.16
0.854140	31.00	0.010426	0.010234	1.88	0.010310	1.13
0.950350	30.00	0.012907	0.012737	1.34	0.012760	1.15
1.046500	28.00	0.015653	0.015506	0.94	0.015472	1.17
1.143000	27.00	0.018663	0.018516	0.79	0.018458	1.11
1.239000	26.00	0.021940	0.021717	1.03	0.021688	1.16
1.335000	25.00	0.025476	0.023934	6.44*	0.025180	1.18
Average				1.56	N/A	1.18

3.4.3 3D drag calculation

In this section, the details of the 3D drag calculation sub-module, as used in the TSM is discussed. Here specific attention is given to the profile drag calculations, winglet drag contributing assumptions, and how the CFD drag results for the fuselage/empennage configuration are used to arrive at the total 3D drag for the aircraft.

In this research, finding the total 3D drag of a given sailplane reduces to solving the

following equation:

$$C_D = C_{D_{profile}} + C_{D_{induced}} + C_{D_{f/t}} + C_{D_{winglet}} \tag{3.9}$$

where $C_{D_{profile}}$ the total wing profile drag coefficient, $C_{D_{induced}}$ the total wing induced drag coefficient, $C_{D_{f/t}}$ the total drag coefficient for the fuselage/empennage configuration, and $C_{D_{winglet}}$ the total drag coefficient for the winglet.

As mentioned earlier, the $C_{D_{f/t}}$-term for a range of operating conditions was calculated by Jonker Sailplanes via CFD simulations and made available for this research. The details of these simulations are out of the scope of this research and hence are not reproduced here.

For the $C_{D_{induced}}$ values, the dynamic station LLT is employed and simulation runs over a wide range of operating conditions are executed. This means the $C_{D_{induced}}$ values are known for multiple free-stream speeds and angles of attack.

For the winglet drag calculation i.e. $C_{D_{winglet}}$, it is assumed that this term is a function of the profile drag term and is calculated using the following equation:

$$C_{D_{winglet}} = (\alpha_{wt} + C_{D_{profile}} * \gamma_{wt}) \tag{3.10}$$

where α_{wt} represents a winglet contribution constant and γ_{wt} represents a profile drag scaling factor. These quantities vary depending on the design of the winglet. In this research the value of α_{wt} and γ_{wt} is chosen to be $1.05e^{-5}$ and 0.02 respectively. These values are chosen such that they approximated the winglet design of the JS-3 18 m sailplane.

The final drag term required to calculate the total sailplane drag coefficient is the $C_{D_{profile}}$. The calculation of this term is more involved than the previously discussed drag terms because it requires the integration of both XFOIL and the dynamic station LLT in multiple stages of the process. The calculation of the $C_{D_{profile}}$-term can be bucketed in two steps. Step one is the standard 2D airfoil performance analysis calculation whereas step two can be seen as the Inverse 2D airfoil performance analysis.

In the 2D airfoil performance analysis calculation, XFOIL is employed to calculate the zero-lift angle of attack and the lift-curve-slope of each airfoil present along the wing

span. From here these sectional properties, together with the wing design specifications, are used to calculate (1) the lift distribution along the wing span, and (2) the station-wise Reynolds numbers and Mach numbers. With the lift distribution along the wing span known, the average 3D lift generated per station is calculated.

This quantity is then used in the Inverse 2D airfoil performance analysis process step. In this step, XFOIL is invoked with the average 3D lift generated per station together with the local Reynolds and Mach numbers present at each station. This is done to calculate the sectional 3D profile drag.

To arrive at the total profile drag coefficient for the wing, the individual sectional 3D profile drag coefficients are then summed. Because the reference areas are different for each station this cannot be done directly. Hence, the coefficients are first converted to their dimensionalised representations. This means that the coefficients can now directly be summed. To then finally arrive at a profile drag coefficient for the wing, the total dimensionalised drag is converted back to its non-dimensionalised representation.

With all of the drag quantities accounted for and in both their dimensionalised and non-dimensionalised formats, Equation 3.9 is used to calculate the total sailplane 3D drag coefficient.

3.4.4 Verification of the 3D drag calculation sub-module

In this section, the verification of the 3D profile drag calculation as done by the TSM is discussed. Here, the total drag calculation is not verified directly for three reasons. First, the induced drag calculation has already been verified in Section 3.4.2. Secondly, the winglet drag term is merely a linearly weighted function of the profile drag. And thirdly, the fuselage/empennage configuration drag is externally simulated and verified with CFD software and implicitly assumed to be accurate. Therefore verifying only the main drag component of the drag calculation sub-module, i.e. the profile drag, means effectively verifying the accuracy of the entire 3D drag sub-module.

To verify this sub-module, the same simulation set-up as discussed in Section 3.4.2 is defined in the TSM, XFLR5, and the JSCode. Once again the simulations are run for free-stream speeds ranging from 25 m/s (90 km/h) to 68 m/s (244 km/h) with the wing lift varying between 0.18 dim to 1.34 dim to ensure a feasible range of sailplane cross-country operation is covered.

The results from this analysis can be seen in Table 3.7. From this table, the following is observed:

1. The coefficient of determination between the TSM and XFLR5 profile drag values are 0.9975. This is compared to a coefficient of determination of 0.9993 for the case of the TSM profile drag compared with the JSCode.
2. The MAPE for the TSM versus the XFLR5 is 8.77% compared to 4.78% for the case of TSM versus JSCode.
3. The discrepancy seen when comparing TSM and XFLR5 in the point 25 m/s can again be attributed to XFOIL convergence issues.

From this analysis, it is clear that the TSM results are in closer agreement with the outputs of JSCode. This is for three main reasons:

1. The first is that both the TSM and the JSCode makes use of a linear dynamic LLT model as opposed to the non-linear LLT method used by XFLR5.
2. The second reason is that the assumptions made in both the TSM and the JSCode are similar with only slight differences in how the winglet drag contribution term is treated.
3. Finally, the TSMs implementation and that of the JSCode do not have convergence issues due to the specific design and integration of the 2D performance analysis module. In the JSCode the 2D airfoil performance coefficients are manually computed and convergence issues are dealt with as and when they arise. In the TSM,

this process is automated. A simple framework is implemented whereby, if a non-convergent case is encountered, the Reynolds and Mach numbers as well as the panel density and angle of attack are iteratively adjusted upward or downward in very small increments, until convergence is met. This means that for cases where XFOIL struggles to converge, the lift and drag coefficients solved for by the JSCode and the TSM will differ slightly, especially for cases with high separation. However, these differences were on average below 5% and hence acceptable.

Table 3.7: Verification results of the TSM's profile drag prediction for multiple operating conditions compared to the JSCode and XFLR5 model outputs. Cases highlighted with an * indicate the operating points where the XFOIL simulations did not/ or struggled to converge.

Operating Conditions		TSM	XFLR5		JSCode	
C_L (dim)	V (m/s)	C_{D_p} (dim)	C_{D_p} (dim)	MAPE	C_{D_p} (dim)	MAPE
0.180720	68.00	0.003520	0.003540	0.56	0.003615	2.64
0.276900	55.00	0.003730	0.003940	5.33	0.003890	4.11
0.373000	47.00	0.004310	0.005010	13.97	0.004711	8.51
0.469300	42.00	0.006055	0.005853	3.45	0.005988	1.12
0.565000	39.00	0.007663	0.008454	9.36	0.008158	6.07
0.661700	36.00	0.008569	0.008815	2.79	0.008734	1.89
0.757900	33.00	0.009715	0.010924	11.07	0.010354	6.17
0.854140	31.00	0.010428	0.011766	11.37	0.011191	6.82
0.950350	30.00	0.012210	0.013263	7.94	0.012748	4.22
1.046500	28.00	0.013823	0.015494	10.78	0.014739	6.21
1.143000	27.00	0.015845	0.017484	9.37	0.016677	4.99
1.239000	26.00	0.018830	0.021283	11.53	0.020096	6.30
1.335000	25.00	0.021760	0.026067	16.52*	0.022462	3.13
Average				8.77	N/A	4.78

3.5 Overview of the fully integrated TSM

To form the final TSM[3], the airfoil generator module, the airfoil performance analysis module, the sailplane performance analysis module, and the cross-country performance

[3]This entire TSM is implemented in Python and follows the exact process flow depicted in Figure 3.5. In this implementation, each module is developed as a stand-alone class and the classes are then unified in a single piece of software to form the final TSM.

analysis model (derived in Section 2.2) are combined. The end result is a model capable of predicting the average cross-country performance of a given sailplane design for a defined cross-country flight task. The inference process of the final TSM is as depicted in Figure 3.5.

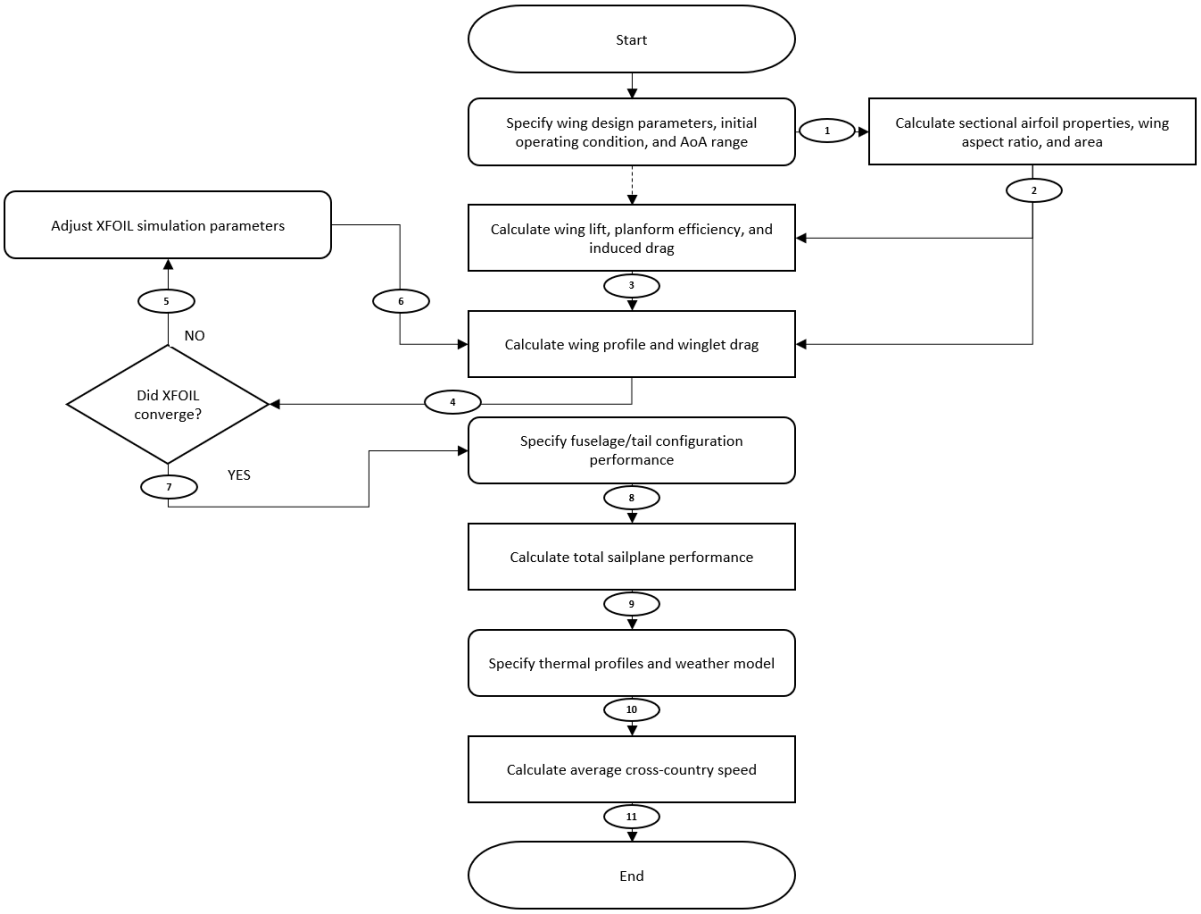


Figure 3.5: Process diagram of the entire TSM inference process to acquire an average cross-country performance estimation for a given sailplane design and cross-country flight task definition

The inference process starts with the specification of the airfoil profile at each wing station, the lengths and chords of each station, the AoA range of the flight envelope, and the initial operating conditions at which the sectional airfoil properties at each station should be calculated. It is important to note that the airfoils specified here are generated with the SPM – which as previously discussed is used as the airfoil generator module in the traditional framework.

From here, in step (1), the sectional properties of the airfoils present at each station are calculated with the airfoil performance analysis module i.e. XFOIL. Also here the wing aspect ratio and the wing area are calculated given the defined wing planform in terms of the station chords and lengths.

Then in step (2), the sectional properties of the airfoils present at each station alongside the wing geometry and AoA range are passed to the dynamic station LLT for the calculation of the wing induced drag, planform efficiency, total wing lift, and the lift distribution along the wing span.

In step (3), the sailplane's calculated lift-related characteristics alongside the wing geometry and AoA range are passed to the 3D drag calculation sub-module of the sailplane performance analysis module. This sub-module is then tasked with calculating the total wing profile drag and winglet drag in steps (4)-(7) where a solution is iteratively searched for if the airfoil performance analysis module does not converge. Once a solution is found for the winglet and profile drag, in steps (8)-(9), the fuselage/empennage performance specification is used to interpolate for the corresponding fuselage/empennage drag for every flight condition over the flight envelope.

Then, in steps (10)-(11) a thermal and weather model specification as defined in Section 2.2.5, and specifically Table 2.2, is used in the cross-country model derived in Section 2.2.2 to calculate the average cross-country speed.

In the following section, the accuracy of the TSM in terms of average cross-country flight speed prediction for a given sailplane design and weather model is verified. This verification is done both experimentally and by means of simulation.

3.6 TSM verification

As mentioned, verifying the accuracy of the final TSM is done in the form of comparing the model's output with both experimental and simulation results.

For the experimental results, the flight test data acquired by Idaflieg on the JS3-18m class sailplane[4] is used. Idaflieg, short for Interest Group of German Academic Aviation Groups eV, is the umbrella organisation for Academic Aviation Group (Akaflieg) in Germany. They are internationally recognised through the flight performance measurements carried out jointly with the Institute for Flight Guidance (IFF) and the German Aerospace Center. In the summer of 2020, Idaflieg carried out a series of tests on the JS3-18m class sailplane which they graciously made available to Jonker Sailplanes.

As for simulation results, the JSCode developed by the head aerodynamicist of Jonker Sailplanes, is again used.

Effectively comparing the accuracy of the TSM with the JSCode simulation and the Idaflieg experimental results means the same sailplane design configuration must be used in all three cases. However, seeing that the JS3-18m class sailplane design is proprietary, the details of this setup are not reproduced here.

For verification, a two-step process is followed. First, the sink speed curves of the TSM, the JSCode and the Idaflieg experimental measurements are compared. The accuracy of the TSM and the JSCode is then quantified by using the Idaflieg results as a baseline. Secondly, the overall cross-country performance calculation of the TSM is compared to the calculations of the JSCode.

The results of the respective sink speed curve comparisons can be seen in Figure 3.6. This figure shows that there is strong agreement between the Idaflieg sink speed measurements, the JSCode sink speed calculations, and the TSM's sink speed calculations in the range of 120 km/h to 200 km/h. From this analysis, it can also be observed that the TSM's calculated sink speed curve strongly correlates with the JSCode calculations and that the coefficient of determination is 0.994. For the case of the TSM's sink speed and the Idaflieg measurement, the coefficient of determination equates to 0.8851. Using the Idaflieg measured sink speed values as a reference, the MAPE between the Idaflieg measurements and the TSM calculations is 6.11%. For reference, the MAPE between the Idaflieg measurements and the JSCode calculations is 4.52%.

[4]The JS3 is a proprietary 18-meter sailplane designed and manufactured by Jonker Sailplanes.

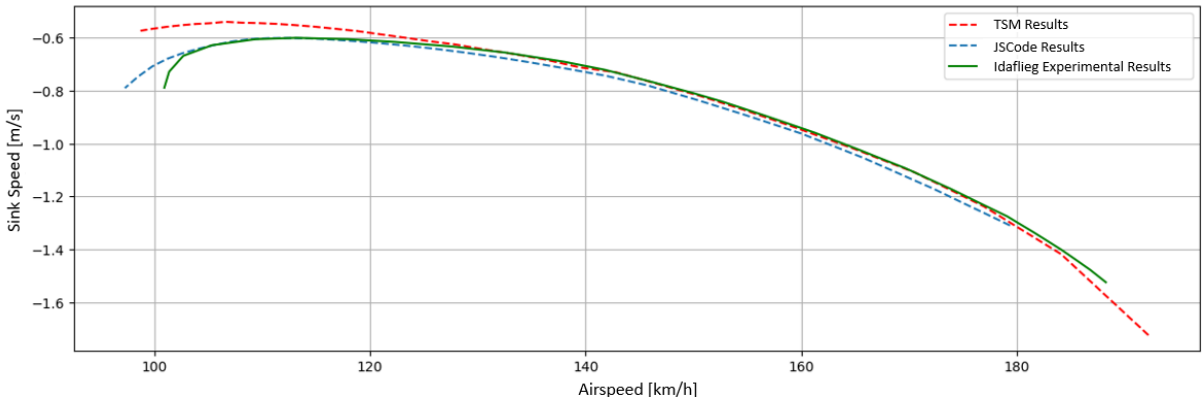


Figure 3.6: Sailplane predicted sink speed of the TSM compared to the simulation results of the JS-model and the experimental results of Idaflieg

For the second step in the verification process, the cross-country performance predicted by the TSM is compared to that of the JSCode. This verification takes place under the assumption that the sailplane is tasked to fly 300 km, with Quast's weather model governing the thermal types present. That is, in the cross-country simulation the sailplane is tasked with flying 24 km in an A1 type thermal, 126 km in an A2 type thermal, 24 km in a B1 type thermal, and 126 km in a B2 type thermal. This is the Quast weather model defined in Section 2.2.5. The results of the cross-country simulation of the JSCode and the TSM are depicted in Tables 3.8 - 3.9 respectively. From these results, it is seen that there is close agreement between the cross-country simulation results of both models with slight differences in the thermal-specific predicted maximum climbing speed. This difference is attributed to the different assumptions made for estimating the contribution of winglet drag – see Section 3.4.2. The JSCode calculates this quantity as a percentage of profile drag whereas the TSM calculates this quantity with Equation 3.10. This variation in winglet drag contribution also gives rise to the different sink speeds calculated between the JSCode and the TSM in the ranges of 100 km/h to 120 km/h – as seen in Figure 3.6. Although minor, these differences compound throughout the remaining sailplane cross-country drag calculations and result in a lower average cross-country speed predicted by the TSM, compared to the JSCode. The difference, however, is minor with the JSCode predicting an average cross-country speed of 94.29 km/h and the TSM predicting 92.54 km/h highlighting the accuracy of the TSM.

Table 3.8: JS-model cross-country simulation results for a 300 km task

JS-Model Cross-Country Simulation 300 Km Task					
Thermal model	Max climb [m/s]	Speed [km/h]	C_L	$C_{D_p} + C_{D_o}$	L/D
A1	0.343	128.544	0.681	0.008	54.561
A2	1.781	176.848	0.360	0.008	39.147
B1	0.767	147.759	0.516	0.007	51.067
B2	2.444	191.914	0.306	0.008	32.973
Quast's weather model					
Thermal model	Percentage Allocation	Distance[km]	Height required [m]	Time for climb [s]	Total time for phase [s]
A1	8%	24.000	439.872	1284.183	1956.326
A2	42%	126.000	3218.667	1807.717	4372.625
B1	8%	24.000	469.970	612.937	1197.673
B2	42%	126.000	3821.303	1563.814	3927.374
Average speed	94.29	km/h			

Table 3.9: TSM cross-country simulation results for a 300 km task

TSM Cross-Country Simulation 300 Km Task					
Thermal model	Max climb [m/s]	Speed [km/h]	C_L	$C_{D_p} + C_{D_o}$	L/D
A1	0.357	135.961	0.608	0.008	53.916
A2	2.037	191.694	0.306	0.009	30.273
B1	0.784	157.507	0.453	0.008	46.518
B2	2.489	198.998	0.284	0.010	26.732
Quast's weather model					
Thermal model	Percentage Allocation	Distance[km]	Height required [m]	Time for climb [s]	Total time for phase [s]
A1	8%	24.000	445.134	1246.233	1881.708
A2	42%	126.000	4162.175	2043.102	4409.373
B1	8%	24.000	515.931	658.444	1206.992
B2	42%	126.000	4713.506	1894.070	4173.492
Average speed	92.54	km/h			

3.7 Conclusion

In this chapter, the sub-components required to develop the TSM are specified, developed and individually verified. These sub-components are then integrated to form the final TSM. The accuracy of the TSM is verified both experimentally and by means of simulations. For the flight conditions investigated, the TSMs predicted sink speeds are in

agreement with the experimental results with a coefficient of determination of 0.8851. The MAPE between the TSM and the experimental results for the sink speed values is 6.11%. It is therefore concluded that the TSM sink speed values are accurate within roughly 94% of the Idaflieg experimental results and 95% of the JSCode simulation model. When comparing the average cross-country speed prediction by the TSM with the JSCode a difference of 1.75 km/h (1.86%) is observed.

For the flight conditions tested, which are strategically chosen to cover a wide range of typical sailplane cross-country operations, the TSM is shown to be accurate in sink speed prediction, average cross-country speed prediction, and 3D drag calculation. Although the results of the TSM are not perfectly aligned with the simulation models and experimental results, they are highly correlated with these outputs and are accurate within an acceptable margin. This means a result with increased performance as predicted by the TSM will translate to a design with increased performance as predicted by the JSCode or experimental results (under the verified operating conditions). This, therefore, means that the model is suitable for later use in the DSM definition as well as general multi-objective optimisation.

Chapter 4

Extracting parsimonious shape and structural features

4.1 Introduction

In this chapter, a framework is presented for extracting parsimonious airfoil shape and structural variables from their raw x and y coordinates representations. To ensure an accurate and efficient representation of these variables, we first discuss the necessary requirements in Section 4.2. Then, in Section 4.3, we identify and examine the relevant parsimonious variables in terms of an airfoil's shape and structural characteristics. Next, in Section 4.4, a process is developed for extracting these variables based on numerical calculations. Finally, in Section 4.5, comments are made on the efficacy of the developed parsimonious variable extraction framework.

4.2 Requirements

In this section, the requirements of the parsimonious variable extraction framework are discussed.

The main objective for the parsimonious variable extraction framework is to have the capability of calculating an airfoil's parsimonious variable representation from its raw x and y coordinate definition. The most important requirement for the framework is therefore to enable the extraction of Padulo (Padulo et al., 2009) satisfying variables (see Section 2.4.1).

At a high level, this means that the parsimonious variables should be complete, and thus able to describe any airfoil. The variables should be orthogonal, meaning that a unique set of parsimonious variables should only generate a single unique airfoil. The number of variables used should be limited and have intrinsic meaning. Finally, the method should not generate impractical airfoil shapes.

The second design requirement is one of computational efficiency. This is because numerous numerical procedures are used in this framework to calculate the identified variables and, seeing that these calculations are done implicitly in the data mining pipeline, efficiency is a key design requirement.

The final design requirement for this framework is accuracy. It is known that small perturbations to airfoil shape characteristics can lead to large variations in airfoil performance characteristics. Mitigating this issue requires the chosen parsimonious variables to establish a perfect one-to-one mapping between the low-order representation and the high-order representation (i.e., the raw x and y coordinates). Given an accurate model able to predict 2D performance from both of these data domains, this would thus translate into the same aerodynamic characteristics calculated irrespective of the input feature space used.

4.3 Overview of the selected parsimonious variables

In this section, an overview is given of the parsimonious shape and structural variables selected for use in this research. Currently, there is no available consensus on what the most important parsimonious airfoil characteristics are when designing airfoils. The

current, widely adopted airfoil parametrisation technique is the PARSEC (Sobieczky, 1999) method, as introduced in Section 2.4.1. Although this method gives 11 parsimonious airfoil shape features, it lacks providing information about:

1. The camber distribution.
2. Structural characteristics.
3. Maximum camber location and magnitude.
4. Magnitude and location of the maximum airfoil thickness.
5. Curvature and location of the maximum lower surface.

Masters et al. (Masters et al., 2017) determined that at least 25 design variables are needed to cover 80% of the design space regardless of the deformative or constructive airfoil parametrisation method used. This is significantly more than the 11 parameters of PARSEC or the number of control points typically used when optimising 2D airfoils with B-Spline or CST methods. In this research, a total of 23 parsimonious variables are chosen. This means that, if designed correctly, close to 80% of all airfoil design space will be covered, based on the findings of Masters et al. (Masters et al., 2017)[1]. In an attempt to maximise coverage, in these 23 parsimonious variables, the widely used PARSEC shape parameters, the mentioned features not covered by this method, and additional shape and structural features are numerically extracted from raw airfoil x and y coordinates. The framework is designed such that it can utilise the x and y coordinates produced with any constructive or deformative method.

Figure 4.1 shows the airfoil parsimonious shape parameters used in this research to describe an airfoil's profile. In Table 4.1, the detail of these 17 shape parsimonious variables can be seen.

[1]In Section 4.5.1 it is shown that the chosen 23 parsimonious variables are able to approximate all 1 621 airfoils in the UIUC database with an acceptable degree of accuracy (MAPE of 0.37). Although this does not guarantee full design space coverage, it indicates the versatility, accuracy, and flexibility of the proposed method in the approximation of a diverse airfoil population.

Table 4.1: Seventeen identified airfoil parsimonious shape parameters to be derived with the parsimonious variable extraction framework

Item	Number of features	Feature description
1	1	Nose radius of the leading edge
2	3	Magnitude of the minimum lower surface peak Location of the minimum lower surface peak Minimum lower surface peak curvature
3	1	Leading edge camber angle
4	2	Magnitude of the maximum airfoil thickness Location of the maximum airfoil thickness
5	3	Magnitude of the upper surface peak Location of the upper surface peak Upper surface peak curvature
6	2	Magnitude of the maximum airfoil camber Location of the maximum airfoil camber
7	3	Magnitude of the maximum lower surface peak Location of the maximum lower surface peak Maximum lower surface peak curvature
8	1	Trailing edge camber angle
9	1	Camber angle
Total	17	

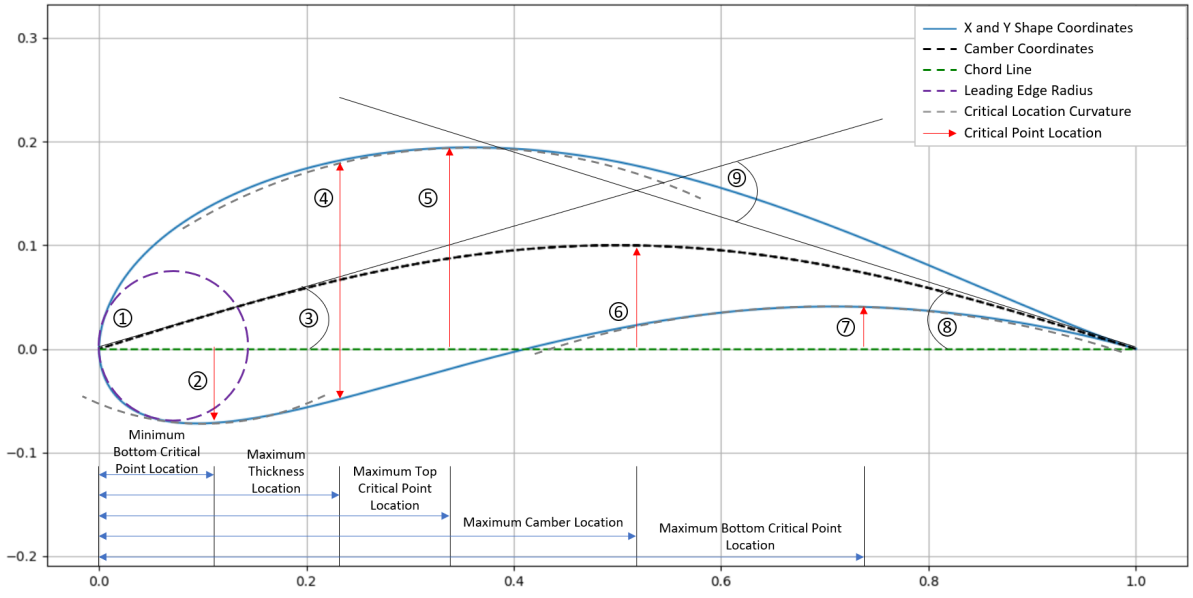


Figure 4.1: Depiction of the chosen airfoil parsimonious shape variables used to describe a diverse range of airfoil profiles

Apart from the 17 parsimonious shape variables mentioned, the profile of a given airfoil is also mapped to its corresponding structural characteristics. For this research, the following six structural properties are used in the parsimonious variable representation of an airfoil profile:

1. Airfoil area,
2. Airfoil perimeter,
3. Airfoil x centroid,
4. Airfoil y centroid,
5. Bending inertia about the y-axis, and
6. Bending inertia about the x-axis.

These additional structural variables are included, as this allows the extension of the final system's flexibility in terms of intuitive and explicit constraint setting. For example, if an airfoil with a low area is required, then upfront all airfoils generated that do not adhere to the area-defined constraint are excluded from the optimisation procedure.

4.4 Parsimonious variable calculations

In this section, the details are provided for extracting each of the aforementioned shape and structural parsimonious variables for any given airfoil profile defined in terms of its x and y coordinates.

4.4.1 LE nose radius and critical airfoil points' curvature

As seen in Figure 4.1, the leading edge is the front point of the airfoil that has the maximum curvature, which corresponds to a minimum radius. This minimum radius

of the leading edge is known as the leading edge (nose) radius. As for the critical point curvature, it is defined as the maximum locations on the top and bottom surfaces, in terms of y coordinate magnitude. The magnitudes of the nose radius and respective critical point curvatures are derived numerically given a set of x and y coordinates describing the airfoil profile. For the calculation of the nose radius (r_{Le}), a parametric description of the nose circle is first required and is defined as:

$$r_{Le}^2 = (x-a)^2 + (y-b)^2 \tag{4.1}$$

with a the x location of the nose circle centre, b the y location of the nose circle centre, and x and y arbitrary coordinate pairs on the airfoil's nose circle circumference. In order to fit the nose circle numerically to the points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ that lie on the airfoil's nose circle circumference, Equation 4.2 is used as a measure of goodness of the fit.

$$SS(a, b, r_{Le}) = \sum_{i=1}^{n} \left(r_{Le} - \sqrt{(x_i - a)^2 + (y_i - b)^2} \right)^2 \tag{4.2}$$

The parameters a, b and r_{Le} are chosen in such a way that Equation 4.2 is minimised. Differentiation of SS yields:

$$\frac{\partial SS}{\partial r_{Le}} = -2 \sum_{i=1}^{n} \sqrt{(x_i - a)^2 + (y_i - b)^2} + 2nr_{Le} \tag{4.3}$$

$$\frac{\partial SS}{\partial a} = 2r_{Le} \sum_{i=1}^{n} \frac{x_i - a}{\sqrt{(x_i - a)^2 + (y_i - b)^2}} - 2(n\widetilde{x} - na) \tag{4.4}$$

$$\frac{\partial SS}{\partial b} = 2r_{Le} \sum_{i=1}^{n} \frac{y_i - b}{\sqrt{(x_i - a)^2 + (y_i - b)^2}} - 2(n\widetilde{y} - nb) \tag{4.5}$$

The parameters a, b and r_{Le} are numerically solved by simultaneously equating these partials to zero.

When calculating critical curvatures for the top, bottom-lower, and bottom-upper positions, the same procedure is followed as discussed for the case of the nose radius calculation. However, here the inverse of the radius for each respective case is calculated to

arrive at the respective curvature values. For example, the lower-bottom critical point curvature is mathematically expressed as:

$$y_{x_{LB}} = \frac{1}{r_{x_{LB}}} \tag{4.6}$$

With $r_{x_{LB}}$ as the radius of the lower-bottom critical point and calculated by the same algorithm used in the nose radius calculation.

4.4.2 Top, bottom-lower, and bottom-upper critical points

With the magnitudes of the curvature of the top, bottom-lower, and bottom-upper critical points known, the next step is to find their respective locations on the x-axis along the chord of the airfoil. For this calculation, the x and y coordinates are separated into pairs such that the upper and lower surfaces are individually defined. For the lower section of the airfoil, we therefore have the coordinate pairs x_L, y_L, which means the calculation of the magnitude of the bottom surface minimum critical point is reduced to:

$$y_{LB} = min(y_L) \tag{4.7}$$

The location of the bottom-lower critical point, x_{LB}, is the corresponding lower surface curve x-coordinate where the minimum point y_{LB} is located. Similarly, the bottom surface maximum critical point is calculated with:

$$y_{LT} = max(y_L) \tag{4.8}$$

Again, the location of the maximum bottom critical point x_{LT} is found with an index matching procedure where the x coordinate of the bottom surface is matched with the calculated maximum of the bottom curve. For the case of the upper curve critical points, Equation 4.8 is used, but instead of using the bottom curve description y_L, the description of the top airfoil curve, y_{up}, is used.

4.4.3 Airfoil critical camber parameters

Airfoil camber refers to the curve of the upper and lower surfaces of an airfoil. In other words, it is the amount of curvature that is present on the top and bottom surfaces of an airfoil. Camber is an important factor in the design of an airfoil, as it can significantly affect the lift and drag characteristics of the airfoil. Positive camber, which is when the upper surface of the airfoil is more curved than the lower surface, can increase the lift generated by the airfoil and vice versa.

The critical camber parameters include the location and magnitude of the maximum airfoil camber defined as x_{cm} and y_{cm} respectively, and the leading and trailing camber angles are defined as α_{Te}, and α_{Le} respectively. For the calculation of these parameters, it is assumed that the camber curve described with x_c, y_c can be obtained. Similar to Section 4.4.2, the maximum camber magnitude can be calculated as follows:

$$y_{cm} = max(y_c) \tag{4.9}$$

Once again the location of the maximum camber x_{cm}, is calculated by finding the x coordinate on the camber curve that corresponds to the maximum camber magnitude y_{cm}. For the calculation of the leading and trailing edge camber angles, it is assumed that for small variations in x_c, the relative angles stayed approximately the same. This assumption is valid seeing that the x and y coordinates at the leading and trailing edges are more densely clustered together in comparison to the more sparse coordinate pair allocation for the rest of the airfoil shape. This means that, for the first n-leading edge and n-trailing edge locations, the relative camber angles are constant. The leading and trailing edge camber angles can therefore be calculated by the following two equations respectively:

$$\alpha_{Le} = \frac{1}{n}\sum_{i=1}^{n} \arctan(\frac{C_{M_i}}{x_i})\frac{180}{\pi} \tag{4.10}$$

$$\alpha_{Te} = \frac{1}{n}\sum_{i=1}^{n} \arctan(\frac{C_{M_i}}{1-x_i})\frac{180}{\pi} \tag{4.11}$$

4.4.4 Parsimonious airfoil structure characteristics

The first structural characteristic calculation to be discussed is that of the airfoil area. An airfoil's area can be calculated by evaluating Equation 4.12, given that we have an analytical expression for the upper and lower airfoil curves, $z_{Up}(x)$ and $z_{Lo}(x)$ respectively, and the chord varies C from zero to 1.

$$A_{airfoil} = \int_{C=0}^{C=1} [z_{Up}(x) - z_{Lo}(x)]dx \tag{4.12}$$

The next structural characteristic calculation to be discussed is the airfoil perimeter. The airfoil perimeter represents the total distance of the continuous line forming the boundary of a closed airfoil and can be calculated by Equation 4.13.

$$S_{airfoil} = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{4.13}$$

Finally, the two bending moments about the respective x-x and y-y axes are up for discussion. To calculate the bending inertia about the y-axis, I_{xx}, the centroid of the y-axis is first required, y_c. This parameter is calculated as follows:

$$y_c = \frac{1}{A_{airfoil}} \int_{C=0}^{C=1} \frac{1}{2}[z_{Up}(x)^2 - z_{Lo}(x)^2]dx \tag{4.14}$$

The bending inertia about the y-axis is then calculated by means of the following equation:

$$I_{xx} = \int_{C=0}^{C=1} \frac{1}{3}[(z_{Up}(x) - y_c)^3 - (z_{Lo}(x) - y_c)^3]dx \tag{4.15}$$

For the calculation of the bending inertia about the x-axis, I_{yy}, a different approach is taken. Firstly, the x-centroid x_c is calculated by evaluating the following expression:

$$x_c = \frac{\sum_{i=1}^{n} A_i x_i}{A_{airfoil}} \tag{4.16}$$

where A_i is the airfoil area at slice i and is defined as:

$$A_i = |(x_i - x_{i+1})(y_i - y_{i+1})| \tag{4.17}$$

and $A_{airfoil}$ is determined with Equation 4.12. The airfoil bending inertia about the x-axis is then found by solving Equation 4.18.

$$I_{yy} = \sum_{i=1}^{n} I'_{x;i} + A_i(y_c - y_i)^2 \tag{4.18}$$

with the contribution of the term $I'_{x;i}$ being negligible.

4.5 Method verification

In this section, the proposed parsimonious variable extraction framework is evaluated in terms of its accuracy and its adherence to the criteria set out in Section 4.2.

4.5.1 Accuracy and efficiency verification

To evaluate the accuracy and efficiency of the parsimonious variable extraction framework it is used to calculate the 23 parsimonious variables of all 1 621 airfoils in the UIUC airfoil database. The magnitudes of the calculated parsimonious variables are then compared to the values generated with the XFOIL-GDES and XFOIL-BEND sub-routines. Seeing that XFOIL does not calculate the shape and structural variables of all 23 chosen parsimonious variables, the 10 features that XFOIL does calculate are used for comparison. Specifically, these 10 features are:

1. Maximum camber magnitude,
2. Maximum camber location,
3. Thickness magnitude,
4. Maximum thickness location,
5. Nose radius,
6. Total area,
7. Y-centroid,
8. X-centroid,
9. Bending inertia-y, and
10. Bending inertia-x.

The process for the parsimonious variable calculations with the proposed framework is as follows. First, an airfoil from the UIUC airfoil database is selected. Seeing that some of these airfoils only have a few coordinates ($\leq$ 35 in particular instances), the airfoil x and y coordinates are first cleaned. This is done by first splitting the top x and y coordinates. From here an 8th-order polynomial is used to fit the top and bottom surface curves, respectively. An 8th-order polynomial is chosen because it is found to be able to fit the respective curves with a less than 1×10^{-8} error. From here the respective top and bottom polynomials are used to generate an airfoil with the required density. In this case, 80 x and 80 y coordinate points are used. The airfoil file is then saved for later use in XFOIL. Also here, the coordinates are passed to the parsimonious variable extraction framework and all relevant features are calculated and stored.

For the XFOIL calculation, the process is simpler seeing that all the airfoil pre-processing has been done. Here all that is done is the cleaned airfoil file for every UIUC airfoil is iteratively loaded into XFOIL. From here the 10 relevant parameters are calculated and stored for comparison.

With the respective values for the 10 parsimonious features calculated with XFOIL and the proposed framework, the MAPE values can be calculated. This is done by using Equation 6.1, with the XFOIL values as a baseline. In Figure 4.2 the MAPE distributions for the 10 features can be seen for all 1 621 airfoils of the UIUC database.

From this figure and analysis, the following is concluded:

1. The average MAPE across all 10 features is significantly low with a value of 0.37.
2. The most accurate calculation is for the x-centroid and here the maximum error is 0.28%.
3. The most inaccurate feature calculation is for the nose radius and here a maximum error of 3.12% is observed. This is likely due to the difference in the calculation method applied between XFOIL and the proposed framework.
4. The average standard deviation MAPE across all cases investigated is 0.34 attesting

to the robustness of the numerical procedures employed.

In terms of numerical efficiency, when including the airfoil cleaning steps, the proposed framework is able to process an airfoil every 0.78 seconds – which is insignificant given the number of numerical iterations, integrals, and matrix algebra that is involved in each instance.

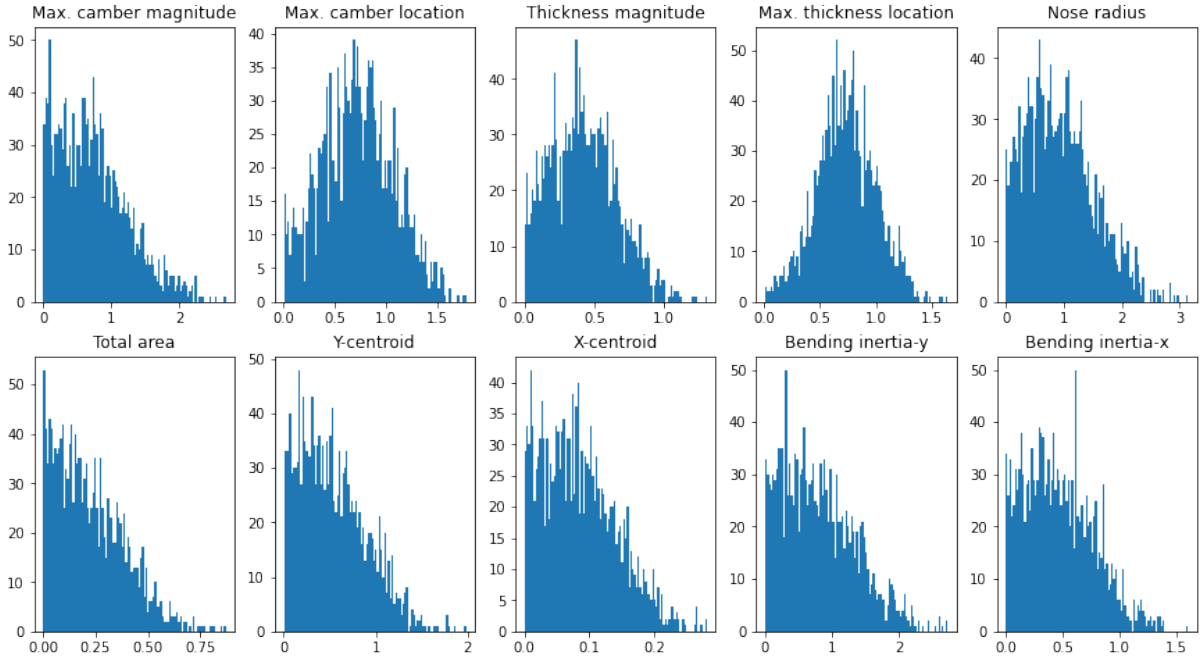


Figure 4.2: MAPE distribution calculated between proposed framework and XFOIL for all UIUC airfoils

4.5.2 Padulo completeness

In the verification of the Padulo completeness (discussed in Section 2.4.1) of the framework, four requirements are evaluated, i.e. completeness, orthogonality, parsimonious, and practicality.

For the completeness requirement i.e. the ability of the framework to represent any airfoil, the results of the previous analysis are commented on. Here it is seen that, for the *fitting* features of interest, the framework is able to approximate all 1 621 airfoils of the UIUC

airfoil database with less than 0.38% error. Also, the framework is developed to include 23 variables, close to the number Masters et al. found to be required for 80% coverage of all airfoils. Although not perfect, the method is shown to be able to approximate a diverse population of airfoil samples. This, therefore, highlights the framework's adherence to the completeness requirement.

For orthogonality, i.e. the ability to represent a unique airfoil for a unique set of inputs, the proposed framework does this by design. In the parsimonious variable framework, a set of 23 unique variables maps to a unique airfoil in the higher-order raw x and y airfoil coordinate space. If only one of the 23 parameters is slightly changed, this changes the higher-order distribution as well. By design, the proposed framework, therefore, is orthogonal. The same can be said for the parsimonious requirement. Each parsimonious variable was chosen so that it has intuitive and intrinsic meaning and that the perturbation of the feature has a predictable influence on the airfoil profile. Thus by design, the framework also adheres to the parsimony requirement.

The final requirement is that of practicality. That is, can the framework only represent practical airfoil shapes. Here the proposed framework fails, seeing that impractical airfoil shapes will also have calculable nose radius, bending inertia, maximum thickness, and other characteristics. This means that the proposed framework will also be able to represent and ultimately generate impractical airfoil shapes. However, ways to circumvent this limitation of the proposed framework are discussed in Section 7.7.4.

4.6 Conclusion

In this chapter, 23 variables are identified that are used as the parsimonious lower-order representation of an airfoil profile. Of these 23 variables, 17 are used to characterise the shape of an airfoil and 6 are used to characterise the structure. These variables are strategically chosen to not only allow for an intuitive description of an airfoil profile but also to allow for explicit control over the global and local structure.

Later, it will be shown how these 23 parsimonious variables form the foundation of the data mining (Section 5.2-5.3) and airfoil analysis (Section 6.2-6.5) objectives.

Finally, in Sections 9.4 and 10.3, it is shown how these strategically defined variables are used in the end-to-end cross-country optimisation objectives to facilitate intuitive constraint setting on the airfoils as input to the optimisation process.

Chapter 5

Data mining

5.1 Introduction

In this chapter, the process of mining 2D airfoil performance data for use in the development of the PDLM (Section 6.8.1), and hence the IPDLM (Section 6.8.2), is discussed. This dataset is referred to as the 2D performance dataset.[1] In Section 5.2, an overview of the 2D performance dataset is given with a focus on the eight response variables. Then in Section 5.3 the process of acquiring this 2D performance dataset is discussed. Here, attention is given to the strategies employed to ensure a diverse population of airfoil profiles and operating conditions are present in the mined data set. In Sections 5.4-5.5 the final 2D performance dataset is developed and analysed in terms of its statistical properties, integrity, and overall schema. Finally, the partitioning of the final 2D performance dataset into train, validation, and testing sets is discussed in Section 5.6, where attention is given to ensure that all three partitions share the same statistical distribution.

[1]The 2D performance dataset mined and analysed in this Chapter, differs slightly from the dataset discussed in Section 7.2. In Section 7.2, the airfoil shape dataset is used in the development of the airfoil generator module and the airfoil shape conversion module. This airfoil shape dataset, therefore, has no 2D airfoil performance data, nor any operating condition data. For the case of the 2D performance dataset, 2D airfoil performance data as well as operating conditions are present.

5.2 Data mining overview

The PDLM is responsible for accurately modelling the 2D aerodynamic performance of an airfoil, using a lower-dimensional parsimonious variable description and a set operating condition. This module is also used to derive the IPDLM, as discussed in Section 6.8.2, which is used to effectively replace the full functioning of XFOIL in the TSM (see Section 3.4.3).

As discussed in Section 2.6, using DL as a surrogate model for 2D airfoil performance analysis enables mapping of both conventional and non-conventional 2D performance characteristics. This mapping, however, requires a rich corpus of parsimonious airfoil profile characteristics, operating conditions, and conventional and non-conventional 2D airfoil performance characteristics. Such a 2D performance dataset was not publicly available at the time of writing this thesis. A custom data mining pipeline is therefore developed to mine these parsimonious variables (as defined in Section 4.4) and performance coefficients.

The conventional and non-conventional 2D airfoil performance characteristics mined and hence used in this research, can be seen in Table 5.1. In this table, a list of eight response variables, their motivation, and possible use in the DSM optimisation model is described.

Table 5.1: Eight response variables, their explanations, and use cases as required by the PDLM. Response variables with * indicate the non-conventional 2D airfoil performance characteristics.

Response variable	Explanation	Use case
Lift coefficient	Represents lift generated by an airfoil for a given operating condition	Used in the sailplane analysis model
Drag coefficient	Quantifies the resistance an airfoil experiences within a fluid environment	Used in the sailplane analysis model
Moment coefficient	The resultant torque or pitching moment produced as a result of the forces acting on the airfoil	Can be used to optimise for a minimum pitching moment
Top transition	The top surface x location of laminar to turbulent fluid flow transition	Can be used to restrict the sectional airfoils to only laminar flow airfoils
Bottom transition	The bottom surface x location of laminar to turbulent fluid flow transition	
Slope*	The gradient in the linear region of the Cl vs AoA curve	Used in the LLT model
Zero-lift AoA*	The angle at which the airfoil produces no lift	Used in the LLT model
Maximum lift coefficient*	The maximum Cl over all AoA for a given Reynolds and Mach number	Can be used to restrict the lower or upper limit of the maximum Cl for every station along the wing span

5.3 Data mining approach

With the 17 shape and six structural parsimonious variables used to describe any given airfoil defined alongside the 2D airfoil performance characteristics, it is possible to implement a data mining process. A process map depicting the data mining framework to create the 2D performance and airfoil shape datasets is shown in Figure 5.1. This data mining process has a total of nine steps. At a high level, the airfoil data mining pipeline makes use of an existing constructive airfoil generator technique (the SPM) to generate

airfoil profiles in terms of x and y coordinates. From here the framework developed in Section 4.4 is employed to extract intuitive parsimonious features from this x and y coordinate space. Throughout the data mining process, airfoil validity and airfoil geometry checks are set up with the purpose of (1) as far as possible, filtering out impractical airfoil shapes up front, and (2) generating airfoils with feasible sailplane profiles. Next, the identified performance coefficients are calculated using an XFOIL simulation. Finally, the disparate data sources are aggregated and stored for use during later model building.

A more detailed description of each step in the data mining process is given in the following sections.

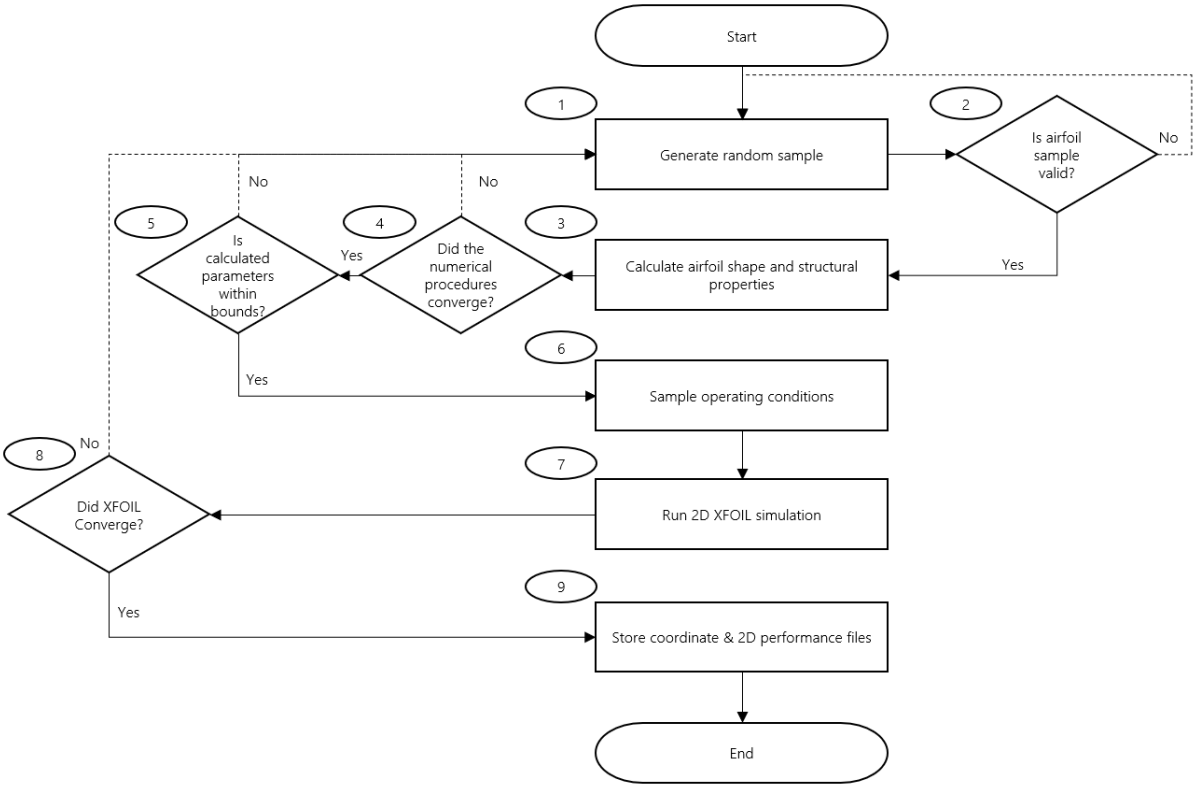


Figure 5.1: Airfoil data mining process map used to acquire both the 2D performance and airfoil shape datasets. For the 2D performance dataset all nine steps are executed whereas only steps one to five are required for the airfoil shape dataset.

5.3.1 Process step 1: Airfoil generation

The first step of the data mining process is concerned with generating a random airfoil sample from which its raw x and y coordinates are extracted. As mentioned in Section 3.2, in this research, the SPM (Ziemkiewicz, 2017) technique is used to generate the required airfoil profiles – although any constructive or deformative method can be used for this purpose. Generating airfoil profiles with adequate variability in the mined population involves randomly perturbing the input space of the SPM.

This perturbation is a two-step process. First, a combination of the six SPM parameters is uniformly sampled within the bounds defined in Table 5.2.[2] Next, these sampled coefficients are given as inputs to the SPM and the resulting airfoil profile is generated. It is important to note that the bounds defined in Table 5.2 are strategically chosen to maximise the variability of airfoil profiles generated. Although this occurred infrequently, these bounds resulted in the generation of impractical airfoil shapes. Additional steps are therefore implemented to filter out these impractical airfoil shapes, as discussed in Sections 5.3.2 and 5.4.

Table 5.2: Upper and lower SPM bounds used in the airfoil generation phase by uniformly sampling the six coefficients within the defined bounds

SPM parameter	Lower bound	Upper bound
B	1.800	2.200
T	0.010	0.700
P	0.100	8.000
E	0.010	8.000
C	0.000	0.350
R	-0.035	0.035

Seeing that the bounds of Table 5.2 are very wide, diverse airfoil profiles will result. This airfoil population may or may not include a large set of workable airfoils for standard class sailplane applications. Given enough high quality data, the downstream models trained on this data set will have adequate performance for general optimisation, such as seen in

[2]These upper and lower bounds were iteratively determined by sequentially varying the six parameter values to determine where the threshold was for generating impractical airfoil shapes.

Section 9.4. However, given that the data set is not tailored to only airfoils applicable for standard class sailplane operation, optimisation of SOTA gliders may not be possible. This discrepancy is however easily addressed, as seen in Section 10.3, where the data mining process is refined to only include airfoils from the desired distribution, thereby maximising downstream model accuracy potential over the operating ranges of interest. It is however still of importance to have a diverse data set to identify downstream model discrepancies and individual module shortcomings before developing a specialist or refined standard class sailplane data set.

5.3.2 Process step 2: Generated airfoil geometry validity check

In step two, the airfoil's raw x and y coordinates are evaluated for geometric validity. This is accomplished by analysing the airfoil leading edge nose radius for extreme curvature. A highly curved nose radius indicates that there is excessive sharp geometry present at the leading edge. Although this might be desirable for aircraft travelling at supersonic speeds, this is not where sailplanes operate during cross-country flight. Therefore, airfoil samples with highly curved nose radii are excluded from the data mining process. Specifically, this is achieved by filtering out any airfoils that have a maximum thickness percentage of below 7%, with a maximum camber below 1.2%, and a total thickness that is more than three times the nose radius. These characteristics were chosen because airfoils with these properties are most likely to be supersonic airfoils or to have impractical shapes due to excessive camber-to-thickness ratios.

5.3.3 Process step 3-5: Airfoil structural and shape parameter calculations

In step three, the parsimonious airfoil shape and the structural parameters are determined numerically with the methods discussed in Section 4.4. During the next step, step four, the numerical procedures used to calculate the parsimonious shape and structural properties are verified to ensure that they converged within the specified tolerance. If this is not

the case, the airfoil parsimonious variables solved for are considered to be inaccurate and hence not fit for further use in the model-building phase. Hence all airfoils with numerical solutions that did not converge are excluded from the data mining pipeline.

If the numerical calculations did converge, the airfoil passes through another checking phase (in step five) to ensure that the generated airfoil has the desired geometric properties, i.e. that it has the required minimum and maximum thickness, camber, curvature, leading-edge radius, and other properties. The bounds set here are purposely chosen to be wide and therefore encompass airfoils that would not typically be used in sailplane applications. This is again done to maximise the diversity of airfoils in the final dataset population. Table 5.3 shows the chosen parsimonious variable upper and lower bounds used in step five to filter out extremely thick and thin airfoil shapes along with airfoils with extreme camber or large nose radii.

Later, in Section 5.4, it will be shown how the design choices made in the data mining process as described in Tables 5.2 and 5.3 lead to airfoils with the desired shape characteristics.

Table 5.3: Upper and lower airfoil geometric constraints used to filter out airfoils with extreme thickness, camber, or large nose radii

Geometric feature	Lower bound	Upper bound
Camber	0.0000	0.400
Total thickness	0.0400	0.700
Top thickness	0.0300	0.700
Bottom lower thickness	-0.4000	-0.005
Bottom upper thickness	0.0000	0.350
Nose radius	0.0035	1.000

5.3.4 Process step 6-8: Performance data calculation

Once the airfoil shapes are generated, parametrised, and have passed the filtering criteria, it is time for the calculation of the 2D performance of the airfoil. For this, both the airfoil geometry file, containing the airfoil profile defined as raw x and y coordinates, and the operating conditions are required.

The operating conditions are randomly sampled from a uniform distribution (step 6) to ensure a diverse operating condition population is realised. Once again these upper and lower bounds are strategically chosen to ensure full coverage of the flight conditions present in typical sailplane cross-country flights. The upper and lower sample space bounds for the operating conditions are as follows:

$$250000 \leq Reynolds \leq 18000000 \tag{5.1}$$

$$0 \leq Mach \leq 0.3 \tag{5.2}$$

$$-10^{\circ} \leq AoA \leq 20^{\circ} \tag{5.3}$$

Once the operating conditions and the raw x and y coordinates, of dimension (160,1), are given to XFOIL, the simulation starts (step 7). Once the simulation ends, i.e. the total number of iterations is reached, XFOIL's convergence is validated. If XFOIL did not converge, the sample is disregarded and the process restarted.

5.3.5 Process step 9: Data generation and schema

Step nine is the final step in the data mining process and is concerned with the accumulation, merging, structuring, and storing of all the relevant data.

After running all nine steps for the generation of the 2D performance dataset, a data corpus with the schema of Table 5.4 is delivered:

Table 5.4: Data schema for the 2D performance dataset after running steps one to nine in the data mining process

Variable	Input or response variable	Dimensionality
Parsimonious variables	Input variable	(23) 17 shape variable 6 structural variables
Operating conditions	Input variable	(3) Reynolds number Mach number Angle of attack
2D performance coefficients	Response variable	(8) See Table 5.1

Similarly, after running only steps one to five, and then step nine, for the generation of the airfoil shape dataset, the schema of Table 5.5 results:

Table 5.5: Data schema for the airfoil shape dataset after running steps one to five, and then step nine, in the data mining process

Variable	Input or response variable	Dimensionality
Parsimonious variables	Input and response variable	(23) 17 shape variable 6 structural variables
SPM coefficient	Input variable	(6) B, T, P, E, C, and R coefficients
x and y coordinates	Response variable	(160) 80 x and 80 y coordinates

5.4 2D performance dataset analysis

In this section, the 2D performance dataset is generated, cleaned, statistically analysed, and finally partitioned for use in the PDLM development.

5.4.1 Intermediate dataset generation

The 2D performance dataset is developed in two sequential steps. First, an intermediate dataset is generated by means of employing the data mining pipeline discussed in Section 5.3. Next, this dataset is analysed and cleaned to remove outliers, ill-converged simulations, and noise, to form the final 2D performance dataset.

For the intermediate dataset generation, the full data mining pipeline is run for approximately 7 million cases distributed across four AWS EC2 instances. Each instance has a minimum of 16 gigabytes of RAM, between 6 to 12 cores, and a base processing speed between 3.0 and 4.0 GHz. In total, the data mining takes approximately 15 820 CPU hours.

In this generation process, the mentioned filtering steps are applied – resulting in the data mining pipeline having a 30% efficacy, meaning that approximately 4.9 million samples are either impractical airfoil shapes, do not have convergent parsimonious calculations, do not fall within the set constraints, or do not yield converged XFOIL simulations. This means that 2.1 million samples passed all filtering stages and yielded converged simulations, and hence are part of the intermediate dataset.

The intermediate dataset, therefore, had the schema of Table 5.4 with 2.1 million samples, 115 operating conditions per airfoil and hence 18 261 unique airfoils.

5.4.2 Intermediate dataset cleaning

The descriptive statistics of the intermediate dataset, along with a breakdown of the features per category are depicted in Table C.1. From these descriptive statistics, it is observed that there are still outliers present, specifically for the lift-curve-slope feature which has a minimum value of -50.1 and a maximum value of 18.5. This indicates that, even though there are mitigation steps built into the data mining pipeline, impractical airfoils and ill-converged XFOIL simulation records still make their way into the intermediate dataset.

The next step, therefore, is to ensure these impractical shapes and ill-converged simulation records are removed. This is firstly done by guaranteeing the lift-curve-slope feature values are in the feasible range as predicted by thin airfoil theory, i.e. approximately 2π (Houghton et al., 2016). This is done by excluding lift-curve-slopes lower than 5.5 1/rad and higher than 8 1/rad. In order to account for factors such as airfoil thickness and Reynolds numbers, which affect the lift-curve-slope feature, we set the bounds for this feature between 5.5 and 8 1/rad, rather than exactly at the theoretically determined value of 2π 1/rad for the inviscid domain. This is because we are working in the viscous domain, where these factors cause deviations from the inviscid theoretical values, as explained in (Houghton et al., 2016).

The next and final cleaning step is to remove ill-converged XFOIL simulations. This is achieved by removing the airfoil simulation which has disjointed C_l vs AoA and C_d vs AoA curves. The process of doing this is as follows: (1) monitor the rate of change between successive C_l and C_d values, and (2) remove airfoil records where this rate of change oscillated between positive and negative, indicating saw-tooth-like curves, therefore alluding to ill-convergent XFOIL simulations.

5.4.3 Final 2D performance dataset

The distribution of each input and response variable for the final dataset is depicted in Figures B.1 - B.5. From these distribution plots the following observations are made regarding the final cleaned dataset:

1. For the operating conditions, all feature distributions indicate that the values are correctly sampled from a uniform distribution within the defined boundaries.

2. From the Reynolds number distribution plot, it is noted that the ranges between 250 000 and 1 000 000 are underrepresented which indicates that XFOIL tends to converge less often for these lower Reynolds numbers.

3. From the maximum thickness magnitude and maximum thickness location feature

distributions it is seen that most of the airfoils have a thickness of $0.05 < y < 0.15$ situated between $0.075 < x < 0.25$ which is in the range required to ensure feasible sailplane airfoils were covered. This finding therefore implicitly verifies the decisions made in the data generation and data filtering stages as it results in an airfoil population with desirable properties.

4. The maximum C_l is centred between 1.5 and 2, which is again in the desired range for sailplane applications.

5. The C_d feature does not take on any negative values; the transition locations vary between 0 and 1, and lift-curve-slope values are between the filtered bounds. This further verifies the integrity of the final dataset and hence the data mining framework.

After all the cleaning steps have been applied, the final 2D performance dataset has 6 770 unique airfoils with the schema of Table 5.4 and a total of 78 550 samples.

5.5 Correlation analysis

In this section, the final 2D performance dataset is statistically analysed in terms of the correlation between the input and response variables. This is done mainly for three reasons: (1) to determine the strength and direction of the relationship, (2) to further verify the data integrity, and (3) to identify potential multicollinearity. Understanding these three characteristics allows for a deeper understanding of the data structure and hence facilitates the respective model-building processes.

For the correlation analysis, the Kendall-τ (Kendall, 1938) correlation coefficient is used. The Kendall-τ correlation coefficient measures the strength and direction of the monotonous relationship between a given set of continuous numerical variables.

This coefficient is defined to take a value between -1 and 1, with 1 indicating a perfect positive monotonous relationship, and -1 indicating a perfect negative monotonous rela-

tionship. Observing the p-values for each calculation is crucial to determine whether the correlation results from our sample population correlation calculations are significant. In this case, the p-values for the correlation between all features and targets are less than the significance level of 0.05, which indicates that the correlation coefficients are significant. The Kendall-τ correlation calculation results can be seen in Figure 5.2.

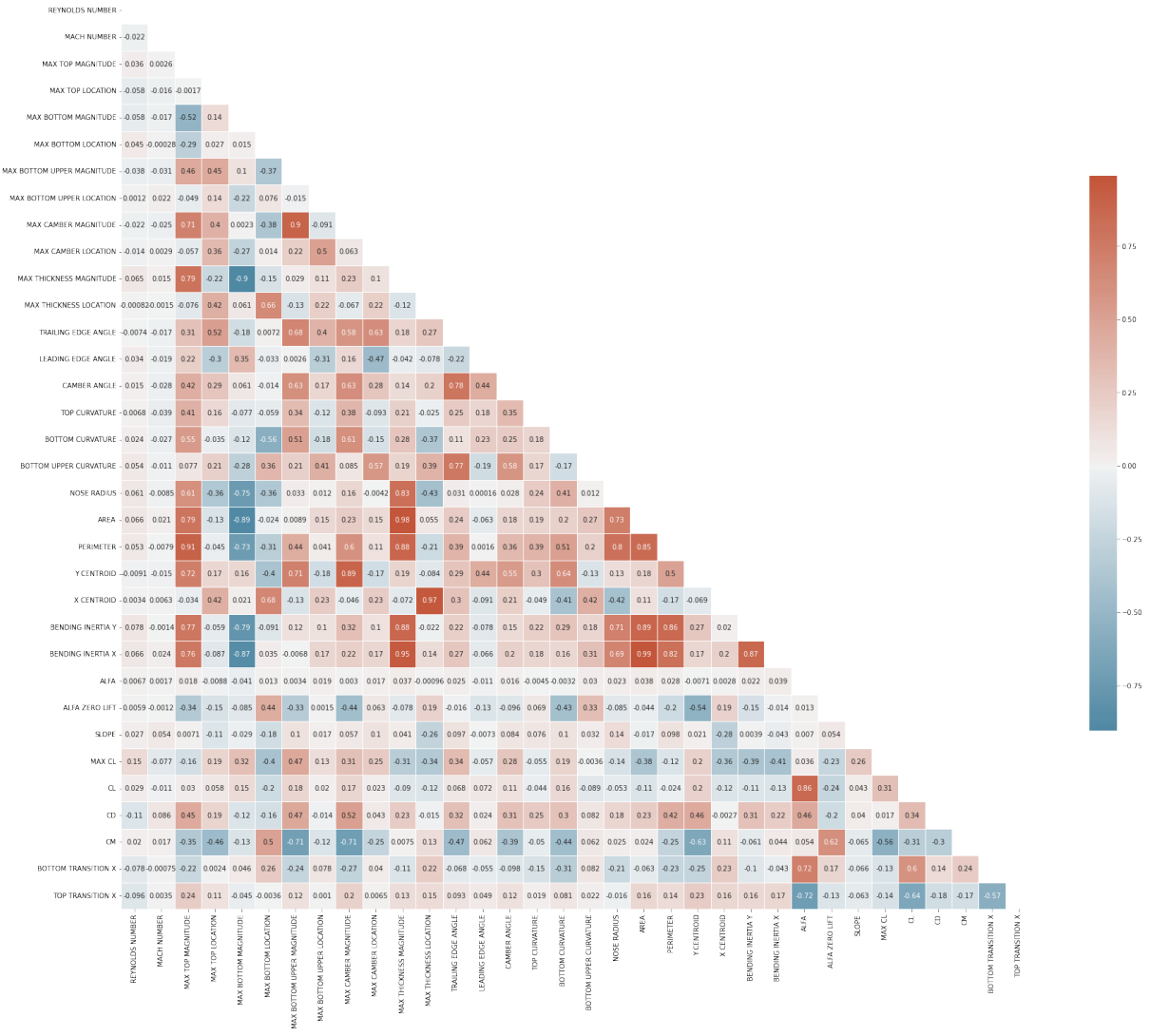


Figure 5.2: Kendall-τ correlation analysis of mined 2D airfoil shape features, structural properties and response variables

This analysis suggests that there are several important relationships between the airfoil parsimonious variables and their corresponding aerodynamic performance metrics. The most important relationships are summarised as follows:

1. There is a strong negative relationship between the zero-lift AoA and the magnitude of the top surface extreme point. The inverse is true for the relationship between the C_d and the magnitude of the top surface extreme point. This means that airfoils with more prominent top surfaces tend to have higher drag coefficients and lower angles where zero-lift is produced.

2. There is a positive relationship between the maximum lift and drag response variables and the maximum camber magnitude, meaning more camber results in higher drag and higher lift production which intuitively makes sense and serves as a data integrity check.

3. There is a strong negative relationship between the lift-curve-slope response variable and the position of the maximum thickness. This means that the closer the maximum thickness to the Leading Edge (LE), the higher the value of the lift-curve-slope.

4. The closer the maximum thickness is situated to the trailing edge, the lower the value of the lift-curve-slope.

5. There is multicollinearity present between multiple shape and structural variables. This is expected seeing that multiple shape and structural properties are dependent on others, for example, a high thickness ratio will lead to an increased area, perimeter, and bending about the $x - x$ and $y - y$ axes – all of which are observed in the data.

6. The correlation between the shape and structural variables and the Reynolds number, Mach number, and angle of attack is negligible, as expected if the data has a high degree of integrity.

5.6 Data partitioning and distribution analysis

The last step is to partition the final 2D performance dataset into training, validation, and testing sets, to facilitate model building and evaluation. The training, validation,

and testing sets are partitioned according to a random 70/20/10 sampling without replacement. This means that the final training dataset has 544 972 records, the validation dataset has 155 706 records, and the testing set has 77 853 records.

After the respective partitions are successfully created, the next step is to analyse if the respective partitions have the same response variable distributions. This is done in two ways. First, the Student-t test and the Mann–Whitney U test are applied to compare the validation and testing set response variable distribution with that of the training set. In each case, the p-value is noted, as seen in Table 5.6, and for each case, the null-hypothesis is not rejected. This means that both the Student-t test and the Mann–Whitney U test agree that the validation and testing set's response variables have the same distribution as the training set ($p<0.05$).

Table 5.6: Student-t test and Mann–Whitney U test p-value results showing that the response variables of the training, testing, and validation partitions have come from the same distribution

Response variable	Student t-test		Mann–Whitney U Test	
	Validation set	Testing set	Validation set	Testing set
	p-value			
Lift coefficient	2.2×10^{-10}	1.3×10^{-22}	1.7×10^{-9}	1.2×10^{-23}
Drag coefficient	4.6×10^{-44}	6.0×10^{-36}	1.2×10^{-73}	1.8×10^{-26}
Moment coefficient	4.6×10^{-113}	8.6×10^{-190}	2.8×10^{-138}	8.8×10^{-172}
Top transition	4.9×10^{-53}	1.8×10^{-24}	2.9×10^{-19}	1.0×10^{-7}
Bottom transition	3.7×10^{-29}	1.8×10^{-11}	1.9×10^{-28}	3.0×10^{-13}
Slope	9.6×10^{-125}	1.1×10^{-59}	3.5×10^{-129}	1.1×10^{-73}
Zero-lift AoA	2.2×10^{-89}	1.1×10^{-14}	7.2×10^{-120}	1.2×10^{-20}
Maximum lift coefficient	3.1×10^{-2}	8.0×10^{-97}	2.4×10^{-2}	9.7×10^{-115}

The second method of distribution comparison is that of visual inspection of the kernel density plots of the most important response variables and operating conditions – see Figures 5.3-5.10 From visual inspection of this figure, it is clear that the training, testing, and validation datasets share similar distributions for the important operating conditions and response variables. This is especially visible in the similar bimodal distributions of the zero-lift AoA and C_l response variables, for all three data partitions. This similarity is also highlighted for the case of the C_d response variable where all three partitions share

the same gamma distribution.

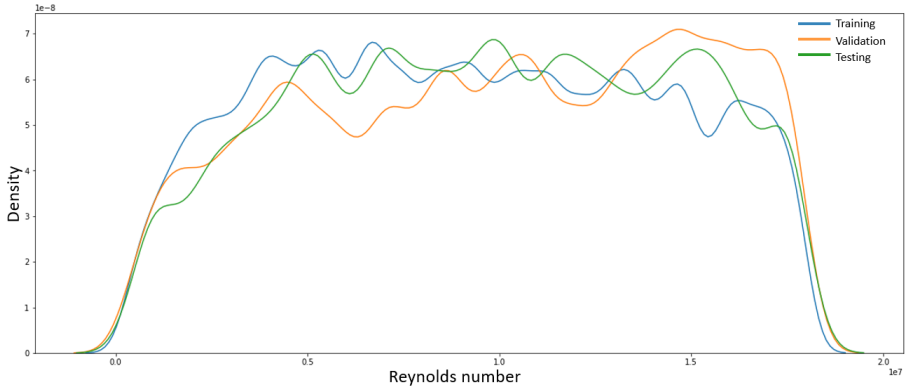


Figure 5.3: Reynolds number distribution

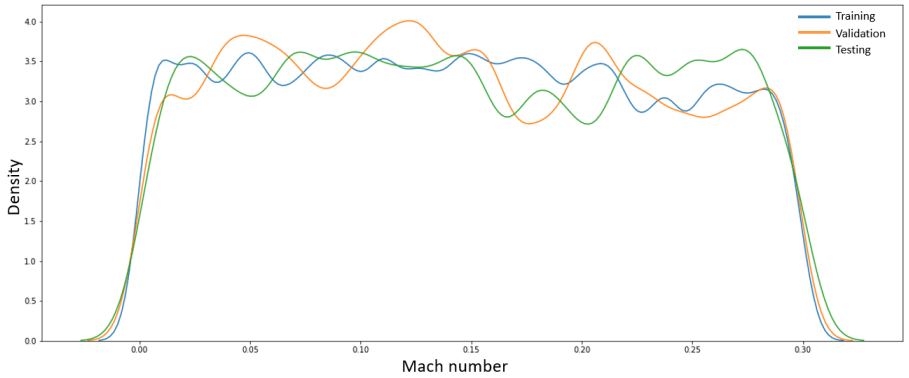


Figure 5.4: Mach number distribution

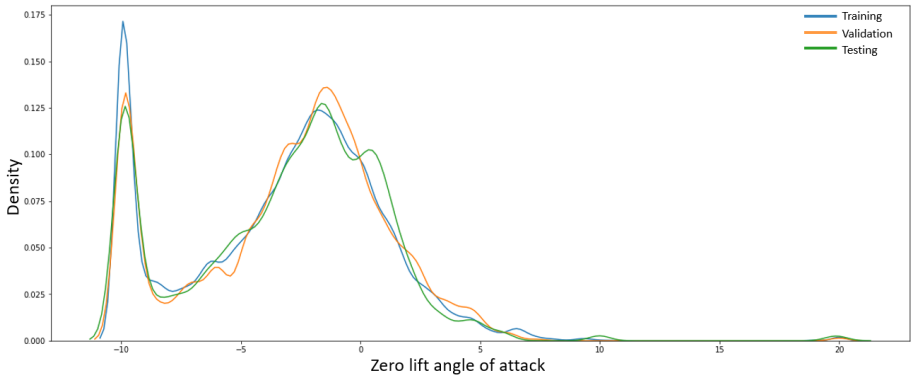


Figure 5.5: Zero-lift AoA distribution

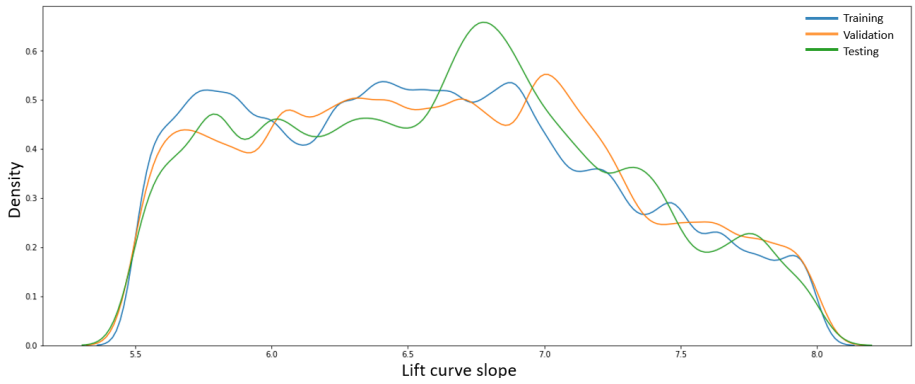


Figure 5.6: Lift-curve slope distribution

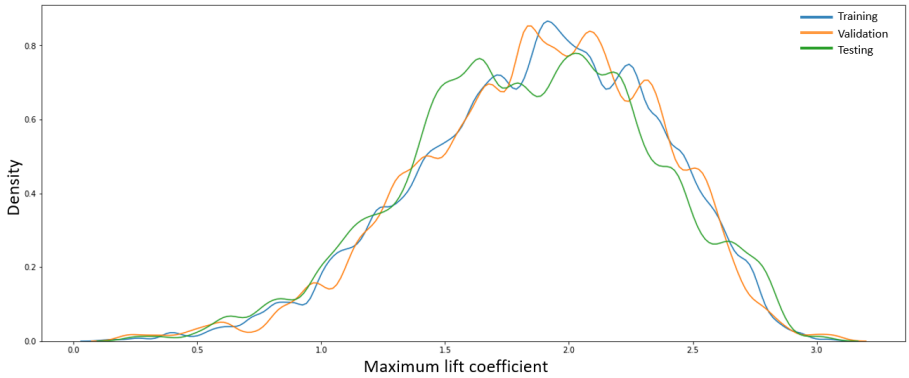


Figure 5.7: Maximum C_l distribution

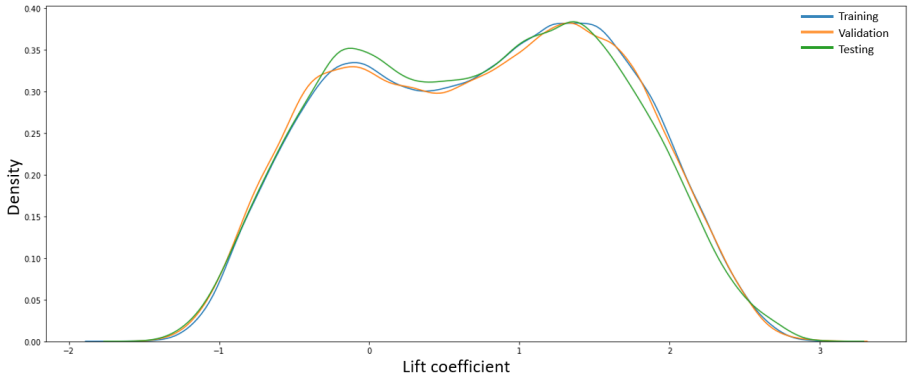


Figure 5.8: C_l distribution

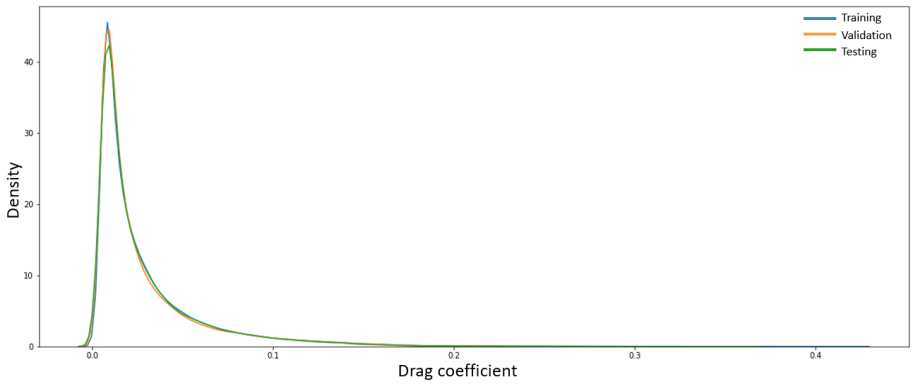


Figure 5.9: C_d distribution

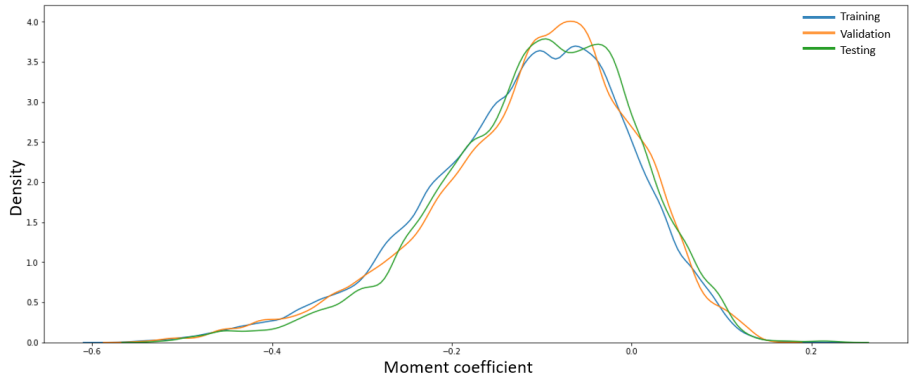


Figure 5.10: C_m distribution

5.7 Conclusion

In this chapter, the data mining process is detailed. This process is then executed to acquire the intermediate dataset; i.e. the dataset with practical and impractical airfoil shapes, and converged and ill-converged XFOIL simulations. This intermediate dataset is then passed through several cleaning steps to arrive at the final 2D performance dataset which is free from impractical airfoil shapes, has no ill-converged simulation results, and has airfoils with desirable shape characteristics. The dataset, therefore, has 6 770 unique airfoils with 115 operating condition simulations per airfoil. This dataset, therefore, has a total of 778 550 samples.

This dataset is later used in Section 6.3 for the development of the PDLM. Also, the data mining process detailed in this chapter is later used in Section 7.2 to develop the datasets required for the development of the airfoil shape generator (Section 7.3.1) and the airfoil shape conversion module (Section 7.6.1).

Finally, the data mining process detailed in this chapter is used in Section 10.2.1, to mine data from a very specific distribution to allow for refinement of the various sub-modules of the DSM to facilitate the optimisation of the JS4's cross-country performance.

Chapter 6

The 2D airfoil performance deep learning module

6.1 Introduction

In this chapter, the development and evaluation of the PDLM are discussed. First, in Section 6.2, an overview of the PDLM is provided, with a specific focus on its differences from XFOIL. Next, the architectural design and modelling approach is discussed in Section 6.3, before the model is developed and optimised in Section 6.4. In Section 6.5, the top-performing models are evaluated and a model is selected for use as the core PDLM module. The selected model is evaluated in two different ways in Sections 6.6 and 6.7 respectively. First, the model's performance and generalisation on the hold-out dataset across all response variables are evaluated before an additional accuracy analysis of the most critical response variables i.e. the C_l and C_d is supplied. Finally, in Section 6.8, the architectures of the core PDLM and IPDLM are discussed and they are integrated with the TSM.

6.2 Model overview

Throughout the design process thus far, every decision made has been in pursuit of a larger goal: developing a deep learning-based model capable of intuitively optimising wing design for cross-country flights. This includes decisions related to the selection of parsimonious variables to describe the airfoil as detaled in Section 4.3, as well as the design of the data mining pipeline discussed in Section 5.2.

As mentioned in Section 1.3.2, this capability of cross-country optimisation by means of intuitively constraining the sectional airfoils present at each wing station is not possible in the TSM. The main reason for this is that the model is restricted by the airfoil performance analysis module used, i.e. XFOIL or CFD simulations. As discussed in Section 2.6, these methods take in operating conditions and raw airfoil x and y coordinates used to describe the airfoil profile and output the 2D airfoil performance. Seeing that the raw x and y coordinates are not parsimonious, setting constraints directly and intuitively on the shape and structural characteristics of this airfoil at the time of generation is not possible. Also, these methods output only the conventional 2D performance characteristics, i.e. lift, drag, and moment coefficients and the top and bottom transition locations. Therefore, constraining the performance of airfoils generated as part of the optimisation process is limited to these few performance coefficients.

For this reason, a new airfoil performance analysis module is required. This module should map intuitive shape and structural airfoil parameters to conventional and non-conventional 2D airfoil performance coefficients (see Table 5.1). In Figure 6.1 the comparison between the proposed PDLM architecture and the XFOIL architecture can be seen.

From Figure 6.1 the differences between the XFOIL architecture and the PDLM architecture are clear. Some notable differences seen here are:

1. The PDLM uses 23 parsimonious variables as airfoil profile descriptions as opposed to XFOIL which uses the raw airfoil x and y coordinate definitions.

2. Both models require the three operating conditions i.e. Reynolds number, Mach number, and Angle of Attack.

3. The PDLM has three additional 2D performance coefficients as part of the response, i.e. the airfoil's zero-lift Angle of Attack, the maximum lift coefficient (that is AoA agnostic), and the airfoil's lift-curve-slope in the linear region.

It is important to note that in this figure, only the core functionality of both XFOIL and the PDLM are compared. That is, the mapping of an airfoil profile description and the operating conditions of Mach number, Reynolds number, and Angle of Attack to 2D performance coefficients. The respective module's capability of mapping an airfoil profile description, Mach number, Reynolds number, and a target lift coefficient, is however not displayed here. Both of these functionalities are required in the TSM as detailed in Section 3.4.3. Seeing however, that both inference processes are very similar and require mostly the same input variables for the same response variables, only one core module needs to be developed, i.e. the PDLM. By bringing slight alterations to this core module in a sub-module, both functionalities are achieved. This is done in the development of the IPDLM and is discussed in Section 6.8.2. The remainder of this section and Sections 6.3-6.7 are therefore only focused on the development of the core PDLM.

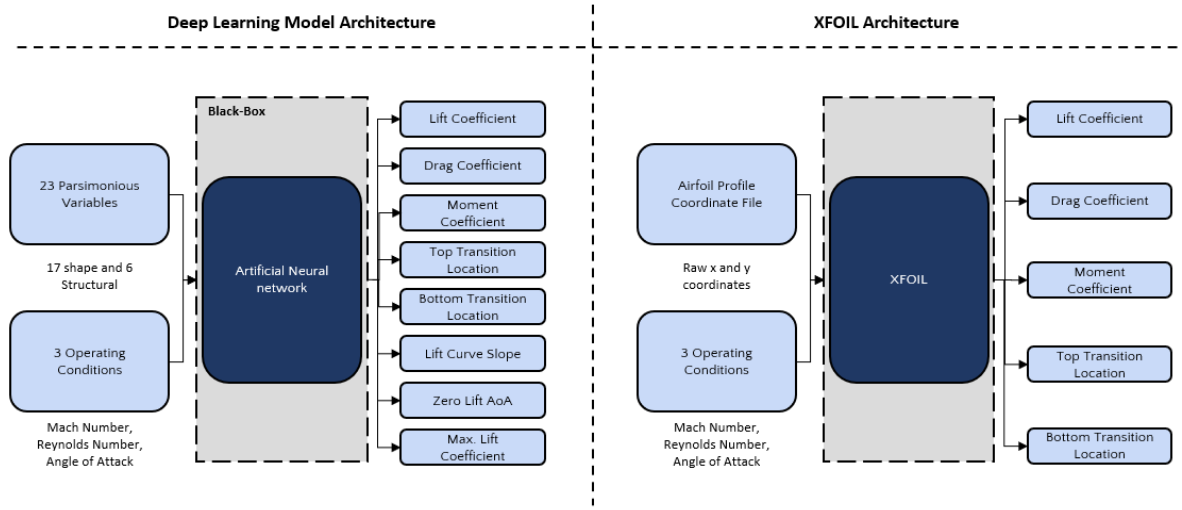


Figure 6.1: Deep learning airfoil analysis module architecture versus XFOIL architecture

6.3 Architecture and approach

For the PDLM, an MLP is used to model the eight target variables as a function of 26 input variables. For the model development, the final cleaned dataset discussed in Section 5.4.3 is used with the data partitions as discussed in Section 5.6. This partitioned dataset's feature and target ranges are normalised on the basis of the training dataset to a range of 0-1. This is done to ensure the data is suitable for use in the MLP.

In all model optimisation pipelines, the models are trained until convergence, with early stopping implemented to prevent overfitting. Overfitting is defined in this case as the point at which there is no improvement in the realised model performance over the last 15 epochs. Early stopping is used to balance training the model for as long as necessary to achieve convergence and to avoid memorisation of the training data, which may result in poor performance on unseen data. This approach facilitates the development of deep learning models that generalise well to new data, whilst still achieving convergence within a reasonable amount of time.

For model parameter optimisation a standard setup is implemented where the efficient Adam (Kingma & Ba, 2014) algorithm is used with the β_1, and β_2 initial decay rates set to 0.9 and 0.999 respectively. Furthermore, in each model experiment each MLP had the following fixed components:

1. Layer constants:
    - Input layer with 26 neurons.
    - Output layer with 8 neurons.
    - Dense layers initialised with He uniform initialisation (He et al., 2015).
    - Batch normalisation after every hidden layer (Ioffe & Szegedy, 2015).
    - Relu activation function (Agarap, 2018).
2. Regularisation constants (Cortes et al., 2012):
    - L2 activity regularisation.

- L2 bias regularisation.
- L1L2 kernel regularisation.

6.4 Hyperparameter optimisation

In this section, the PDLM's development is discussed. This module development consists of three parts: (1) a random search, (2) a refined search, and (3) a machine learning-based hyperparameter search strategy. In the random search strategy, wide bounds are set to find initial feasible hyperparameter settings. These feasible hyperparameter bounds are then used in the second development phase where a refined grid search strategy is employed. Finally, in the third step, machine learning-based hyperparameter search strategies are employed as an alternative attempt to produce a more accurate model than those resulting from the previous two steps.

6.4.1 Random Search

The random search is performed by randomly sampling hyperparameters from a grid with wide upper and lower bounds. The hyperparameter sampling space used here is as displayed in Table 6.1. Here the hyperparameters can be seen alongside their corresponding upper and lower bounds, and their sensitivity.

Table 6.1: Upper and lower hyperparameter bounds for the initial hyperparameter search

Hyperparameter	Lower bound	Upper bound	Sensitivity
Learning rate	1×10^{-5}	1×10^{-1}	10^{-n} $n \in 1, 5$
Number of neurons	8	2048	$2^n n \in 3, 11$
Hidden layers	1	10	n $n \in 1, 10$
L1 regularisation	1×10^{-7}	1×10^{-1}	10^{-n} $n \in 1, 7$
L2 regularisation	1×10^{-7}	1×10^{-1}	10^{-n} $n \in 1, 7$
Batch size	8	2048	2^n $n \in 3, 11$

A total of 250 initial random search pipelines are run to establish a feasible grid of

hyperparameters of which the top 20 performers can be seen in Table 6.2. From these 20 runs, the following features are observed:

1. When analysing the top two performers we can see both have similar validation performance despite having contrasting depth and width.
2. Three out of the five top-performing models have 2 048 neurons with more than six layers.
3. For the top 20 models, batch sizes of 2 048 are always accompanied by larger learning rates.
4. When analysing the top five models, a combination of low L1 and L2 regularisation is observed.
5. When a model has high L1 and low L2 regularisation or vice versa, model performance tends to be poor.
6. Models with fewer layers ($\leq$3), generally, have the worst model performance.
7. Deep-wide models perform better with more regularisation.

Table 6.2: Architectures of the top 20 best performing models as found in the initial random search hyperparameter phase

Batchsize	L1Reg.	L2Reg.	LR	No.layers	No.neurons	Training MSE	Valid. MSE
128	1.0E-07	1.0E-04	1.0E-05	7	2048	0.0034	0.0045
64	1.0E-07	1.0E-05	1.0E-05	4	256	0.0017	0.0051
256	1.0E-06	1.0E-05	1.0E-04	8	2048	0.0014	0.0052
64	1.0E-07	1.0E-04	1.0E-05	10	2048	0.0021	0.0053
2048	1.0E-06	1.0E-07	1.0E-02	8	64	0.0026	0.0062
32	1.0E-06	1.0E-07	1.0E-04	3	16	0.0075	0.0069
256	1.0E-06	1.0E-03	1.0E-05	10	512	0.0020	0.0070
128	1.0E-06	1.0E-04	1.0E-04	7	2048	0.0022	0.0101
16	1.0E-07	1.0E-03	1.0E-03	1	128	0.0107	0.0102
1024	1.0E-06	1.0E-07	1.0E-05	9	128	0.0080	0.0127
2048	1.0E-07	1.0E-05	1.0E-02	4	1024	0.0035	0.0128
32	1.0E-04	1.0E-05	1.0E-02	1	32	0.0152	0.0151
32	1.0E-03	1.0E-04	1.0E-04	5	32	0.0137	0.0152
2048	1.0E-06	1.0E-04	1.0E-03	7	128	0.0019	0.0154
2048	1.0E-04	1.0E-06	1.0E-05	5	32	0.0191	0.0194
512	1.0E-03	1.0E-07	1.0E-05	3	2048	0.0216	0.0216
2048	1.0E-05	1.0E-01	1.0E-02	1	8	0.0251	0.0226
2048	1.0E-07	1.0E-01	1.0E-02	4	32	0.0399	0.0393
2048	1.0E-05	1.0E-01	1.0E-02	9	128	0.0420	0.0415
32	1.0E-01	1.0E-06	1.0E-03	1	16	0.0443	0.0435

6.4.2 Refined search

The results from the initial random search pipeline are used to set the upper and lower hyperparameter bounds for the refined search – see Table 6.3. The changes seen here can be summarised and motivated as follows:

1. Seeing that the initial random search experiment indicates that the majority of the top-performing networks have 2 048 nodes and more than six layers, i.e. close to the upper search bound, in the refined search the upper bounds for both the layers and nodes are increased. This is done to mitigate the effect of the best performing network(s) having parameters close to the extremes of the upper bounds.

2. Table 6.2 shows that models with less than three hidden layers and 64 neurons per layer tend to perform poorly. For this reason, the lower bounds of the number of

neurons and hidden layers are increased to 64 and three respectively.

3. Because the initial random search pipeline does not show that batch sizes of 8-32 yield a top 5 performing architecture, these selections are excluded from the refined search.

4. Training the PDLM requires significant computational resources; therefore, in order to make the process more efficient, the number of hyperparameters and architectural permutations is reduced. Specifically, in each of the refined hyperparameter search experiments, the L1 and L2 regularisations are set to equal values. This is done in an attempt to simplify the process of finding the optimal hyperparameters for the model, given the limited computational resources available. This choice is, however, justified by the fact that, in the initial random search experiments, the top-performing models have a combination of low L1 and low L2 regularisation. This is opposed by the worst performing models either having a combination of low L1 and high L2 regularisation, or high values for both L1 and L2 regularisation.

5. The aforementioned upper and lower hyperparameter bound adjustments increase the probability of generating deeper and wider networks in the refined search. For this reason, the bounds of the L1 and L2 regularisation are also lowered from those used in the initial random search. This is done to mitigate overfitting for these deeper and wider architectures.

Table 6.3: Upper and lower hyperparameter bounds for the refined hyperparameter and architectural search

Hyperparameter	Lower bound	Upper bound	Sensitivity
Learning rate	1×10^{-5}	1×10^{-1}	10^{-n} $n \in 2, 5$
Number of neurons	64	4 096	$2^n n \in 6, 12$
Hidden layers	3	12	n $n \in 3, 12$
L1 regularisation	1×10^{-10}	1×10^{-1}	10^{-n} $n \in 1, 10$
L2 regularisation	1×10^{-10}	1×10^{-1}	10^{-n} $n \in 1, 10$
Batch size	64	2 048	2^n $n \in 6, 11$

A total of 250 refined hyperparameter search experiments are run. The 20 best, median, and worst performing model architecture and hyperparameter combinations can be seen in

Tables E.1, E.2, and E.3 respectively. From these experiments, the following conclusions are drawn:

1. The majority of the top-performing networks either have deep-wide or shallow-wide architectures.

2. Irrespective of architecture, i.e. (deep-wide, shallow-narrow, deep-narrow, or shallow-wide networks), the top five performing models have small batch sizes and low L1 and L2 regularisation.

3. The ten worst-performing networks have high learning rates and high values for both L1 and L2 regularisation.

4. Deep-wide networks generally perform poorly if they have high learning rates and high regularisation, regardless of the batch sizes and architecture.

For further analysis of the top five models, their respective learning curves are analysed – see Figure 6.2. From this analysis, it is observed that training for all models stabilised after approximately 20 epochs and that early stopping terminated training before 150 epochs are reached. It can also be observed that there is a sufficiently small gap between the validation and training curves across all five models, indicating good generalisation capability.

Given that the top five models share similar validation Mean Squared Error (MSE) values, any one of these models can be chosen as the model for integration in the final sailplane cross-country optimisation system. These models are revisited in Section 6.5 during the final model selection.

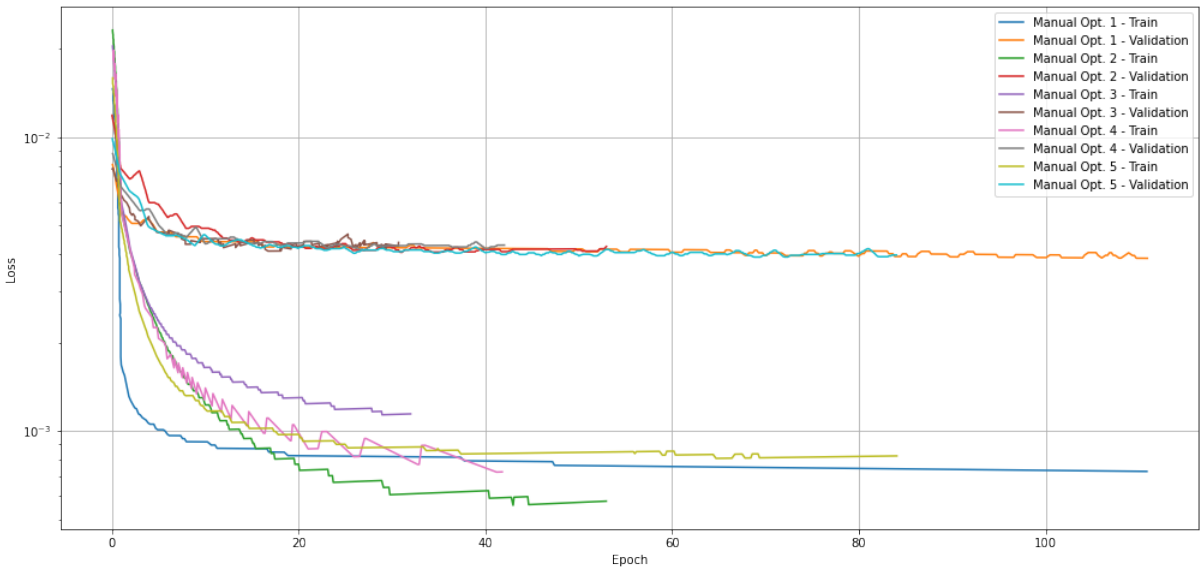


Figure 6.2: Learning curves of the top five models from the refined search experiments

6.4.3 Hyperparameter search algorithms

In the final phase of the model development objective, the Hyperband and Bayesian optimisation algorithms (L. Li et al., 2016; Snoek et al., 2012) are used in an attempt to arrive at a model, or set of models, with superior performance to those found using the earlier manual grid search.

The first model search architecture to be used in this regard is the Hyperband algorithm. The underlying principle of the Hyperband algorithm (L. Li et al., 2016) exploits the intuition that if a hyperparameter configuration is destined to be the best after a large number of iterations, it is more likely than not to perform in the top half of configurations after a small number of iterations. That is, even if performance after a small number of iterations is very unrepresentative of the configuration's absolute performance, its relative performance compared with many alternatives trained with the same number of iterations is roughly maintained.

For the Hyperband algorithm, the upper and lower hyperparameter bounds are set as in Table 6.4.

Table 6.4: Upper and lower hyperparameter bounds for the Hyperband hyperparameter search

Hyperparameter	Lower bound	Upper bound
Learning rate	1×10^{-6}	1×10^{-1}
Number of neurons	8	4 096
Hidden layers	1	15
L1 regularisation	1×10^{-10}	1×10^{-1}
L2 regularisation	1×10^{-10}	1×10^{-1}
Batch size	8	2 048

These bounds are not set to exactly the same values as those seen in Table 6.3 of the refined search for two reasons. The first of these is that this method is more computationally efficient and therefore a larger number of hyperparameter permutations can be explored. The second is that in the event that the bounds are unnecessarily restricted in the refined search, they are relaxed here in an attempt to find the optimal architecture. This is regardless of whether the optimal network has hyperparameters beyond those searched for in the refined search. Specifically, the learning rate's lower bound is decreased to 1×10^{-6} , the neurons per layer lower bound are decreased to eight, the number of hidden layers upper and lower bounds are changed to one and 15 respectively, and the batch size's lower bound is decreased to eight. With Hyperband search, the algorithm determines the sensitivity, i.e. step size of each designated ML hyperparameter and hence no sensitivity on the hyperparameters is required to be specified as input to the search process. The algorithm is executed with the function arguments as seen in Table 6.5:

Table 6.5: Hyperband algorithm optimisation parameters used in the machine learning-based hyperparameter optimisation phase

Parameter	Discussion	Value
Objective function	What the algorithm uses to guide the hyperparamter optimisation process	Validation MSE
Factor	The reduction factor for the number of epochs and the number of models for each bracket	3
Maximum epochs	The maximum number of epochs to train one model	500
Hyperband iterations	The number of times to iterate over the full Hyperband algorithm	3
Maximum trials	The number of unique models to evaluate	1000

After convergence of the Hyperband search algorithm, the optimal model found has a training MSE of 0.00030, a validation MSE of 0.0044, and the architectural and hyperparameter setup as seen in Table 6.6.

Table 6.6: Hyperparameter and architectural definition of the best model found by the Hyperband optimisation algorithm

Parameter	Value
Layers	5
No. neurons per layer	1 100
Learning rate	5×10^{-5}
Batch size	400
L1 and L2 regularisation	1×10^{-10}

These parameters of the best Hyperband model show the same tendencies as noted in the manual optimisation pipeline: that is, lower learning rates and low L1, and L2 regularisation coupled with a deep-wide architecture generally yield the best-performing models. It is also interesting to see that the optimisation method converged to an architecture that has hyperparameters within the bounds of the refined search and similar to those seen in Table E.1.

This model's learning curve is shown in Figure 6.3. From this curve, it is clear that: (1) there is no overfitting or underfitting present, (2) the training loss decreases to a point of

stability after approximately 20 epochs, (3) early stopping terminates training before it reaches 180 epochs, and (4) the gap between the training and validation learning curves, i.e. the generalisation gap, is sufficiently narrow.

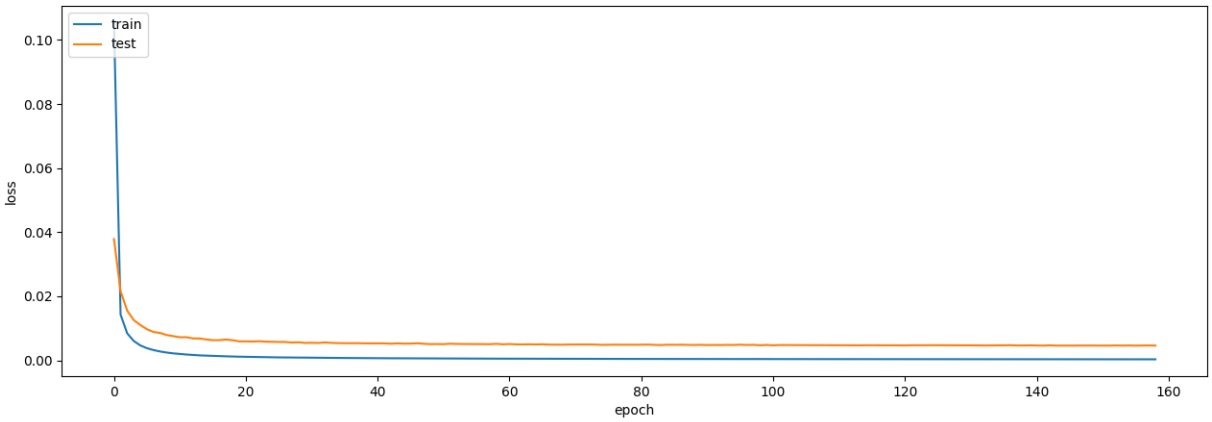


Figure 6.3: Optimal Hyperband optimised model's learning curve

The next, and final, hyperparameter search algorithm to be employed is the Bayesian search framework (Snoek et al., 2012). Bayesian optimisation is a global optimisation method for noisy black-box functions. Applied to hyperparameter optimisation, Bayesian optimisation builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set. By iteratively evaluating a promising hyperparameter configuration based on the current model, and then updating it, Bayesian optimisation aims to gather observations revealing as much information as possible about this function and, in particular, the location of the optimum. This algorithm tries to balance exploration (hyperparameters for which the outcome is most uncertain) and exploitation (hyperparameters expected close to the optimum).

For the Bayesian optimisation algorithm the upper and lower hyperparameter bounds are once again set as in Table 6.4 and the algorithm is executed with the function arguments of Table 6.7:

Table 6.7: Bayesian optimisation algorithm parameters used in the machine learning-based hyperparameter optimisation phase

Parameter	Discussion	Value
Objective function	What the algorithm uses to guide the hyperparamter optimisation process	Validation MSE
Number of initial points	Number of randomly generated samples as initial training data for Bayesian optimisation	50
Alpha	The value added to the diagonal of the kernel matrix during fitting	0.0001
Beta	The balancing factor of exploration and exploitation	2.6
Maximum trials	The number of unique models to evaluate	1 000

The optimal model as found by the Bayesian optimisation process has a training MSE of 2.37e-4, a validation MSE of 0.0048, and the architectural and hyperparameter setup as defined in Table 6.8.

Table 6.8: Hyperparameter and architectural definition of the best model found by the Bayesian optimisation algorithm

Parameter	Value
Layers	4
No. neurons per layer	1 800
Learning rate	5×10^{-5}
Batch size	1 000
L1 and L2 regularisation	1×10^{-10}

Here again it is observed that the best performing Bayesian optimised model shares similar characteristics with both the optimal Hyperband and manual optimised models. This model also has a deep-wide architecture with a low learning rate and low values for regularisation. The Bayesian model has one fewer hidden layer than the Hyperband model; however, here each hidden layer has 700 more nodes. The similar architectures lead to similar performance on the validation set – with the Hyperband model slightly outperforming the Bayesian model in this area. However, in terms of training MSE the Bayesian model greatly outperforms the Hyperband model as well as the top five manually optimised models. When analysing the learning curve in Figure 6.4, similar comments to those made for the Hyperband case can be made here. That is, the learning curves show no sign of over- or underfitting; early stopping ceased training before the full 300 epochs

are reached, and the generalisation gap is acceptably small.

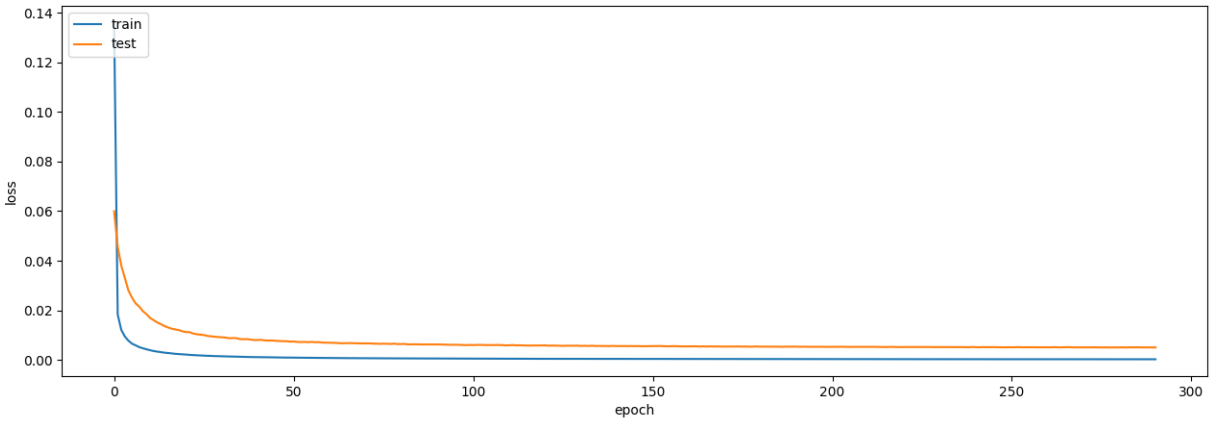


Figure 6.4: Optimal Hyperband optimised model's learning curve

6.5 Model selection

In this section, the comparison of the best manual search models, Hyperband and Bayesian models, are discussed in an attempt to establish which one of the models has the best generalisation and prediction capacity. The first method of analysis is by means of Random Matrix Theory (RMT) (Martin & Mahoney, 2021). This is a data agnostic method that can to an extent be used to compare the generalisation capability of MLP models with the same and differing architectures. The second method of analysis is that of the comparison between the actual versus predicted plots of the models under question for the two most important 2D airfoil performance coefficients, i.e. the lift and drag coefficients.

6.5.1 Model weight analysis

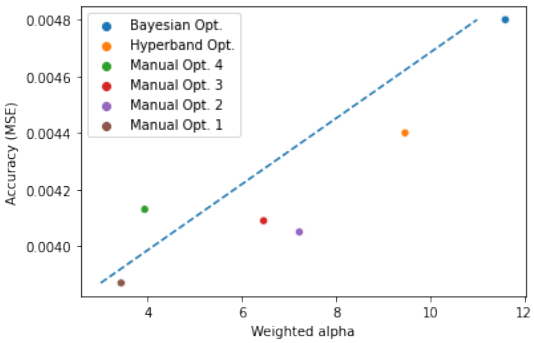
As mentioned, the first method used to compare the MLP models is the data agnostic model of Martin and Michael (Martin & Mahoney, 2021). Martin and Michael showed that RMT can be applied to analyse the weight matrices of deep neural networks to quantify generalisation error. They proposed the Theory of Heavy Tailed Self-Regularisation (HTSR), which states well-trained, well-correlated layers should have eigenvalue distribu-

tions of weight matrices that are significantly different from the Marchenko-Pastur (MP) distributions random bulk, and, even more specifically, be heavy-tailed. To make their research reproducible they created WeightWatchers, a python package that encapsulates their research findings and implements the main methods discussed by Martin et al. (Martin & Mahoney, 2021). In summary, the WeightWatchers package can be used to gauge the test error of a series of trained models, without needing access to training or test data. Following is a list of metrics that can be used as a proxy for testing errors (items 1-3) or for analysing the quality of learning (item 4). Here, lower values for average alpha, average log spectral norm, or log alpha norm allude to better model performance and generalisation, whereas an alpha metric below two indicates that the model may be overtrained and hence early stopping is necessary. These HTSR metrics are summarised as follows:

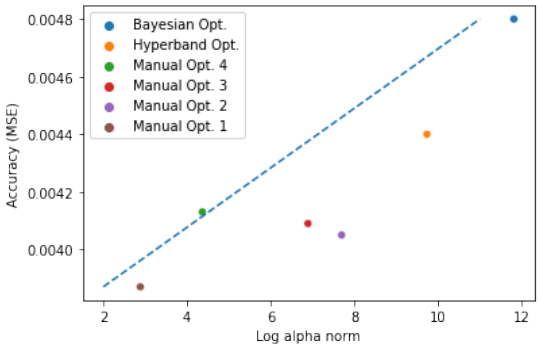
1. Average alpha: used to compare one or more DL models with different hyperparameter settings but of the same depth.
2. Average log spectral norm: used to compare models of different depths.
3. Average weighted alpha and log alpha norm: suitable for DL-models of differing hyperparameters and depths simultaneously.
4. Alpha metric: used to detect overtraining of a model.

The WeightWatchers package is used to analyse the four manually optimised models with the lowest validation MSE metrics, as well as both the Bayesian and Hyperband optimised models. For simplicity, the manually optimised models are indexed from 1 to 4 with one representing the model with the lowest validation MSE and 4 representing the highest of the group. In Figure 6.5 the results of this analysis can be seen: in Figure 6.5a the weighted alpha and validation errors are compared and in Figure 6.5b the log alpha norm and validation errors are compared for the six models. Both Figures 6.5a and 6.5b show that there is a strong linear correlation between the model validation MSE and the weighted alpha and the log alpha norm parameters. The analysis of both HTSR metrics also indicates that the manually optimised models have a lower than expected

generalisation error. Also, the average of the weighted alpha values across all layers, for the six models, ranges between 3.3 and 11.57, indicating that early stopping is applied effectively and overtraining is not expected to be present in any of the models.



(a) Weighted alpha versus validation error for the top performing models



(b) Log alpha norm versus validation error for the top performing models

Figure 6.5: Analysis of the Empirical Spectral Density of the top performing models

6.5.2 Model accuracy analysis

In this section the single best model of the Hyperband optimisation, Bayesian optimisation, and manual optimisation processes are compared with regard to two aspects: (1) the analysis of R^2, and (2) the analysis of the C_l and C_d prediction accuracies.

In terms of the coefficient of determination, as calculated on the validation dataset, the Hyperband model achieves an R^2 of 0.876 whereas the Bayesian model has an R^2 of 0.869. This is compared to the top-performing manually optimised model with an R^2 of 0.892. Hence, by purely looking at the coefficient of determination, the manually optimised model with the lowest validation MSE has the best performance of the three models.

Next is the comparison of the models' C_l and C_d predictions, as done on the validation datasets. This is done by comparing the C_l and C_d prediction of the Bayesian optimised model with the manually optimised model, and the Hyperband optimised model with the manually optimised model respectively. The results of this analysis can be seen in Figures 6.6, and 6.7. From this analysis, the following remarks can be made:

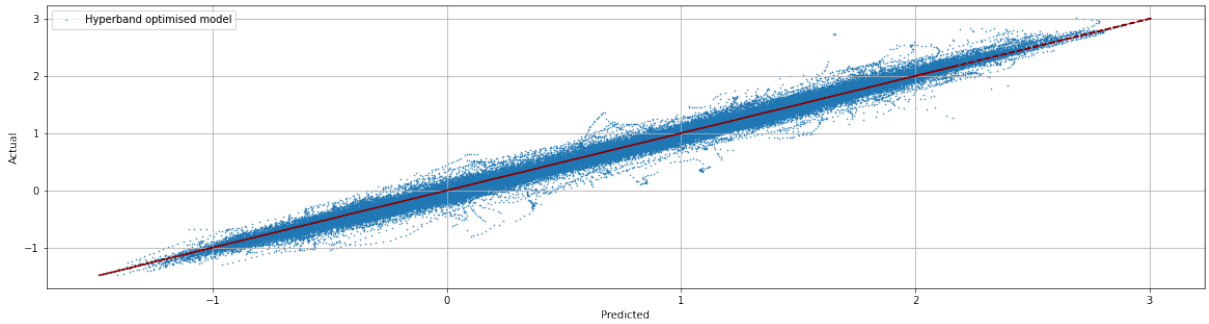
1. Both the Hyperband and Bayesian models have a significantly higher variance for the C_l predictions compared to the manually optimised model.

2. The Hyperband and the Bayesian models' C_l prediction variance is higher for cases where the Cl lies between 0 and 2 in contrast to the constant variance throughout all C_l cases for the manually optimised model.

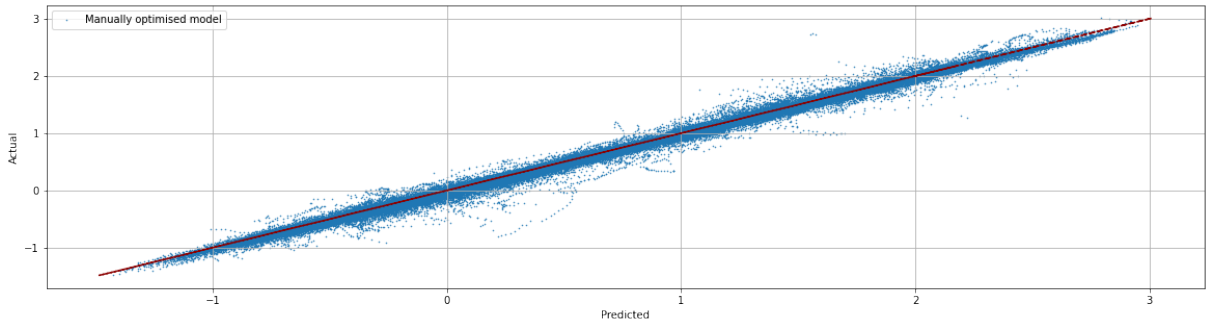
3. For extremely high lift coefficient cases, although slight, the manually optimised model tends to underpredict the Cl coefficient.

4. The Hyperband model more frequently overpredicts the cases with high lift coefficients as opposed to the Bayesian and manually optimised cases.

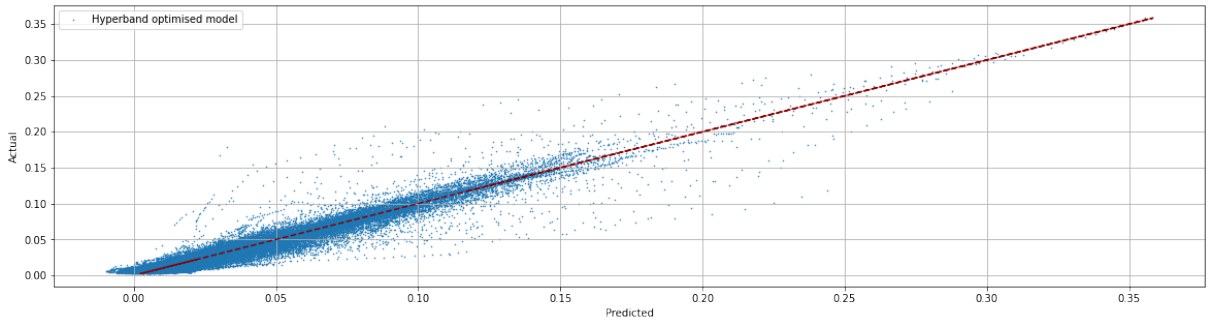
5. For the drag coefficient predictions, both the Hyperband and Bayesian optimised models struggle with C_d predictions at the upper and lower extremes, i.e. high and low drag values respectively. These models also predict negative drag coefficients for cases where very low drag is present. Negative drag is physically not possible and also not present in the dataset, hence showing poor generalisation capability for the C_d in this specific operating condition region.



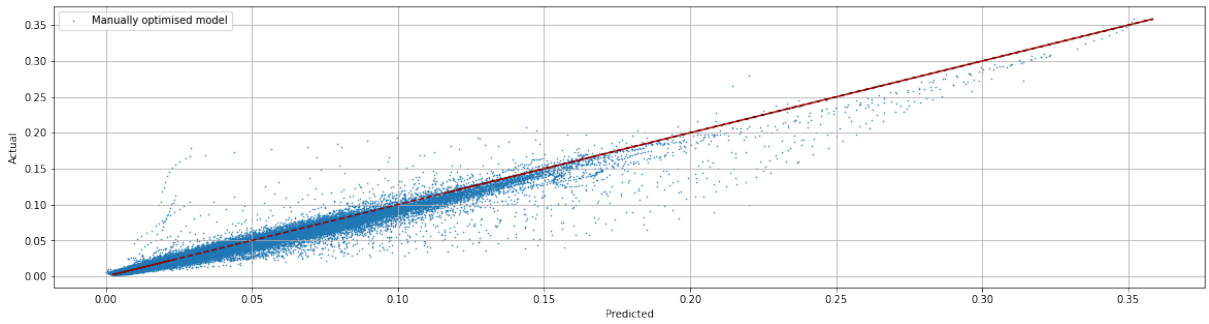
(a) Hyperband optimised C_l prediction vs actual C_l



(b) Manually optimised C_l prediction vs actual C_l

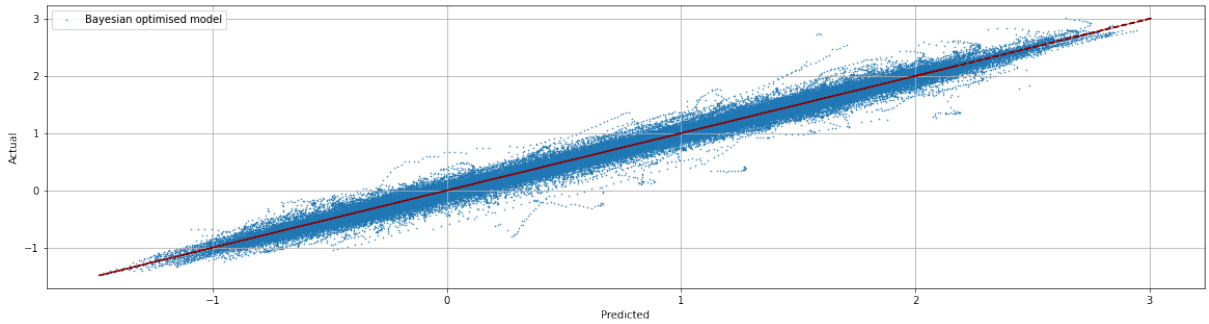


(c) Hyperband optimised C_d prediction vs actual C_d

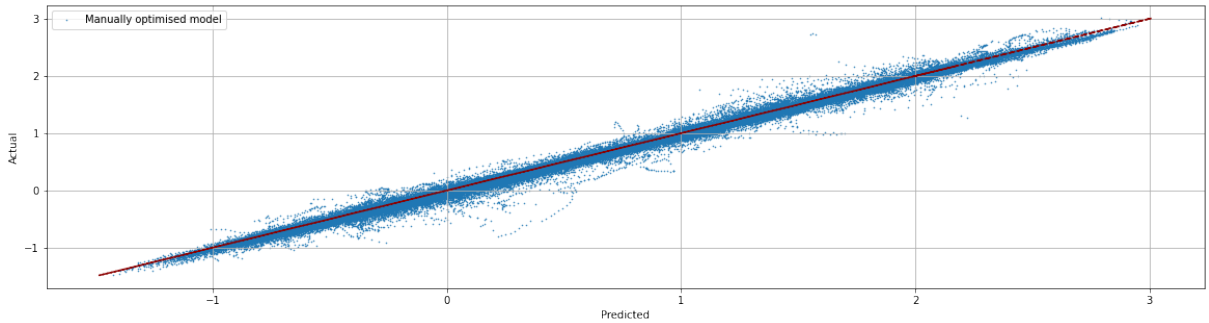


(d) Manually optimised C_d prediction vs actual C_d

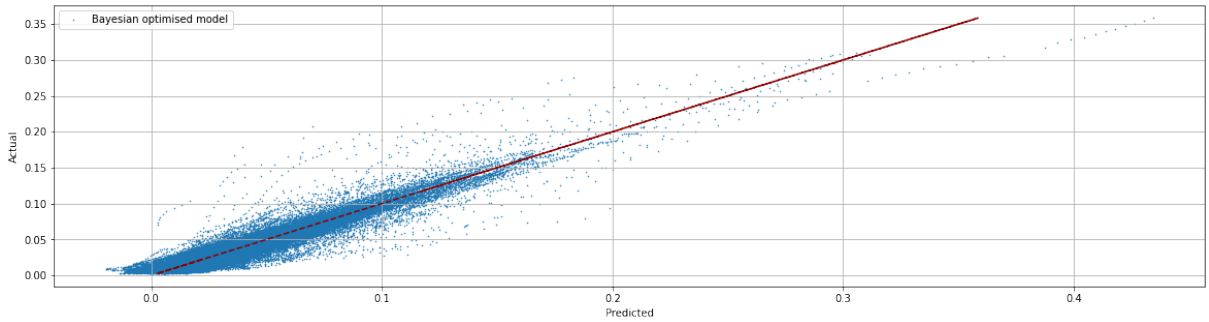
Figure 6.6: Manual and Hyperband optimised model predictions for lift and drag coefficients on the hold-out dataset



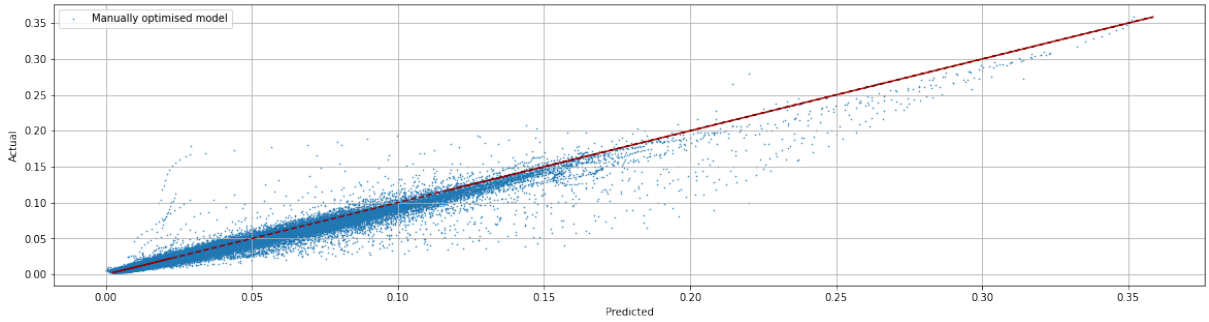
(a) Bayesian optimised C_l prediction vs actual C_l



(b) Manually optimised C_l prediction vs actual C_l



(c) Bayesian optimised C_d prediction vs actual C_d



(d) Manually optimised C_d prediction vs actual C_d

Figure 6.7: Manual and Bayesian optimised model predictions for lift and drag coefficients on the hold-out dataset

6.5.3 Final model choice

In Section 6.5.1 it is shown that the manually optimised models have lower generalisation error (as per the HTSR theory) than the Hyperband and Bayesian optimised models. Also, in Section 6.5.2, it is shown that the manually optimised model with the lowest validation MSE outperforms both the Hyperband and Bayesian optimised models in the two areas of greatest importance, i.e. the C_d and C_l prediction. Specifically, it is shown that the manually optimised model has the lowest variance in the prediction of the C_l and C_d response variables. From this analysis, it can also be seen that the Hyperband and Bayesian optimised models have low generalisation for the C_d response variables – where negative drag is predicted for specific operating conditions. Here it is also shown that the manually optimised model has a higher coefficient of determination, calculated for all features, than those of the Hyperband and Bayesian optimised models.

For these reasons it is therefore concluded that the manually optimised model is indeed the better performing model and its use as the core PDLM is therefore motivated. The final PDLM, therefore, has the hyperparameters and architectural setup of Table 6.9, with the fixed components as defined in Section 6.3:

Table 6.9: Hyperparameter and architectural definition of the best model found by the manual search pipeline

Parameter	Value
Layers	7
No. neurons per layer	2 048
Learning rate	1e-5
L1 and L2 regularisation	1e-10

This results in a final model with 25 307 144 parameters of which 25 278 472 are trainable and 28 672 are non-trainable.[1]

[1]The batch normalisation layer has fixed weights that cannot be adjusted during training – referred to as non-trainable weights. These non-trainable weights are used to monitor and maintain the mean and variance of the input data as it passes through the layer during the training process.

6.6 Performance and generalisation analysis

In this and the following section, the performance of the best-performing model as discussed in Section 6.5.3 and hence chosen as our core PDLM, is discussed. This section, however, is specifically focused on the generalistion evaluation of the PDLM across all eight response variables – as determined on the hold-out dataset.

Before predictions and performance metrics can be calculated on the hold-out dataset, the data needs to be transformed into the appropriate format. This means first normalising the input and response variables with the same scaler metrics as derived from the training dataset. For each sample in the transformed hold-out dataset the corresponding 26 input variables are then given to the PDLM to determine the respective eight response variables. Next, the response variables are descaled to allow for the chosen performance metric calculations. These metrics are: (1) the feature means, (2) the feature standard deviation, (3) the coefficient of determination, (4) the MSE, (5) the MAE, and (6) the MAPE. For all eight response variables, these metrics are therefore calculated for the descaled predictions and the hold-out dataset.

The results from this analysis can be seen in Table 6.10. From this table, and as previously shown in Section 6.5.2, the average coefficient of determination across all eight response variables is 0.8925. This result is significant seeing that, although surrogate models are low-fidelity models and per definition less accurate than higher-fidelity models, a predicted relative improvement is expected to produce an actual improvement, even when accounting for the MAPE error margin. This R^2 of 0.8925 means that roughly only 11% of the variation in the hold-out dataset cannot be explained by the model. When studying the R^2 values for all response variables it is observed that, for the first six response variables, the average is 0.9719. This is significantly higher than the overall average indicating that, for these six variables, most of the variation in the dependent variable can be explained by the independent variables in the MLP model. However, for the two remaining response variables, i.e. the zero-lift AoA and lift-curve slope response variables, this R^2-average is 0.65. This value indicates a moderate to good fit of the model to the data for these

two response variables, meaning that the independent variables are moderately correlated with the dependent variable and can predict it with a moderate degree of accuracy.

Table 6.10: Analysis of best performing MLP on the hold-out dataset

Response variable	Prediction		Actual		Model performance metrics			
	Mean	Std.	Mean	Std.	R^2	MSE	MAE	MAPE
Maximum C_l	1.8416	0.4860	1.8360	0.4910	0.9297	0.0170	0.0720	2.33
C_l	0.7405	0.8885	0.7150	0.8960	0.9927	0.0059	0.0518	4.59
C_d	0.0303	0.0338	0.0280	0.0310	0.9481	0.0001	0.0035	10.38
C_m	-0.1040	0.1092	-0.1040	0.1090	0.9871	0.0002	0.0085	6.31
Top transition location	0.2803	0.2670	0.2820	0.2590	0.9864	0.0009	0.0163	6.24
Bottom transition location	0.4043	0.2703	0.4000	0.2690	0.9875	0.0009	0.0141	2.63
Zero-lift AoA	-2.7598	4.0049	-3.0440	4.1650	0.6988	5.2254	0.7861	12.65
Lift-curve slope	6.5826	0.5699	6.6140	0.6350	0.6095	0.1578	0.2479	2.06

From the results in Table 6.10, the average and standard deviations of the predicted data and the hold-out data can be seen. Overall, the average difference between the predicted means and the actual means for all response variables is -0.13%. This low average MAPE means that the prediction model is well calibrated, and it is making predictions that are consistent with the out-of-sample data. More specifically, having similar means across all samples is a desirable property for our prediction model, as it indicates that the model is not over- or underpredicting the target variables. This is important because it means that the PDLM can be relied upon to make accurate predictions when new unseen data is encountered in the end-to-end optimisation. In evaluating the means, it is important to understand where the biggest average over- or underprediction is. From analysis, it is noted that the biggest underprediction average is for the C_d response variable with a value of -7.59%. This is aligned with the findings of the MAPE analysis where this response variable has the second-highest error rate when measured with this metric. The biggest overprediction average is for the zero-lift Angle of Attack with a value of 10.3% – again aligned with the MAPE findings where this variable has the lowest prediction accuracy. Reasons for these over- or underpredictions are ascribed to the following reasons:

1. The model does not have sufficient training samples for lower Reynolds numbers and thick airfoil combinations and hence does not generalise well in these regions, especially for the zero-lift Angle of Attack and C_d response variables.

2. The zero-lift Angle of Attack response variable is not calculated by XFOIL directly but derived from other XFOIL simulated aerodynamic characteristics, which leads to more noise for this variable. For instance, if the lift coefficient from the XFOIL simulation is noisy, so would the derived calculations of the zero-lift Angle of Attack response variable be, as these calculations are sensitive to variation in lift versus Angle of Attack changes,

3. These two variables have more complex relationships than the remaining six variables and hence the model does not generalise as well for them.

4. The 23 parsimonious variables are less strong predictors of the C_d and zero-lift Angle of Attack response variables than for the remaining six variables and hence lower accuracy.

Similar conclusions can be drawn when analysing the standard deviations of the predicted data versus the out-of-sample data. Overall, the average difference between the predicted and out-of-sample standard deviation is 0.67%. The standard deviation is a measure of the spread of the data, which indicates how much the individual data points deviate from the mean. When the standard deviation of the actual data and the predicted data are similar, it suggests that the model is able to capture the variation present in the data. Seeing that our model very accurately captures the variation in the data across all the response variables, it can be concluded that the model is robust and generalises well on the unseen dataset.

The final analysis done to ensure that the model is accurate in the prediction of all eight response variables is the MAPE analysis. Here, the MAPE is calculated as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{6.1}$$

where n is the total number of samples, y_i is the actual value of the i-th sample, $\hat{y}_i$ is the predicted value of the i-th sample, and $|\cdot|$ represents the absolute value. Overall the average MAPE across all response variables is 5.52. This means that on average, for a given sample in the hold-out set, the PDLM's prediction is within 94.48% of the actual value.

From studying Figure 6.8, depicting the kernel density estimate of the MAPE distribution for all eight response variables, it is also clear that most of the MAPE values across all metrics lie between 0-3% with only a few outliers – indicating severe over- or under prediction. This further confirms the robustness and generalisation capability of the PDLM for out-of-sample data.

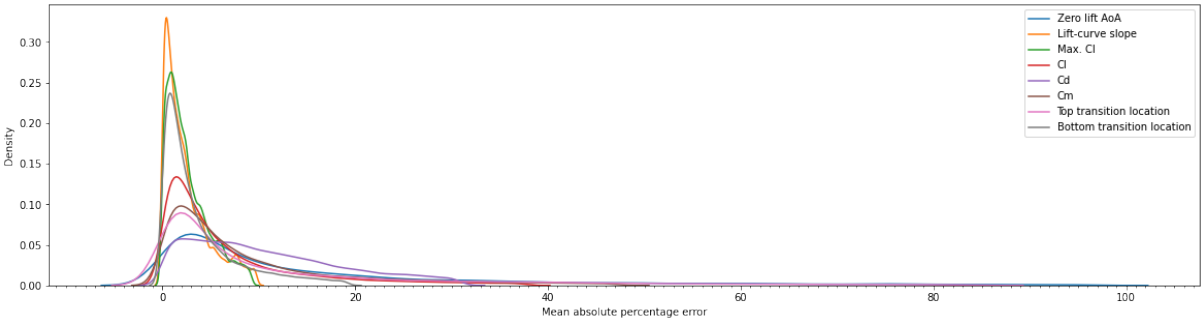


Figure 6.8: Kernel density estimate of the mean absolute percentage error distribution as calculated on the hold-out dataset across all eight response variables

Although this accuracy in terms of MAPE and R^2 (excluding the zero-liftAoA and lift-curve slope response variables) is aligned with similar studies in the literature (Andrés-Pérez & Paulete-Periáñez, 2021; Lalonde et al., 2021; Renganathan et al., 2021; Tao & Sun, 2019) it is not sufficient for global optimisation of the cross-country performance of a SOTA standard class glider. As seen in Section 9.4, the PDLM's accuracy is however sufficient for optimisation of simple baseline glider, however, the high error in the C_d prediction will make optimisation of a SOTA glider difficult. This is because in the optimisation of a SOTA sailplane design, the typical expected performance improvements are incremental (in the order of 0.5%-1%) and hence low error in the underlying PDLM module is required. It is later shown in Section 10.2 how this inaccuracy problem is addressed by means of mining more data from a (smaller) desired distribution and retraining the various modules before integration in the DSM.

Upon analysing the results presented in this section, it is evident that the proposed PDLM exhibits adequate prediction capability across multiple response variable modes for usage in the development and validation of the optimisation module, as seen in Chapter 9.

However, as pointed out earlier, to enable optimisation of a SOTA model, a higher degree of accuracy is required. These concerns are addressed in Chapter 10.

6.7 Critical response variable analysis

In this section, the PDLMs accuracy and generalisation capability are evaluated in terms of the lift and drag response variables. This is done in four steps. First, the distribution of the prediction error (i.e. the MAE) of the PDLM is analysed. Next, the distribution of the predicted versus the actual response variables is inspected. From here a random sample of 250 predictions and actuals are drawn to enable visual inspection of the quality of predictions. Finally, an analysis of the residuals for each prediction sample is done.

6.7.1 C_l analysis

When analysing the lift coefficient MAE distribution of the hold-out set (Figure 6.9a) it can be seen that most samples have a MAE below 0.052. Further analysis shows that the C_l MAE has a median value of 0.038 and a 75th percentile of 0.067. The difference between the mean and the median is due to outliers skewing the data and hence the distribution is not normal but rather log-normally distributed. This means that, although most of the predictions are accurate, there are very few cases of extreme inaccuracy. The biggest MAE is found to be 1.182 and is for an extremely thick and highly cambered airfoil.

To further isolate these outlier cases, all samples with a MAE higher than 0.1 are filtered for; this equates to $\sim$ 1 842 samples or 2.3% of the hold-out set. Analysis of these 1 842 samples shows that they all have undesirable profiles that are near-impractical and very underrepresented in the training dataset. Broadly, the unfavourable characteristics of these 1 842 samples can be summarised as follows:

1. The outliers exhibit a higher degree of curvature on the top surface (average of 1.89), while the non-outliers have an average of 1.47.

2. The outliers have sharper radii at the bottom trailing edge regions (average of 5.72) than the non-outliers (average of 3.44).

3. The outliers have more prominent nose radii at the leading edge (average of 0.123) compared to the non-outliers (average of 0.078).

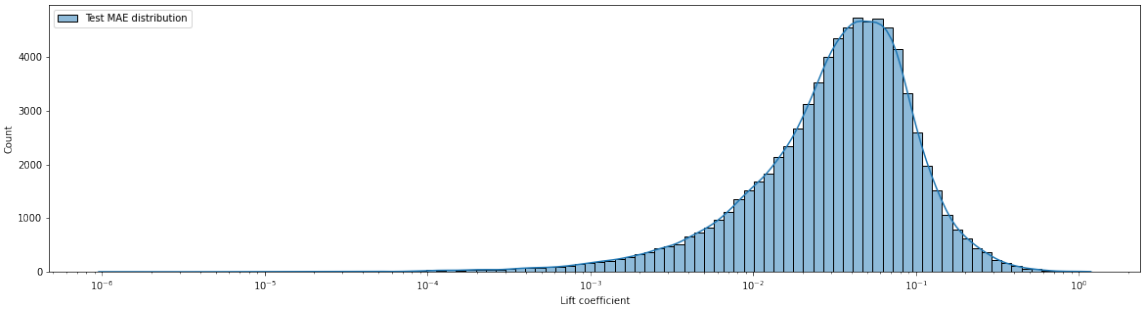
4. The outliers have a higher level of total thickness (average of 0.31) compared to the non-outliers (average of 0.23).

5. The outliers exhibit a higher degree of camber (average of 36.3°) compared to the non-outliers (average of 27.7°).

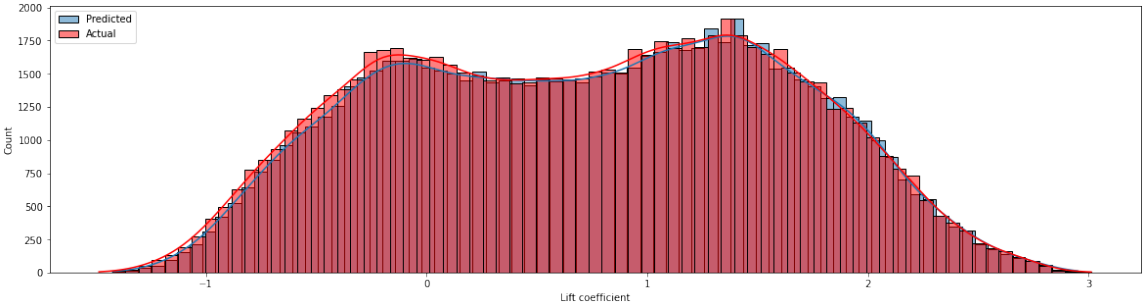
For the outlier cases identified with the above-mentioned characteristics, the predicted lift coefficient is higher for low AoAs and lower for high AoAs than the actual values. This indicates that the model lacks the ability to generalise well for these cases. However, because these airfoils have near-impractical shapes and hence are not suitable for sailplane flight, the model's inaccuracy for these cases is acceptable.

When evaluating the distribution of the predicted versus actual lift coefficient, as seen in Figure 6.9b, it is clear that the chosen model accurately captures the bimodal distribution. This means the model is able to appropriately predict lower lift coefficients for lower AoAs and higher lift coefficients for higher AoAs.

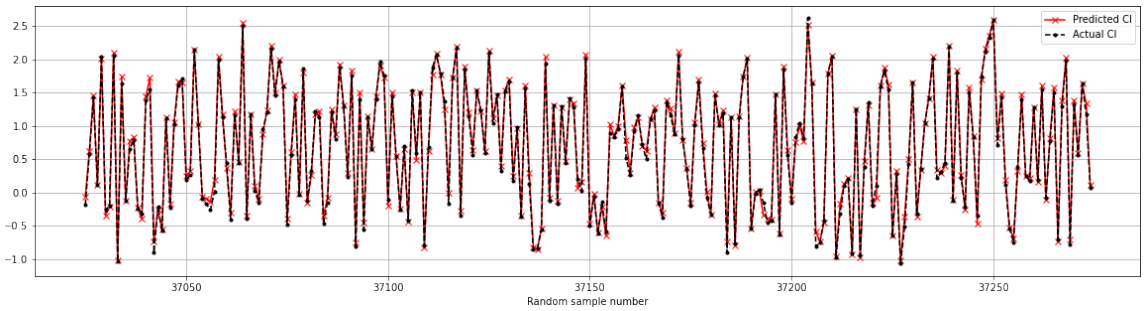
When analysing Figure 6.9c it can be observed that, for a random sample without replacement of 250 predictions, the model accurately predicts the C_l coefficient. It can also be seen here that for extremely high lift coefficients, i.e. cases where C_l is above 1.5, the model tends to slightly overpredict the lift coefficient. However, for most cases i.e. cases with C_l values between -1 and 1.5, the model is accurate in its predictions. Finally, from the residual plot in Figure 6.9d it can be observed that the residuals are mostly clustered around 0, indicating good performance and generalisation capability.



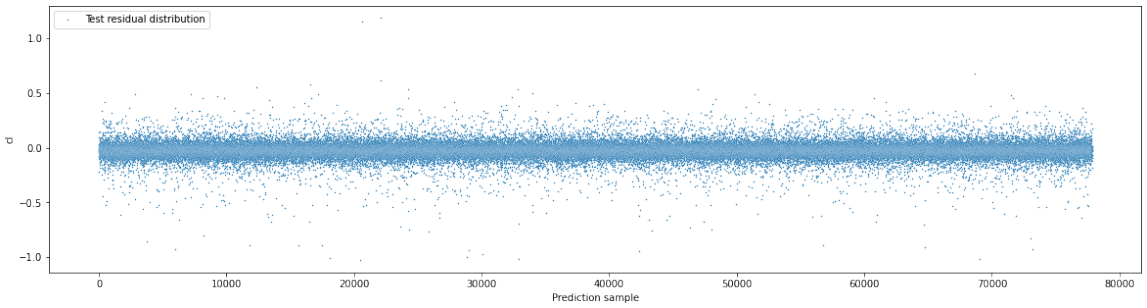
(a) Prediction error distribution of the C_l response variable



(b) Predicted vs actual distribution of the C_l response variable



(c) C_l predictions vs actual for 250 random samples from the hold-out dataset



(d) C_l residuals for all samples in the hold-out dataset

Figure 6.9: Evaluation of the lift response variable prediction and actual values in the hold-out dataset

6.7.2 C_d prediction analysis

To analyse the quality of the PDLM's drag coefficient predictions, first the error distribution of this feature is investigated, as shown in Figure 6.10a. This analysis indicates that most of the C_d MAE values are below 0.004 with a mean of 0.00347, a median of 0.00179, and a 75th percentile of 0.00373. This indicates that for most samples, there is a high degree of accuracy in C_d predictions. However, again there is the presence of outliers which represent a few cases of extreme over- or underprediction. These outliers result in a lognormally distributed C_d MAE distribution. To identify these outliers, the 99.5th percentile is used as the cut-off criterion. This translates to samples with an MAE of 0.01594 or 159.4 drag counts and equates to roughly 2.8% of the data. When analysing this inaccurate sub-set, the following is clear:

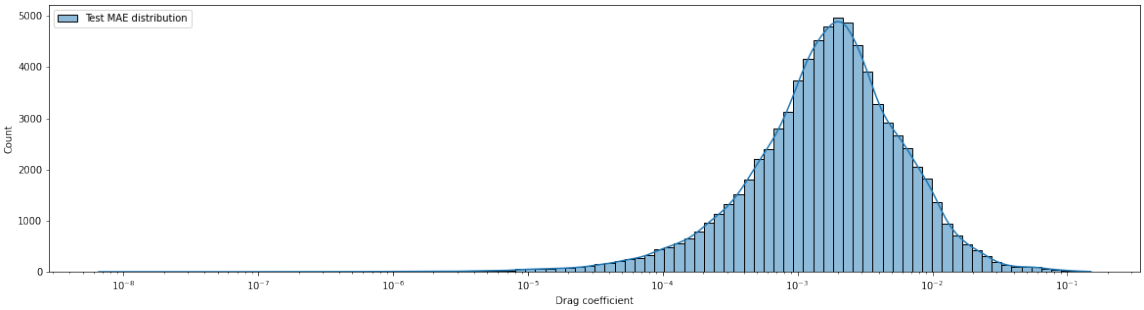
1. The outliers have sharper radii at the bottom leading edge regions (average of 2.51) than the non-outliers (average of 2.01).
2. The outliers exhibit a higher degree of camber (average of 34.5°) compared to the non-outliers (average of 27.8°).
3. The outliers exhibit a higher degree of curvature on the top surface (average of 1.56), while the non-outliers have an average of 1.47.

This is aligned with the findings of the previous section where the C_l predictions are also observed to be less accurate for highly curved, thick airfoils.

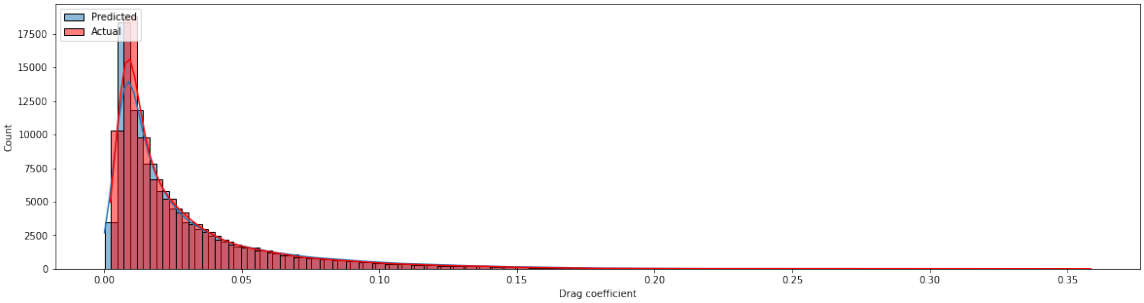
To analyse if the chosen model is able to accurately capture the drag coefficient variance, the predicted versus actual drag values are evaluated (see Figure 6.10b). From this plot, it can be observed that the model accurately captures the gamma distribution with approximately the same shape, centre, and variability of the drag values present in the hold-out dataset. This naturally leads to the conclusion that the chosen model accurately captures the drag value ranges of the diverse airfoil shape and flight condition combinations present in the hold-out dataset – attesting to the model's generalisation capability.

From the analysis of the 250 randomly drawn C_d predictions versus actual values in the hold-out dataset it is clear that the model has a high degree of accuracy for the majority of the samples. However, the same overprediction tendency is seen here where the PDLM overpredicts the C_d values for cases of high separation i.e. high AoAs. Although not desirable, in the case of sailplane cross-country optimisation, overprediction of the C_d is more favourable than underprediction. This is because, when the model finds an optimal design with a tendency of drag overprediction for high separation cases, this design is more likely to have good performance when evaluated with the XFOIL-based TSM. Although countered by the fact that the PDLM also overpredicts lift in this region, Table 6.10 shows that the C_l overprediction tendency is significantly lower than the drag overprediction.

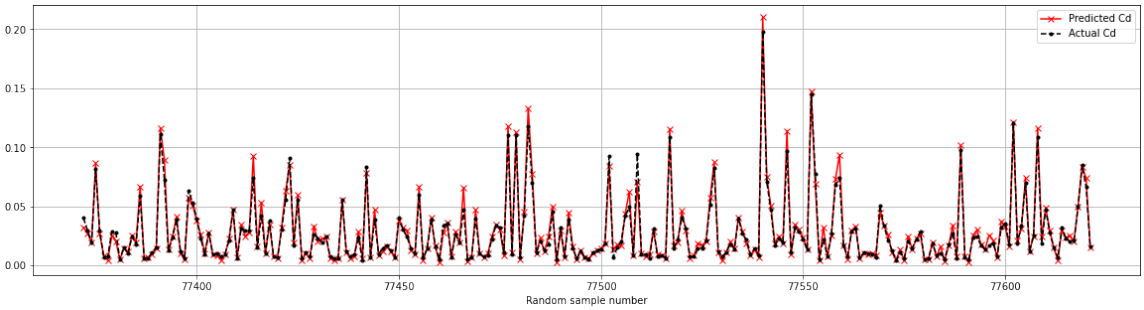
Finally, the C_d prediction residuals are investigated across all samples in the hold-out dataset as seen in Figure 6.10d. From this figure, it is observed that most of the model residuals are centered around 0. This indicates that the model has captured the underlying patterns in the data and is providing predictions that are aligned with the actual values. Also in this figure, the extreme cases of model under- or overprediction can be seen as addressed earlier in this section.



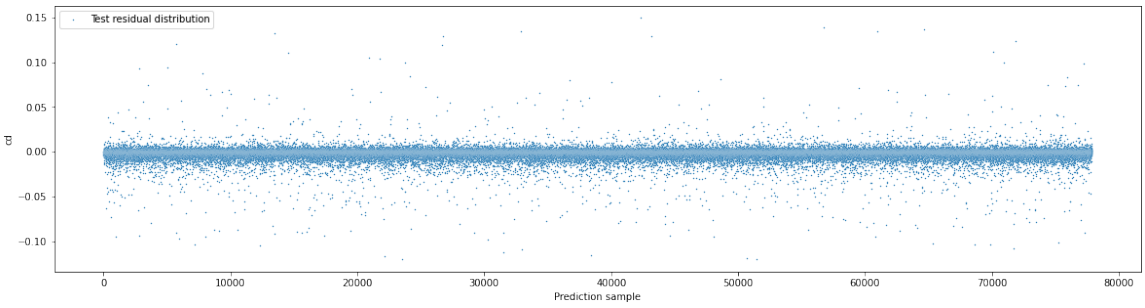
(a) Prediction error distribution of the C_d response variable



(b) Predicted vs actual distribution of the C_d response variable



(c) C_d predictions vs actual for 250 random samples from the hold-out dataset



(d) C_d residuals for all samples in the hold-out dataset

Figure 6.10: Evaluation of the drag response variable prediction and actual values in the hold-out dataset

6.8 Module integration

In this section, the integration of the core PDLM, as well as the IPDLM into the TSM is discussed. These two respective modules are integrated into the TSM, to replace the full functioning of the traditional XFOIL-based airfoil performance analysis module.

In the TSM, XFOIL is called in two distinct ways to calculate the total 3D profile drag of a given wing configuration (see Section 3.4.3). The first call is to calculate the airfoil lift-curve slope and zero-lift Angle of Attack, for each airfoil along the wing span. To replace this functionality the PDLM is used. The detail of this implementation is seen in Section 6.8.1. The second set of XFOIL-based functionality to be replaced is for the calculation of an airfoil's C_d for a given C_l, Reynolds number, and Mach number.

As mentioned earlier in Section 6.2, seeing that the PDLM does not directly calculate the C_d for a given C_l but rather for a given Angle of Attack, a sub-module named the IPDLM is developed. This sub-module is developed such that the exact XFOIL-based airfoil performance analysis module of the TSM is reproduced. The detail of this IPDLM architecture as well as its integration can be seen in Section 6.8.2.

6.8.1 PDLM integration

The PDLM is integrated into the TSM by replacing the first step of the XFOIL-based calculation, i.e. calculating an airfoil's lift-curve slope and zero-lift Angle of Attack for each station along the wing span. Once this sub-module is integrated with the TSM, its individual inference process is as depicted in Figure 6.11 with the inference process as in Algorithm 2.

Algorithm 2 Prediction of Zero-lift Angle of Attack and Lift-curve Slope using the PDLM

Require: $\mathbf{z} \in \mathbb{R}^{23\times1}$, $\mathbf{c} \in \mathbb{R}^{3\times1}$ with $\mathbf{z}$ the parsimonious variable definition vector and $\mathbf{c}$ the operating condition definition vector

Ensure: α_0, a where α_0 the zero-lift angle of attack and a the lift-curve slope

1: Scale $\mathbf{z}$ and $\mathbf{c}$ to form $\hat{\mathbf{z}} \in \mathbb{R}^{23\times1}$, $\hat{\mathbf{c}} \in \mathbb{R}^{3\times1}$

2: Use $\hat{\mathbf{z}}$, and $\hat{\mathbf{c}}$ as input to the PDLM to calculate eight response variable $\mathbf{R} \in \mathbb{R}^{8\times1}$

3: Inverse scale $\mathbf{R}$ to form $\hat{\mathbf{R}} \in \mathbb{R}^{8\times1}$

4: Index $\hat{\mathbf{R}}$ for α_0, a

5: **return** α_0, a

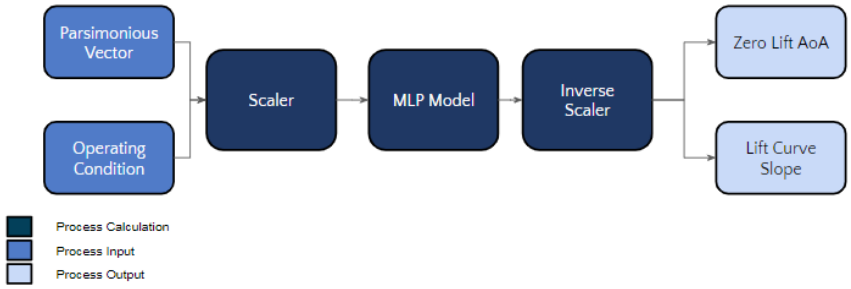


Figure 6.11: Depiction of the final inference process of the PDLM module as integrated with the DSM

6.8.2 IPDLM integration

As mentioned, the second way in which XFOIL is called in the TSM is in the calculation of the 2D profile drag coefficient for a given lift coefficient and operating condition. The way XFOIL does this calculation is by iteratively varying the Angle of Attack until the required C_l is found and then using these simulation conditions to calculate the corresponding profile drag. That is, once the Angle of Attack has been solved for, this is used as an input to find the C_d.

Seeing that the PDLM takes parsimonious variables together with a Reynolds number, Mach number, and angle of attack as input in order to generate the specified response

variables, it is possible to change this module's architecture to mimic what XFOIL does. That is, to develop the IPDLM, the architecture of the PDLM is altered to allow for the iterative variation of the Angle of Attack to arrive at the required C_l. Then, indexing of the model's response variable output is done to find the appropriate C_d. To ensure that this search process is as efficient as possible, a logarithmic search and optimisation routine is used. Once this sub-module is integrated with the TSM, its individual inference process is as depicted in Figure 6.12 with the inference process as in Algorithm 3.

Algorithm 3 Profile drag prediction for a given target lift coefficient with the IPDLM

Require: $C_{L_{target}} \in \mathbb{R}$, $\mathbf{z} \in \mathbb{R}^{23\times1}$, $\mathbf{c} \in \mathbb{R}^{3\times1}$ with $\mathbf{z}$ the parsimonious variable definition vector, $\mathbf{c}$ the operating condition definition vector, and $C_{L_{target}}$ the target lift coefficient

Ensure: C_L, C_D

1: Receive target lift coefficient $C_{L_{target}}$ from dynamic station LLT simulation
2: Scale $\mathbf{z}$ and $\mathbf{c}$ to form $\hat{\mathbf{z}} \in \mathbb{R}^{23\times1}$, $\hat{\mathbf{c}} \in \mathbb{R}^{3\times1}$
3: Use $\hat{\mathbf{z}}$, and $\hat{\mathbf{c}}$ as input to the PDLM to calculate C_L
4: Inverse-scale C_L to obtain actual response variable value $\hat{C_L}$
5: **while** $|\hat{C_L} - C_{L_{target}}| > 0.0001$ **do**
6: Update AoA
7: Re-run prediction pipeline with all inputs constant except updated AoA
8: Index outputs for C_L, and C_D
9: Inverse-scale C_L and C_D to obtain actual values
10: **end while**
11: **return** C_L, C_D

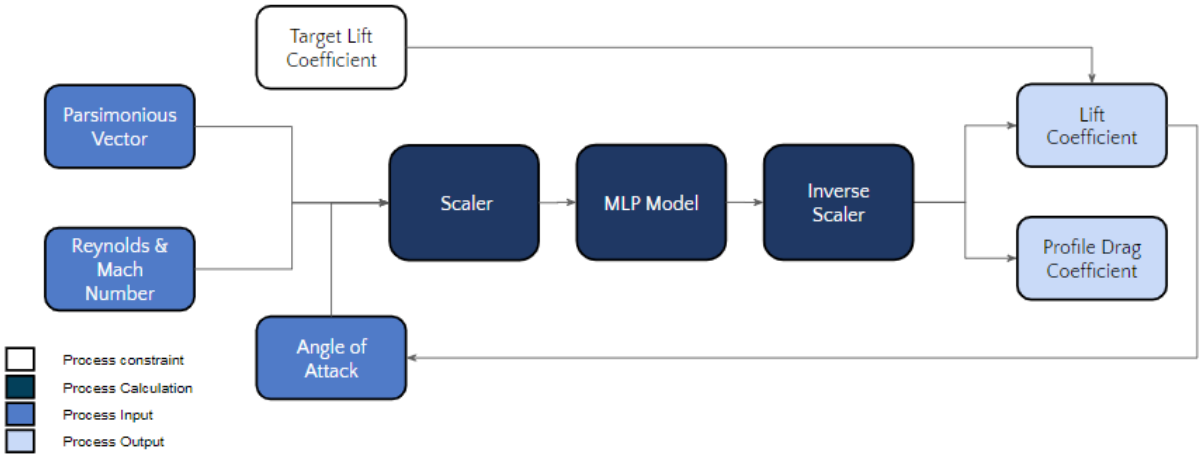


Figure 6.12: Depiction of the final inference process of the IPDLM module as integrated with the DSM

6.9 Conclusion

In this chapter, the PDLM is developed. Here, a two-phased modelling approach is taken. First, a manual model tuning pipeline is employed to find a model architecture with sufficient generalisation and predictive capacity. This manual tuning pipeline consists of two phases. The first phase is concerned with building an initial manual search pipeline where rough bounds are set on the hyperparameter search space. This is done in an attempt to establish a feasible grid of hyperparameters that results in architectures with good performance. From here a refined manual search pipeline is executed. In this, the hyperparameter bounds are modified based on the findings of the initial hyperparameter search experiments. A set of five models with good performance in terms of validations MSE are chosen from these experiments. Next in the two-phased modelling approach, two established hyperparameter search algorithms are employed i.e. the Hyperband- and Bayesian optimisation methods. The objective here is to see if the established methods are able to find an architecture with superior performance, in terms of validation MSE, to the ones found in the manual tuning pipeline.

Once the Hyperband- and Bayesian optimisation algorithms converge, a total of seven

models are available for analysis. That is five manually optimised models and two from the respective hyperparameter optimisation algorithms. These seven models are critically analysed by means of the HTSR, and analysis of the predictive capacity of the two most important response variables, i.e. the C_l and C_d. From this analysis, it is shown that the superior model is the manual-optimised model with the lowest validation MSE. This model is therefore chosen as the final PDLM. This PDLM has a total of 25 307 144 parameters, of which 25 278 472 are trainable and 28 672 are non-trainable.

Next, in Section 6.6 the accuracy of the selected model is analysed across all eight response variables using the held-out test set. In this section, it is shown that the PDLM is sufficiently accurate in predicting the eight defined response variables for a diverse range of airfoil and operating conditions and hence its use in the development and validation of the optimisation module is justified. Specifically, it is shown that the model has an average MAPE of 5.52 across all eight response variables. The confidence in model accuracy is further motivated by the high coefficient of determination reported across all eight response variables (0.89). The robustness and generalisation capability of the model are analysed by means of investigating the average and standard deviation values of the predicted data versus those of the held-out dataset. An average difference of -0.13% and 0.67% is reported respectively for these two metrics.

However, as mentioned, the accuracy is not sufficient for usage in SOTA glider optimisation and hence the approach to address this discrepancy is discussed. Specifically, the accuracy of the PDLM can be improved by narrowing the airfoil diversity such that data is only mined for workable standard class sailplane airfoils. The downstream PDLM will then have the advantage of a more focused distribution which will in turn have a stronger signal mapping parsimonious variables to 2D aerodynamic performance. This module refinement can be seen in Section 10.2

Chapter 7

Development of the airfoil shape mapping module

7.1 Introduction

In this chapter, the development of the airfoil generator and airfoil shape conversion modules, as introduced previously in Section 1.3.2, are discussed. The main objective of the airfoil generator is to enable the generation of an airfoil profile in terms of its 23 parsimonious variable definitions, whereas the airfoil shape conversion module is tasked with converting the 23 parsimonious variables back to raw x and y coordinates. Section 7.2, first gives an overview of the airfoil shape dataset required to develop these two respective modules. In Section 7.3 the airfoil generator module's development is first addressed, alongside its integration in the TSM. In Sections 7.4 and 7.5 the airfoil generator module's accuracy in terms of cluster segregation and efficacy as a generative model is then evaluated.

Next, the development of the airfoil shape conversion module is discussed. In Section 7.6 this module's development is approached by first giving an overview of the requirements, before discussing the model building approach, and then finally developing and evaluating the final optimised module.

In the final part of this chapter, the airfoil generator and airfoil shape conversion modules are combined to form the airfoil shape mapping module. This combined module is required to verify the generalisation capability and accuracy of the two underlying modules. In Section 7.7 an overview of this module is given before the details of airfoil fitting are discussed, as required for the final generalisation verification.

7.2 Overview of the airfoil shape dataset

For both the airfoil generator and airfoil shape conversion modules a rich dataset containing the six SPM shape coefficients, the 23 parsimonious variables as well as the corresponding raw x and y coordinates is required. This dataset is referred to as the airfoil shape dataset, the schema of which is as detailed in Table 5.5.

To mine the required data, a data mining pipeline is set up with Kedro.[1] This data mining pipeline follows the same procedure as depicted in Figure 5.1; however, in this case, the aerodynamic performance is not required and hence no filtering for XFOIL convergence took place. This means that every airfoil generated, with converged parsimonious variable calculations and feasible geometry (see Section 5.3), is included in the airfoil shape dataset.

In total, data is mined for 600 000 uniformly sampled airfoils. Of these, only 18 275 airfoils either did not converge within tolerance when calculating either the shape or structural properties numerically or did not fall within the desired shape bounds. For each airfoil, 80 x coordinates, 80 y coordinates, 17 shape characteristics, 6 structural characteristics, and 6 SPM coefficients are generated, resulting in a dataset with 29 input parameters (shape, structural, and SPM coefficients), 581 725 samples, and 160 response variables (x and y coordinates). This dataset is manipulated and reshaped as required in the airfoil generator and airfoil shape conversion models respectively, as will be seen in the following sections.

Again here it is important to note that the airfoil shape dataset will include a very

[1] Kedro: https://kedro.readthedocs.io/en/stable/

diverse set of profiles that may or may not include a large set of applicable standard class sailplane airfoils. This means that all downstream modules trained with this data set will possess the capacity to generate general aviation airfoils as well as standard class sailplane airfoils. This is again sufficient for use in Section 9.2 where the optimisation module is developed and tested to understand its discrepancies and shortcomings, however it is not sufficient for SOTA glider optimisation. This is because in the SOTA optimisation process it is desirable to generate airfoils from distributions known to yield good performance thereby not wasting computational resources exploring general aviation designs but rather focus on optimising current SOTA airfoil profiles and combinations. This discrepancy is throughly addressed in Section 10.2.2 however the remainder of this chapter is focused on the development of the airfoil shape mapping module making use of the airfoil shape dataset.

7.3 Development of the airfoil generator module

In this section, the airfoil generator module is developed. First, an overview of the requirements for the airfoil generator module is given before the approach to model development is discussed. The module is then developed and optimised.

7.3.1 Overview

For the airfoil generator module, a method of generating airfoil shapes, in terms of the 23 parsimonious variables is required, as discussed in Section 4.3. That is, the main purpose of the airfoil shape generation module is to efficiently generate realistic airfoil parsimonious variables that represent a practical airfoil profile. From the correlation analysis done in Section 5.5 it is observed that the specified 23 parsimonious variables have strong covariance due to the sequential structure implicit in the feature definitions. It is therefore important that the airfoils generated by this module have the same covariance and hence the same global and local structure in their parsimonious variable output. Also,

because the output of the airfoil generator is used as part of the input to the PDLM and IPDLM, the airfoils generated by this module require the same distribution as present in the final 2D performance training dataset. This means that the airfoils generated should exhibit the same thickness, curvature, camber, etc. as present in the PDLM's training dataset. The requirements of similar airfoil distribution, covariance, and global and local structure are thus the main requirements that the airfoil generator needs to adhere to. However, in Table 7.1 three additional design requirements are highlighted, along with their motivation.

Table 7.1: Design principles for the airfoil shape generation module

Design Consideration	Discussion	Motivation
Conditionality	The developed method should allow for airfoil generation conditioned on a specific class/ shape	To allow for more control in airfoil shape generation in the cross-country optimisation phase
Efficient	The method should enable near real-time airfoil shape generation and hence be computationally inexpensive	To allow for quick iteration, evaluation, and regeneration in the optimisation phase
Practical	The method should be subject to the generation of practical airfoils	In the optimisation phase the generation of impractical airfoils should be minimised in an attempt to reduce unnecessary pipeline executions

For this task, a BGMM is chosen. Specifically, the BGMM architecture is chosen as it holds the following relevant advantages:

1. Flexibility: BGMMs are very flexible models that can capture complex data distributions. They allow for a mixture of Gaussians to be fitted to the data, allowing for multiple underlying sub-populations with different means and variances.

2. Uncertainty estimation: BGMMs provide a probabilistic framework for modelling uncertainty in the clustering. They can provide estimates of the uncertainty in the number of clusters, cluster assignments, and the parameters of the mixture model.

3. Automatic model selection: BGMMs can automatically determine the number of

clusters in the data, which is a major advantage over traditional clustering methods that require manual specification of the number of clusters. This is especially advantageous in this context seeing that prior knowledge on the number of clusters inherent in the airfoil shape dataset is not known.

4. Robustness: BGMMs are robust to outliers, as they can model outliers as a separate cluster with a high variance.

5. Scalability: BGMMs can be scaled to handle large datasets by using sampling-based inference algorithms such as Markov Chain Monte Carlo or Variational Inference.

The BGMM is trained to separate the mined airfoil corpus into distinct clusters, or classes. If trained correctly, these classes will effectively separate the airfoils so that airfoils within the same class share similar shape and structural features whilst these attributes differ significantly from one class to another. The trained BGMM can then be used as a generative model where a latent vector is supplied, together with a specific cluster, or class number, to generate the 23 parsimonious airfoil shape and structural features. This means a BGMM has the added advantage of conditional airfoil generation which may enable class-specific optimisation if required in the end-to-end sailplane cross-country performance optimisation phase.

7.3.2 Approach

When using a BGMM to perform clustering there are five hyperparameters that are important. Optimising these hyperparameters directly influences the model's accuracy and hence the quality of the clusters. It also ensures that the fine-grained structure inherent in the data is modelled accurately. These five BGMM hyperparameters are shown in Table 7.2:

Table 7.2: Hyperparameters for Bayesian Gaussian Mixture Models

Hyperparameter	Description	Options
Number of mixture components	Depending on the data and the weight concentration prior, the model may choose not to utilise all the components by assigning weights close to zero to some components. This can result in the effective number of components used by the model being smaller than the total number of components specified.	Integer > 0
Covariance type	A parameter describing the type of covariance parameters to be used.	Full: each component has its own covariance matrix. Tied: all components share the same covariance matrix. Diagonal: each component has its own diagonal covariance matrix. Spherical: each component has its own single variance.
Weight concentration prior type	A parameter describing the type of weight concentration prior to be used.	Dirichlet process: using the Stick-breaking representation. Dirichlet distribution: may favour a more uniform weight distribution.
Weight concentration prior	The Dirichlet concentration of each component on the weight distribution. A greater concentration parameter results in more mass being allocated toward the center of the distribution, thereby activating more components. On the other hand, a lower concentration parameter causes more mass to be situated towards the edge of the mixture weights simplex. It is important to note that the concentration parameter value must be above 0.	Real > 0
Mean precision prior	The precision prior on the mean distribution (Gaussian). Controls the extent to which means can be placed. Larger values concentrate the cluster means around the mean prior. The value of the parameter must be greater than 0.	Real > 0

7.3.3 Model building

For the development of this model, the dataset discussed in Section 7.2 is used. However, seeing that we want to model only the parsimonious variables, the 160 response variables representing the higher-dimensional airfoil profile are excluded from this dataset. This dataset, therefore, has 29[2] input parameters (shape, structural, and SPM coefficients) and 581 725 samples. For this model, it is feasible to include the SPM coefficients seeing that, if the underlying structure and covariance of the features are modelled accurately, then any airfoil generated with the BGMM in terms of its 17 shape and 6 structural variables, would have the same profile when generated with the corresponding predicted SPM coefficients. This also means that additional information is added to the dataset which may aid in the effective separation of the numerous airfoil profile clusters.

In order to find the BGMM that best models, and therefore classifies, the airfoil data, its five model hyperparameters are adjusted with the help of a random search strategy, with model likelihood used as the objective function. For model building, the Scikit-learn[3] implementation of a BGMM is used. A total of 500 random search experiments are executed and in each experiment, the BGMM hyperparameters are allowed to randomly vary between the following bounds:

1. $2 \leq$ number of components ≤ 65 ,
2. covariance type: random choice between *full*, *tied*, *diag*, and *spherical*.
3. weight concentration prior type: random choice between *Dirichlet process*, and *Dirichlet distribution*, and
4. $0.01 \leq$ weight concentration prior $\leq 5\ 000$.

[2]Seeing that the dataset used here has 29 input variables, the resulting latent vector used to generate an airfoil will have the same dimension. An extra variable i.e. the class variable can then be used with this 29-dimensional column vector to generate an airfoil with desirable properties in terms of its parsimonious definition.

[3]Scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.mixture BayesianGaussianMixture.html

Mlflow[4] is employed for experiment tracking and to monitor and compare model results and performance. For reproducibility, the random seed is fixed at 87 for all experiments. However, the parameters for the Gaussian distributions are allowed to be initialised randomly via the K-means algorithm. In each experiment, the solution is initialised five times, here convergence is typically found after 2 or 3 initialisations but to guarantee convergence this number is fixed at five.

The results of the top 25 models resulting from the random search experiments can be seen in Table 7.3. From this table, it can be observed that the experiments with between 25-60 components and a weight concentration prior setting of between 1 000 and 2 000 yield the highest Per Sample Average Log Likelihood (PSALL). Also from this table, it can be seen that the five best-performing BGMM models have a PSALL above 84 – which is significantly higher than the 20 remaining experiments. All five of these models have a weight concentration prior higher than 1 000, use a full covariance type, and make use of a Dirichlet process. Also, all five of the top-performing models have more than 25 components meaning at least 25 clusters are required to effectively separate the data. It is also interesting to note that the top 25 models either make use of a full, tied, or diagonal covariance type and that spherical covariance models perform poorly in comparison.

[4]Mlflow: https://www.mlflow.org/docs/latest/index.html

Table 7.3: The top 25 results from the Bayesian Gaussian Mixture Model optimisation, sorted by Per Sample Average Log Likelihood

Model parameters						Performance metrics	
Components	Converged	Covariance type	Init. method	Weight conc. prior	Weight conc. prior type	Likelihood lower bound	PSALL
57	True	full	random	1572	dirichlet process	3.23×10^{7}	89.35
50	True	full	random	1000	dirichlet process	3.21×10^{7}	88.58
50	False	full	kmeans	2000	dirichlet process	3.21×10^{7}	88.58
42	True	full	kmeans	1403	dirichlet process	3.19×10^{7}	87.69
27	False	full	kmeans	1484	dirichlet process	3.10×10^{7}	84.05
9	False	full	random	7	dirichlet distr.	2.80×10^{7}	72.78
8	False	full	random	1	dirichlet distr.	2.79×10^{7}	72.49
8	False	full	random	6	dirichlet process	2.79×10^{7}	72.49
4	True	full	kmeans	1617	dirichlet distr.	2.59×10^{7}	65.37
2	True	full	random	2	dirichlet process	2.34×10^{7}	56.24
46	True	tied	kmeans	1675	dirichlet process	2.28×10^{7}	53.5
36	True	tied	random	3029	dirichlet distr.	2.26×10^{7}	52.47
23	True	tied	random	891	dirichlet distr.	2.21×10^{7}	51.03
20	False	tied	random	1192	dirichlet process	2.20×10^{7}	50.69
17	True	tied	kmeans	193	dirichlet distr.	2.19×10^{7}	50.41
10	False	tied	kmeans	1	dirichlet process	2.12×10^{7}	47.79
5	True	tied	kmeans	1204	dirichlet distr.	2.07×10^{7}	46.19
4	True	tied	kmeans	286	dirichlet process	2.05×10^{7}	45.5
6	True	tied	random	6	dirichlet distr.	2.00×10^{7}	43.79
6	True	tied	random	8	dirichlet distr.	2.00×10^{7}	43.79
1	True	full	kmeans	1371	dirichlet distr.	2.00×10^{7}	43.8
48	True	diag	random	233	dirichlet distr.	1.79×10^{7}	36.56
47	False	diag	kmeans	68	dirichlet process	1.79×10^{7}	36.22
38	True	diag	random	1452	dirichlet process	1.76×10^{7}	35.26
7	False	diag	random	9	dirichlet distr.	1.45×10^{7}	24.29

Of the top five best-performing models, only three models converged, i.e. the models with 57, 50, and 42 components. There are two models with 50 components, and both have the same PSALL values but the model with the lower weight concentration prior is the only one to converge. Seeing that the PSALL of the top two converged models are very similar i.e. a difference of 0.77, the simpler model is selected. That is, the selected final BGMM for use in the airfoil generator module is the one with the model parameters as depicted in Table 7.4:

Table 7.4: Hyperparameters of the optimal BGMM as used for the final airfoil generator module

Hyperparameter	Value
Number of components	50
Covariance type	Full
Weight concentration prior type	Dirichlet process
Weight concentration prior	1 000
Mean precision prior	0.001

7.3.4 Module integration

With the airfoil generator module developed and its accuracy verified, the next step is the integration of this module with the TSM by replacing the SPM-based airfoil generator module (see Figure 1.2). With the integration of this module, the updated workflow for generating an airfoil is as described in Algorithm 4 and in Figure 7.1 the complete functionality of the new airfoil generation process is depicted. Here it is observed that a latent vector and an airfoil class specification is used in the BGMM to produce an airfoil in terms of its lower order parsimonious variable representation.

Algorithm 4 Generating Shape and Structural Parameters from Latent Variables

Require: $\mathbf{x} \in [-1, 1]^{29\times1}$, $y \in [0, 49]$ with $\mathbf{x}$ the latent space vector and $\mathbf{y}$ the airfoil class.

Ensure: $\mathbf{s} \in \mathbb{R}^{17\times1}$, $\mathbf{t} \in \mathbb{R}^{6\times1}$

1: Use y to obtain the corresponding covariance matrix Σ_y and mean matrix μ_y describing a specific cluster

2: Use $\mathbf{x}$, Σ_y, and μ_y to generate a vector $\mathbf{z} \in \mathbb{R}^{23\times1}$ of 17 shape and 6 structural variables using the BGMM

3: Inverse transform $\mathbf{z}$ to obtain the actual shape parameter values, denoted by $\mathbf{s} \in \mathbb{R}^{17\times1}$, and the actual structural parameter values, denoted by $\mathbf{t} \in \mathbb{R}^{6\times1}$

4: **return** $\mathbf{s}$, $\mathbf{t}$

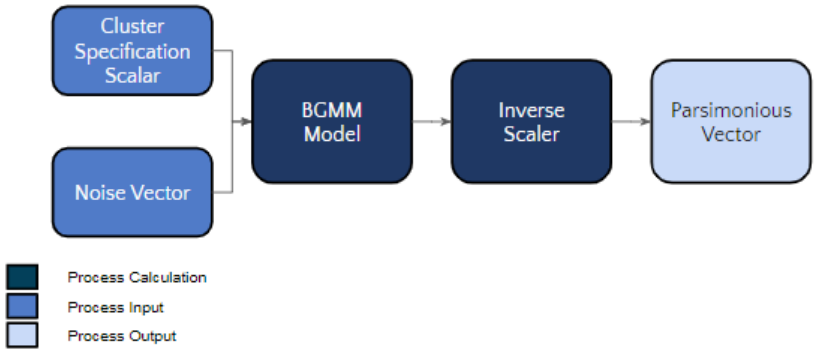


Figure 7.1: Depiction of the full inference process of the airfoil generator module

7.4 Model cluster evaluation

In this section, the evaluation of the trained BGMM as an airfoil generator module, is discussed. Here, the approach is to analyse the clusters, or distinct airfoil classes, found by the optimal BGMM. It is desirable to have clusters with minimal overlap as this would mean good clustering and hence facilitate the conditional airfoil generation process. However, it is important to note that having some overlap in the clusters will not be detrimental. This is because, with conditional airfoil generation, a practical airfoil shape will still result, just not with the parsimonious variables in the exact desirable range. It is however important that the model is highly accurate in separating practical and impractical airfoil profiles to ensure that only the generation of practical airfoils result when conditioned on the appropriate classes.

The cluster analysis is done in three steps: (1) analysis of the quality of the clusters, (2) analysis of the cluster overlap, and (3) analysis of the cluster statistical properties. This provides the necessary understanding of whether the metrics utilised to select the best-performing BGMM results in the creation of high-quality clusters. That is, is the model able to discern that airfoils located in the same cluster have a similar shape and structural characteristics, and vice versa.

7.4.1 Cluster quality analysis

The quality of the clusters generated by the BGMM is first analysed. This is done by employing the t-SNE algorithm (van der Maaten & Hinton, 2008). The t-SNE is a technique used for data visualisation and dimensionality reduction. It can be used to visualise high-dimensional data in a two- or three-dimensional space. Therefore, when the t-SNE is used to visualise clustered data, it can give insight into the quality of the clusters.

For this analysis, the hold-out airfoil dataset is used. The feature values for this dataset are scaled between zero and one by using the scaling metrics of the training dataset. The BGMM is then used to predict the cluster to which each airfoil in the hold-out dataset belongs. From here, the t-SNE algorithm is used with a perplexity setting of 500[5], to project the data to a lower dimensional (3D) representation. Finally, the individual projected data points are colour-coded according to the cluster to which they belong and the point clouds analysed.

Slices of the 3D analysis can be seen in Figure C.1 It is observed from this figure that there is an overlap between some classes, although this happens infrequently. This indicates that there are possible misclassifications or different classes with similar parsimonious variable characteristics. It can also be seen from this figure that classes with similar profiles are closer in proximity to one another in higher dimensional space. Specifically, it is observed that classes 1 and 5 are similar, classes 30 and 34 are similar, classes 27, 39, and 45 are similar, and classes 32 and 48 are similar. Although they have similar profiles, it does not necessarily mean these classes share exactly the same shape and structural characteristics and hence this does not directly indicate misclassification.

[5]The primary factor governing the fitting process in t-SNE is known as perplexity (van der Maaten & Hinton, 2008). Perplexity is a parameter that approximates the number of nearest neighbours taken into account when aligning the initial and fitted probability distributions for each data point. Lower perplexity values indicate an emphasis on local relationships and a focus on immediate neighbouring data points. Conversely, higher perplexity values favour a broader perspective by taking into account a greater number of data points in the analysis. Here a high perplexity value is chosen as the global structure is more important to model than the local structure. A value of 500 is chosen as it is the highest value that could be used due to the computational complexity and the size of the dataset.

Overall, the t-SNE analysis indicates that, although some classes predicted by the airfoil generator are pure,[6] there is overlap present between classes. This property of the airfoil generator is explored in the following sections.

7.4.2 Cluster overlap analysis

As seen in the previous section, some overlap exists between the clusters generated by the airfoil generator which indicates either misclassification or classes with similar, but not exactly the same, profiles. Further analysis is therefore done to understand this overlap. This is done by analysing the Silhouette coefficients (Rousseeuw, 1987) of each cluster, as seen in Figure 7.2. Silhouette coefficients near *+1* indicate that a sample is far away from the neighbouring clusters. A value of 0 indicates that a sample is on, or very close to, the decision boundary between two neighbouring clusters. Negative values indicate that a sample might be assigned to the wrong cluster.

From the silhouette coefficient analysis, the following is observed:

1. The airfoil generator yields clusters with fairly consistent sizes and shapes.
2. There are classes with silhouette coefficients between zero and minus one, confirming the t-SNE overlap or similarity findings.
3. The positive average silhouette coefficient (represented by the red dotted line in Figure 7.2) indicates that most of the classes are located away from their neighbouring clusters.
4. The clusters with the most overlap are 3, 16, 30, 33, and 46. These clusters are also noted to have very sharp curvature at the TE and therefore have a high level of impractical airfoils present. This means that these clusters are less likely to be used for conditional airfoil generation and a higher level of overlap or misclassification here is acceptable.

[6]Indicated by a given cluster located sufficiently far away from neighbouring clusters

5. The highest quality clusters are 0, 4, 10, 17, 22, 29, 43, and 47. From an analysis of these clusters, it can be observed that they have practical airfoil shapes with the majority of the clusters having desirable top surface thickness, bottom surface thickness, and camber distributions. Therefore having high-quality clusters here are desirable as it means conditional generation with these respective clusters will yield practical airfoil shapes with airfoil profiles applicable for sailplane cross-country flight.

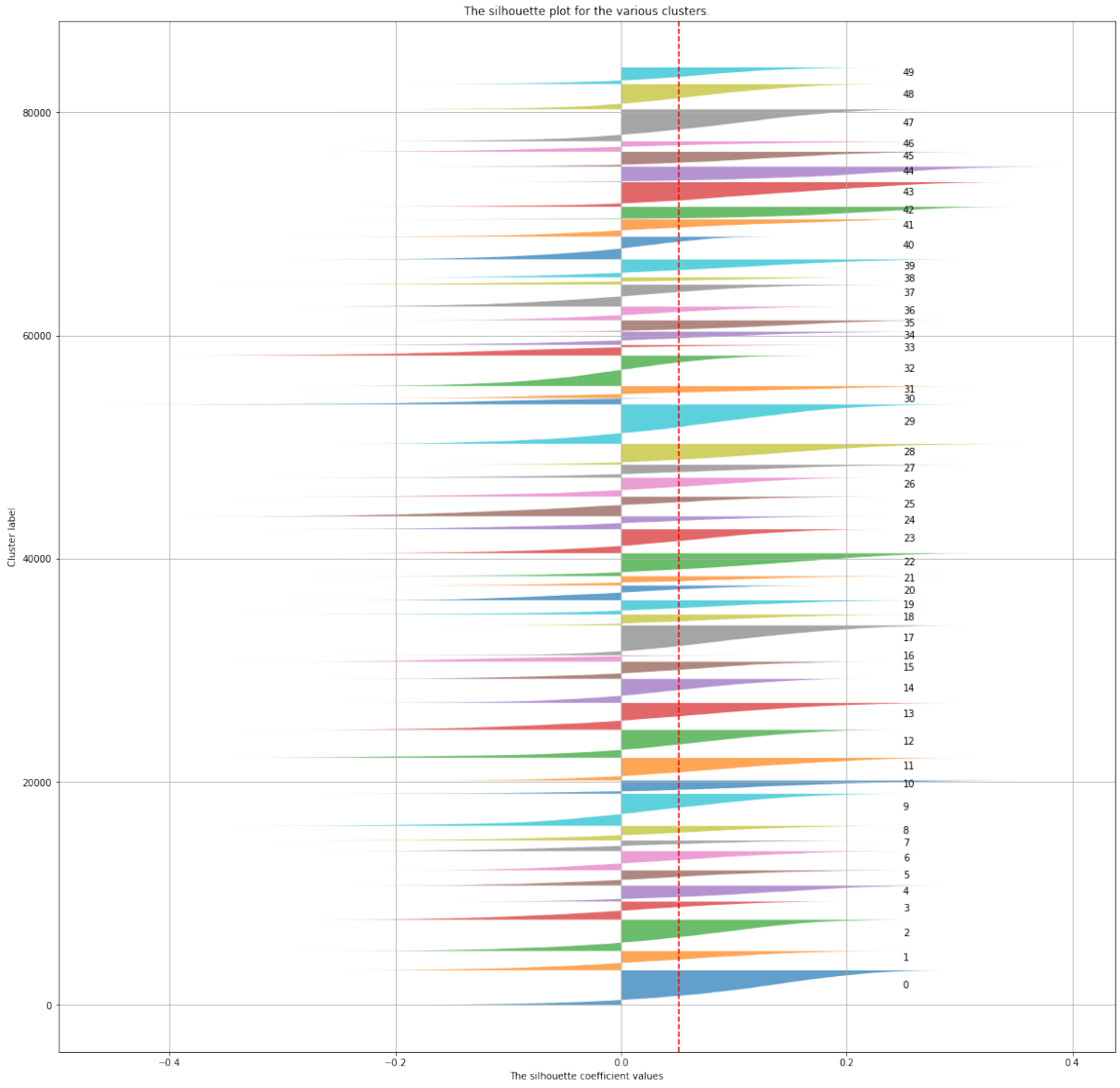


Figure 7.2: Silhouette coefficients for 50 classes as determined with the airfoil generator module

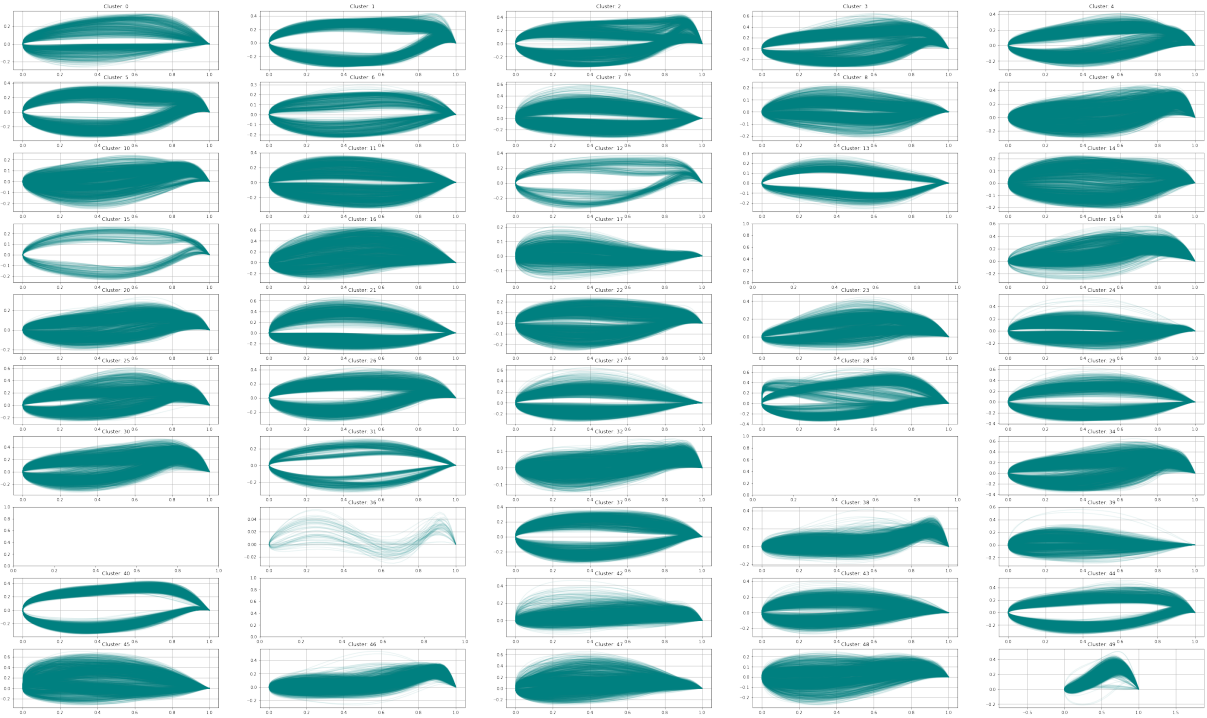


Figure 7.3: Comparing 1 000 random airfoil samples for all 50 classes, as determined by the airfoil generator module

Visually inspecting the worst-performing clusters, namely clusters 3, 16, 30, 33, and 46, involves randomly sampling and generating a total of 1 000 airfoils for each of the 50 classes from their respective SPM coefficients. These generated airfoils, grouped by class, are then overlay plotted as seen in Figure 7.3. From this figure, it can be seen that the airfoils within the same cluster tend to have the same shape characteristics and general airfoil profile. It is also clear that the airfoil generator regards some classes as insignificant and therefore does not assign many data points to them.[7] These are classes 18, 33, 35, and 41, and are excluded from this quantitative analysis. In this figure, it also is observed that classes 1 and 5, classes 30 and 34, classes 27, 39, and 45, and classes 32 and 48 share similar profiles respectively, confirming the t-SNE findings.

From visual inspection of Figure 7.3, it can be seen that there is cluster overlap present in classes 2, 3, 9, 16, 19, 20, 23, 24-25, 27-28, 30, 32, 34, 38-39, 42, 45, and 46-49 and that

[7]This is due to the Bayesian component of the GMM where the model minimises the number of classes, or components until an acceptable number is found that maximises the objective function.

the majority of these classes have impractical airfoil shapes, indicating that conditional generation with these respective clusters should be avoided. Analysis of this figure also shows that the high-quality clusters with low levels of overlap and practical airfoil shapes are classes 0, 6-8, 11, 13, 21-22, 26-27, 29, 31, 37, 40, and 43-44. This visual analysis, therefore, confirms the findings of the silhouette coefficient and t-SNE analysis. From this analysis it is therefore concluded that the BGMM accurately separates practical and impractical airfoil shapes with airfoils in the same cluster sharing the same profile characteristics – thereby facilitating the generation of practical airfoil shapes conditioned on desirable airfoil shape and structural characteristics.

7.4.3 Statistical property analysis

The final step in the cluster analysis is to inspect the statistical properties of the shape and structural variables, for all the data samples, grouped by class. The objective here is twofold: (1) to establish if there is a significant variance between the mean, min, max, standard deviation, 25th, 50th, and 75th percentile of each parsimonious feature of each cluster, and (2) to discern which clusters have the largest differences amongst these descriptive statistics.

To do this, first, the descriptive statistics are calculated for each of the 23 parsimonious variables and then grouped by class. The results from this analysis can be seen in Figure 7.4. From this analysis, the following is observed:

1. For each parsimonious feature, the standard deviation statistics have the least variance between classes.

2. The largest variance by cluster is seen for the maximum and minimum values of the parsimonious features. This indicates that there is a distinguishable difference between the minimum and maximum parsimonious feature values when comparing the clusters.

3. The overall variance across the x-centroid, the nose radius, the bottom curvature,

and the perimeter is the lowest compared to the other parsimonious variables. This indicates that these features are not as important when it comes to cluster separation.

4. The maximum camber magnitude and top magnitude as well as the maximum trailing edge angle have the most variance across the clusters. This indicates that these features are important in the model's class separation decisions.

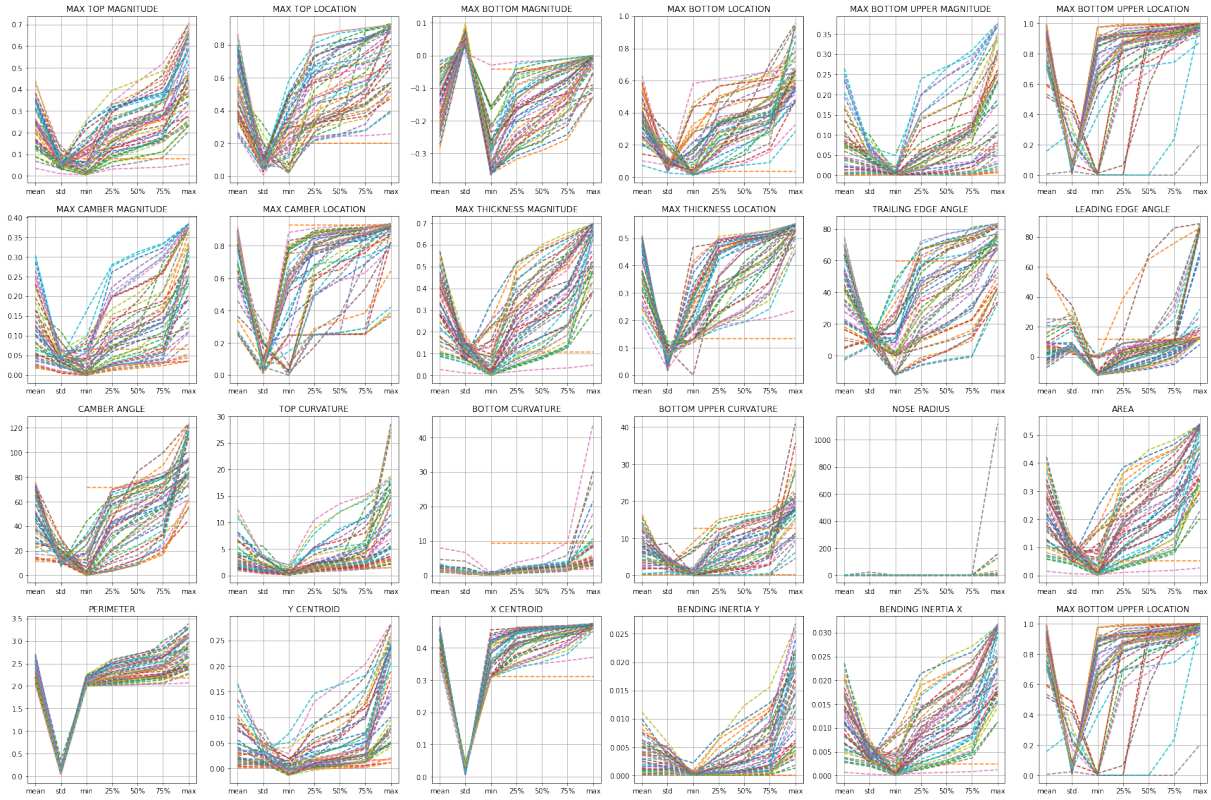


Figure 7.4: Descriptive statistics for all parsimonious features grouped by cluster number

To gain a deeper understanding of the parsimonious feature importance for cluster separation, the following approach is taken to calculate the normalised average difference between the descriptive statistics of all airfoil classes:

1. Normalise the parsimonious feature variables.[8]

[8]This facilitated the comparison on a feature-by-features basis.

2. Calculate the descriptive statistics for each normalised parsimonious feature.[9]

3. Calculate the average difference between the value of a specific class's descriptive statistic and all other class values. This is done using the following equations:

   Let S be a set of n elements, $x_i \in S$ for $i = 1, 2, \ldots, n$. We define the average difference between an element x_i and all other elements in S as:

   $$z = \frac{1}{n-1} \sum_{j \neq i} |x_i - x_j| \tag{7.1}$$

   Then, the average difference for each set of elements $S_1, S_2, \ldots, S_k$ is given by:

   $$\sum_{i=1}^{k} |z_i| \tag{7.2}$$

The results of this analysis can be seen in Table 7.5 where the normalised average distance between the six descriptive statistics across all 50 airfoil classes is depicted. Here it is again observed that the maximum and minimum values for a given parsimonious variable receive high importance in the clustering decision. Specifically, the maximum values of (1) the maximum bottom magnitude, (2) the maximum camber magnitude, (3) the trailing edge angle, and (4) the bending inertia about the y-y axis have the most significant difference across the 50 clusters. For the minimum parsimonious feature values: (1) the maximum thickness location, (2) the leading edge angle, and (3) the bending inertia about the x-x axis have the biggest variance across the 50 clusters. The features that have the least variance across the classes and hence are not as important in the cluster separation decision are: (1) the area, (2) the maximum bottom location, and (3) the nose radius.

Overall these findings are aligned with those from the inspection of Figure 7.4. Also, here a large variance is seen across the majority of the descriptive statistics of the parsimonious features, of each class. These findings are further emphasised in Figure C.2, where the separation by class for various parsimonious shape and structural characteristics are highlighted. If this was not the case, it would have meant that each cluster had similar values for the same parsimonious features. However, because a large variance is observed, it can be concluded that the classes have different shape and structural characteristics

[9]The results from this calculation can be seen in Table C.1.

when comparing one cluster to another. This means that the airfoil generator is effective in separating airfoil profiles and hence is accurate in its clustering approach. This further implicitly validates the model's efficacy in separating practical and impractical classes as also found in Sections 7.4.1 and 7.4.2.

Table 7.5: Normalised average difference between the six descriptive statistics of all airfoil classes

Feature/Metric	25th percentile	75th percentile	Maximum	Mean	Median	Min.	Std.dev
Max top magnitude	0.216	0.260	0.252	0.228	0.248	0.122	0.055
Max top location	0.194	0.242	0.270	0.210	0.226	0.109	0.054
Max bottom magnitude	0.069	0.173	0.326	0.122	0.118	0.014	0.070
Max bottom location	0.014	0.026	0.147	0.022	0.017	0.009	0.019
Max bottom upper magnitude	0.129	0.170	0.230	0.144	0.157	0.055	0.051
Max bottom upper location	0.202	0.198	0.198	0.187	0.209	0.120	0.051
Max camber magnitude	0.087	0.137	0.307	0.119	0.112	0.045	0.080
Max camber location	0.179	0.172	0.187	0.157	0.173	0.168	0.067
Max thickness magnitude	0.265	0.206	0.112	0.226	0.247	0.239	0.059
Max thickness location	0.275	0.120	0.066	0.184	0.184	0.370	0.122
Trailing edge angle	0.170	0.272	0.400	0.218	0.220	0.031	0.080
Leading edge angle	0.274	0.222	0.141	0.243	0.258	0.341	0.078
Camber angle	0.219	0.293	0.340	0.250	0.252	0.111	0.061
Top curvature	0.233	0.157	0.075	0.185	0.202	0.197	0.058
Bottom curvature	0.235	0.277	0.227	0.244	0.266	0.133	0.056
Bottom upper curvature	0.233	0.249	0.226	0.226	0.246	0.198	0.069
Nose radius	0.154	0.178	0.265	0.162	0.164	0.116	0.040
Area	0.000	0.000	0.069	0.000	0.000	0.000	0.003
Perimeter	0.144	0.205	0.291	0.169	0.171	0.072	0.050
Y centroid	0.077	0.147	0.269	0.109	0.108	0.022	0.051
X centroid	0.287	0.282	0.198	0.275	0.293	0.203	0.050
Bending inertia y	0.267	0.153	0.067	0.204	0.209	0.341	0.080
Bending inertia x	0.113	0.196	0.335	0.154	0.144	0.062	0.063

7.5 Model generative capability evaluation

The airfoil generator is demonstrated to have the required capability to separate clusters and, consequently, differentiate between airfoil profiles. Therefore the last stage of the model evaluation process involves testing and confirming the model's ability to generate new, accurate airfoil designs.

This is achieved by using the model to generate 120 000 unique samples. This airfoil generation process can be summarised as follows: The first step is to randomly choose a cluster from the trained model's mixture components. Next, the mean and covariance matrix of the chosen cluster are used to generate a new sample point using a Gaussian distribution, referred to as the latent vector (Section 7.3.4). This new sample is then added to the generated (or fake), dataset. This process is repeated until the desired number of samples has been generated. If trained correctly, by using the learned parameters of the airfoil generator module, the generated samples will share similar characteristics to those in the training dataset, while also incorporating some variability and randomness.

For comparison of the generated data with the actual data, the generated results are appropriately descaled using scaler values derived from the training dataset. The feature distribution and feature covariance of the generated dataset is then compared with that of the actual data in the hold-out set. This comparison can be seen in the following sections.

7.5.1 Means and standard deviations

The first step in the validation of the airfoil generator module's generative capacity is the comparison of the generated and actual features' means and standard deviations. The results from this analysis can be seen in Figure 7.5. If the model generates airfoil samples with the correct distributions, a high R^2 between the generated and actual feature means and standard deviations would be observed. From Figure 7.5 it can be seen that there is very close agreement between the means and standard deviations of the generated and hold-out datasets with an R^2 of 0.99. This indicates that, for the parsimonious features

and SPM coefficients, the generated dataset approximated the hold-out dataset.

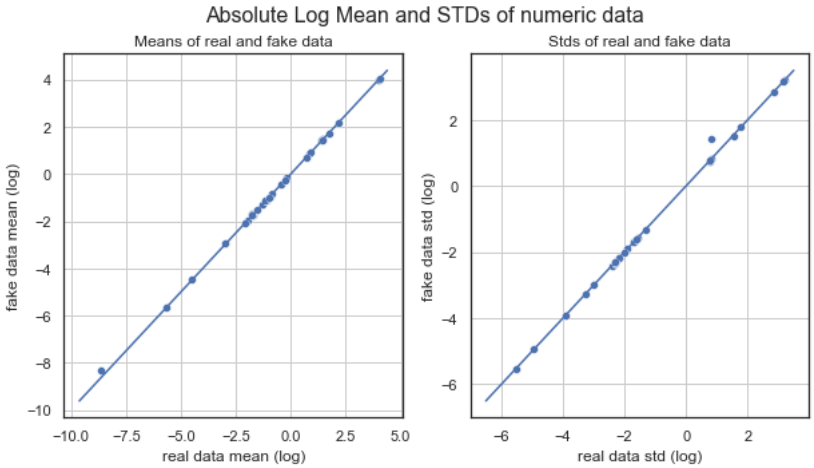
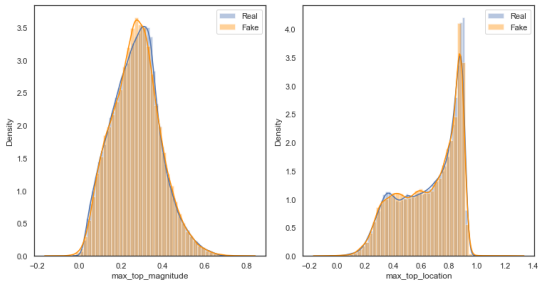


Figure 7.5: Comparison between the individual feature standard deviations and means for the generated and hold-out datasets

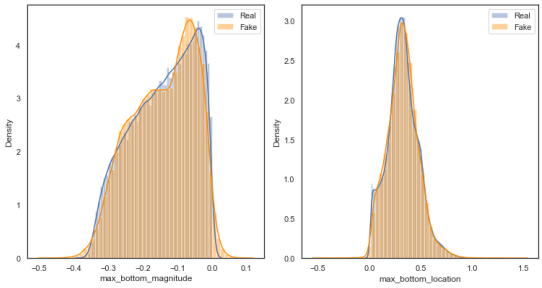
7.5.2 Feature distribution

The next step in this process is to evaluate how the distributions of the 17 shape variables for the generated and hold-out datasets compared. In Figure 7.6 the distribution of the generated and the actual data for the 17 respective shape features can be seen. A high level of correlation can particularly be seen for the maximum top and bottom positions and magnitudes. Conversely, the maximum camber magnitudes and maximum thickness magnitudes show the lowest correlation. The generated Gaussians for these features do not match the actual distribution with the same degree of accuracy as the other shape features. Nonetheless, there is still a strong correlation between the generated and actual distributions for these two features and the overall shape and scale match with an acceptable level of accuracy.

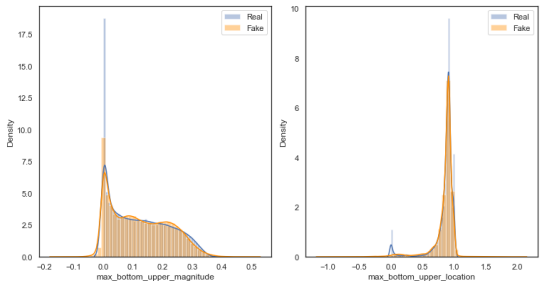
From this figure, it is therefore observed that 50 components give the airfoil generator module sufficient capability to model the underlying airfoil shape feature distributions, with the generated feature distributions closely approximating the hold-out dataset.



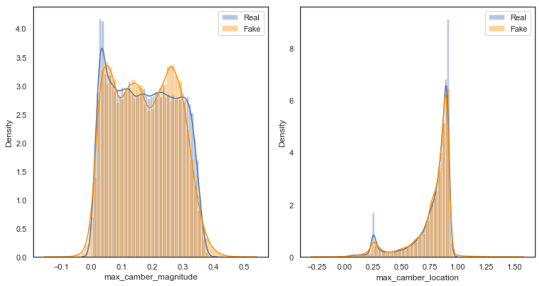
(a) Maximum top position and magnitude



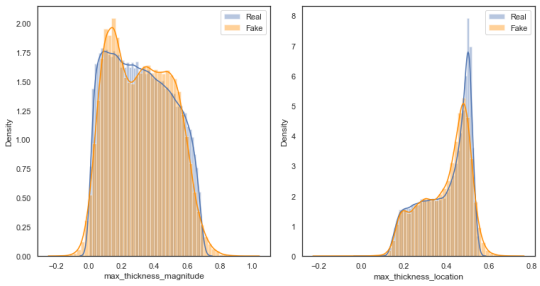
(b) Minimum bottom position and magnitude



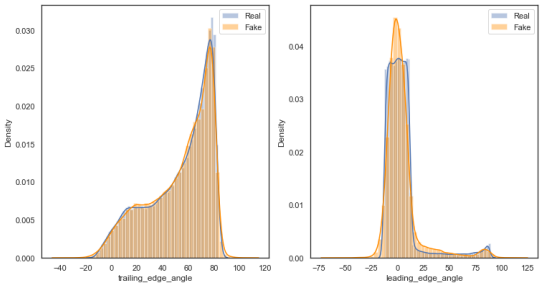
(c) Maximum bottom position and magnitude



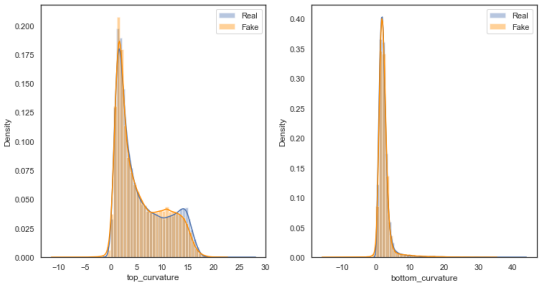
(d) Maximum camber position and magnitude



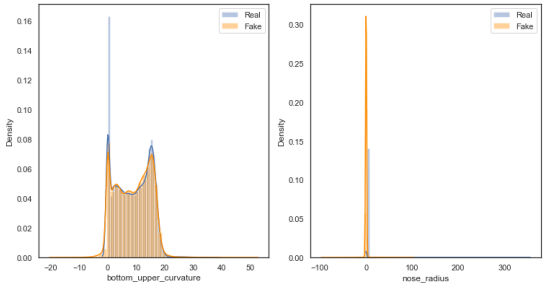
(e) Maximum thickness position and magnitude



(f) Average leading edge and trailing edge angles



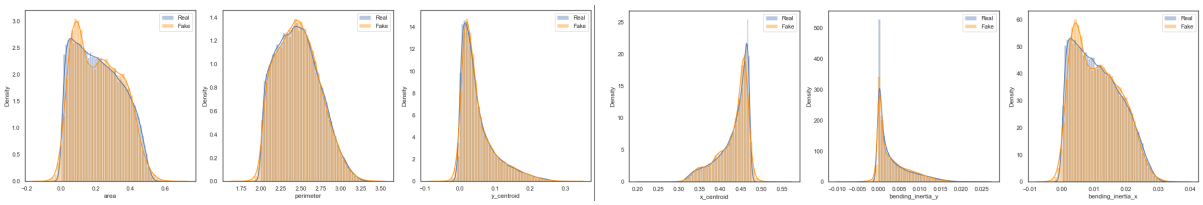
(g) Maximum top and minimum bottom curvature



(h) Maximum bottom curvature and nose radius

Figure 7.6: Airfoil parsimonious shape feature distribution for the hold-out and generated datasets

Next is to evaluate the airfoil generator module's capability to generate data with the same airfoil structural feature distribution. In Figure 7.7 the generated versus actual distribution for the six structural features can be seen. From this figure, it can be seen that, for all the structural features, the generated data distribution closely approximates the actual data. An interesting observation here is how accurately the 50 Gaussians are able to model the complex multimodal distributions of the bending inertia about the x-x and y-y axes as well as the x- and y-centroid distributions.



(a) Area, perimiter, and y-centroid. (b) x-centroid, Ixx, and Iyy.

Figure 7.7: Airfoil parsimonious structural feature distribution for the hold-out and generated datasets.

This analysis, therefore, confirms that 50 components are sufficient in modelling the underlying airfoil shape and structural data distribution, seeing that the distribution of the generated data closely resembled the distribution of the hold-out data.

7.5.3 Correlation

The final step is to analyse the correlation between the parsimonious variables of the generated and the hold-out dataset to ensure that the airfoil generator module accurately captures the input feature local structure. This is done by calculating the Pearson correlation coefficient for each of the 29 features present in the hold-out set and the generated dataset, respectively. The Pearson correlation analysis of the real and generated data is then displayed on a heat map. In order to visually inspect the difference between the Pearson correlation of the real and generated data, an additional heatmap is generated which shows the difference between the Pearson coefficients calculated for the two datasets

– see Figures 7.8–7.10. The maximum errors between the real and generated Pearson correlation coefficients are in the range of 0.001 and 0.047 (less than 2%). This difference is small enough to conclude that the airfoil generator module captures the local temporal and covariance structure between the parsimonious shape variables.

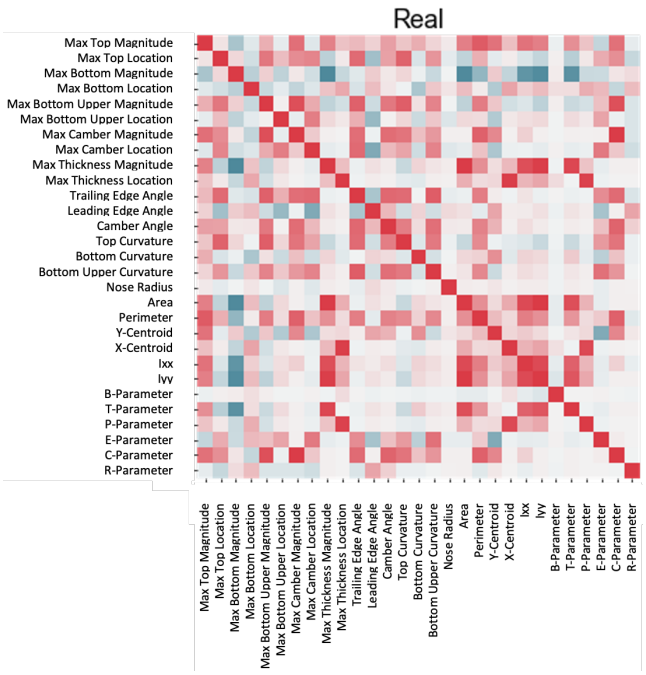


Figure 7.8: Pearson correlation for the features of the real data

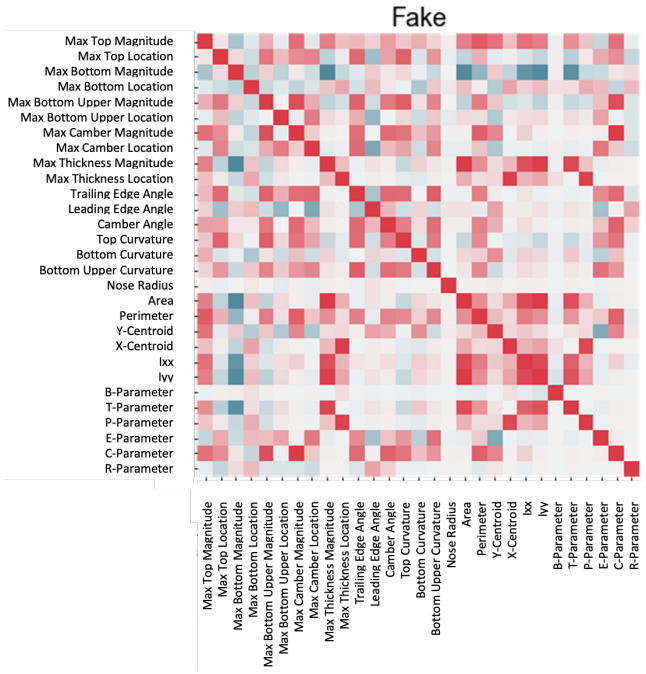


Figure 7.9: Pearson correlation for the features of the generated data

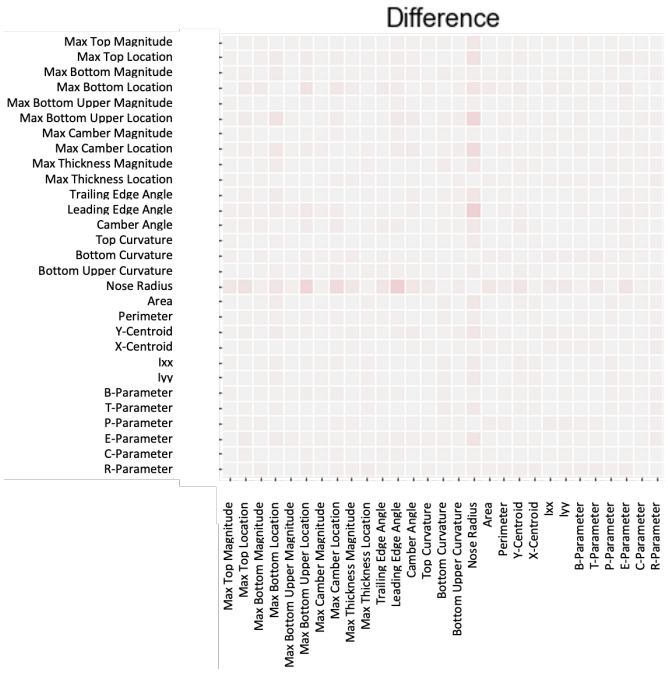


Figure 7.10: Difference between the Pearson correlation coefficients of the real and generated data

7.6 Development of the airfoil shape conversion module

With a model capable of generating realistic airfoil shapes in terms of the defined parsimonious variables, the next step is to develop a model capable of mapping parsimonious shape and structural variables to raw x and y airfoil coordinates. Given the dataset discussed in Section 7.2, it is possible to build a data-driven model to relate these two representative design spaces.

7.6.1 Overview

An important sub-objective of this research is the validation of the developed DSM results with those of the TSM. However, because the DSM defines the sailplane wing geometry with intuitive airfoil parameters at each station rather than raw x and y coordinates as in the TSM, a method is developed to enable mapping the parsimonious airfoil features to raw x and y coordinates. This is accomplished with the shape conversion module. The main purpose of this shape conversion module, therefore, is the accurate prediction of the airfoil profile in terms of raw x and y coordinates for a given input set of parsimonious variables. Stated differently, features such as nose radius, position, and location of the maximum thickness and camber should be accurately represented in the higher dimensional x and y coordinate domain given the lower dimensional parsimonious set.

The three most important design principles that are followed when developing this module are as discussed in Table 7.6. As can be seen from this table, one of the main design considerations is the mapping of data from a lower dimension to a higher dimension. Also, the capability to preserve the sequential structure when mapping between the two data domains is of great importance. Although efficient, ANNs do not inherently possess the capability to model sequential data or effectively map from a lower-order data domain to a higher-order data domain. Converting the parsimonious variables back to raw x and y coordinates analytically is also not possible. This is because each parsimonious variable

is calculated by using multiple top and bottom surface coordinates in complex matrix and integral calculations (see Section 4.4). Doing the inverse of these calculations analytically is not possible.

Table 7.6: Design principles for the airfoil shape conversion module

Design consideration	Discussion	Motivation
Sequential structure	The local and global structure of the x and y coordinates should accurately reflect the parsimonious characteristics and vice versa	To ensure that the conversion between the two data domains preserves the airfoil shape and hence represents the same airfoil profile
Dimensionality	Conversion from lower order dimension to a higher order dimension	The developed method should extend itself to easily convert lower-order parsimonious characteristics to higher-order x and y coordinates
Efficiency	Efficient conversion from a lower-order dimension to a higher-order dimension	The developed method should be able to rapidly convert between the two respective data domains to reduce any additional computational overhead

To solve this problem, the BLSTM architecture is used. This is because a BLSTM is particularly useful where there is a need to model time series or sequence data (Graves & Schmidhuber, 2005). BLSTMs, therefore, have the ability to capture sequential dependencies inherent in the data. BLSTMs also have the ability to learn from both past and future data points, making them ideal for sequence modelling tasks such as in the case of airfoil generation from parsimonious variables. The ability to process data in both directions allows for a better understanding of the context of the data. When translating a given parsimonious variable to the higher-order x and y coordinate space, having context about the data points to the left and right of a specific critical location, therefore, improves the model's sequence modelling capacity.

The development and validation of the airfoil shape conversion module with the use of a BLSTM can be seen in the following sections.

7.6.2 Approach

As mentioned earlier in this section, the dataset discussed in Section 7.2 is used to develop the BLSTM. The airfoil shape dataset is split into three partitions (training, validation, and testing sets) using random sampling without replacement. The training set consists of 70% of the 581 725 airfoil samples, the testing set of 20%, and the validation set of 10% of the samples. Of the 29 input parameters discussed in Section 7.2, the six global SPM parameters are discarded seeing that the aim is to generate the 80 x and y coordinates from the 23 parsimonious parameters.

Ensuring uniform sequential structure for each sample, the output data is ordered as follows: (1) the 80 x and 80 y coordinates are divided into 40 top and 40 bottom coordinates respectively; (2) the 40 top x and y coordinates are arranged from trailing edge to leading edge; and (3) the 40 bottom x and y coordinates are arranged from leading edge to trailing edge. This airfoil profile coordinate arrangement is illustrated in Figure 7.11. The final structure of each sample, therefore, consists of a 23-dimensional vector mapped to a 160-dimensional vector.

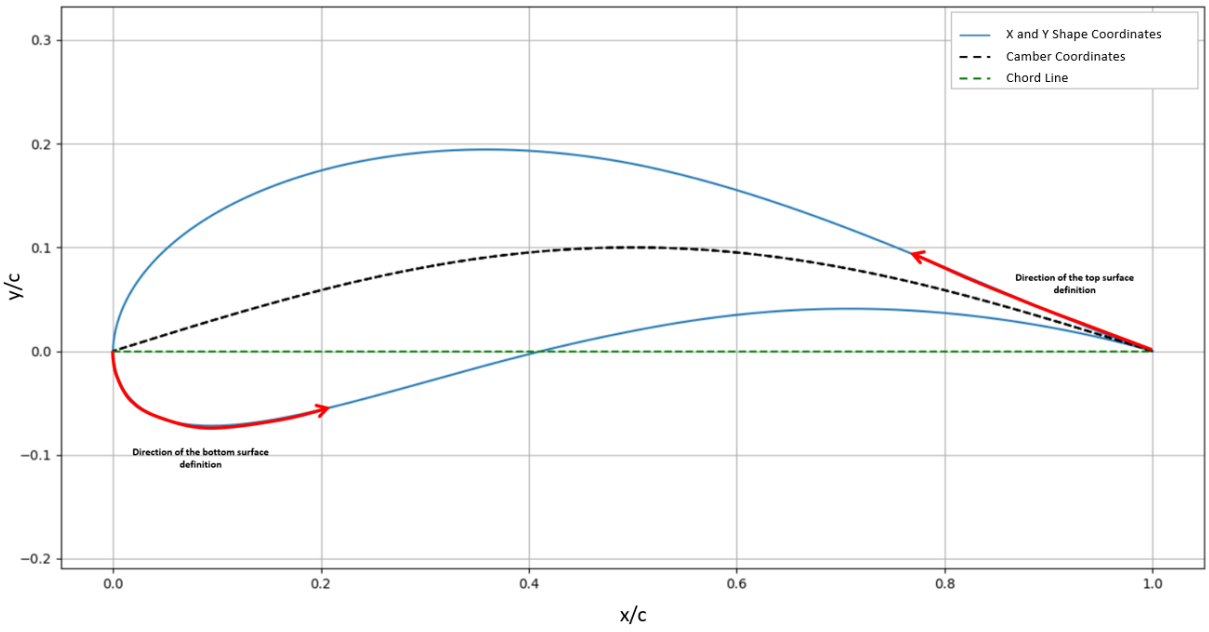


Figure 7.11: Depiction of the airfoil profile definition in terms of the top and bottom surface x and y coordinates

7.6.3 Model building

To find the best model architecture and set of hyperparameters, a two-step approach is followed. First, a random search strategy is employed. In this initial random search, course bounds are set on the chosen hyperparameters to find a feasible range in the set which results in good-performing models – based on their relative validation MSE values. Once the range and combination of hyperparameters that results in well-performing models are known, a grid search strategy is employed within the refined hyperparameter boundaries.[10]

For the initial hyperparameter search, in each random search run, the model is allowed to iterate until convergence is reached i.e. the validation error converges within tolerance. The hyperparameters that are adjusted at the start of each new random search run, as well as their corresponding upper boundaries and lower boundaries, can be seen in Table 7.7. In addition to this, the random search is set up in such a way that, in each new run, the hidden layers have an equal opportunity to be either BLSTM or vanilla LSTM (Sherstinsky, 2020) layers.

Network parameters are optimised using the Adam (Kingma & Ba, 2014) algorithm, which is a stochastic gradient descent method that computes individual adaptive learning rates for different parameters from estimates of first- and second-order moments of the gradients. For recurrent activation, the Sigmoid function is used and for dense layer activation, the Tanh activation is used, as is standard in LSTM networks (Sherstinsky, 2020). Finally, MSE is used as the loss function; that is, the MSE between the actual and generated x and y coordinates are calculated for each sample in the mini-batch and used by the Adam algorithm to determine the direction of the parameter updates.

[10] All models are built using Tensorflow (Abadi et al., 2016).

Table 7.7: Initial search space for mapping parsimonious shape variables to raw x and y coordinates.

Parameter	Lower boundary	Upper boundary
Number of hidden layers	1	5
Units per layer	1	100
Batch size	8	4 096
RNN dropout probability	0	0.5
Learning rate	0.0001	0.1
Dropout probability	0	0.5

In total, 30 random search experiments are performed and from these runs, the following conclusions are made:

1. Models with bidirectional LSTM layers have overall better performance than models with vanilla LSTM stacked layers.

2. Models with a high number of samples per batch and a high number of stacked layers perform poorly, regardless of the learning rate or the number of dense units present.

3. Models with fewer stacked layers and a high number of dense units per layer tend to have better performance than deep models with fewer dense units per layer.

4. As expected, batch size and learning rate have a large effect on the validation accuracy, with favourable results observed when low batch sizes are used in combination with low learning rates. Models with a high learning rate ($\geq$0.1) performed poorly in general.

From the analysis of the first 30 random search experiments, the relevant hyperparameter upper and lower bounds are adjusted and a further 20 experiments are run. Of these 20 experiments only 18 converged: the hyperparameters and performance of the best 10 models are shown in Table 7.8.

Table 7.8: Best models obtained during hyperparameter optimisation, ordered according to validation MSE

Batch size	Bi directional hidden layer(s)	Dense dropout	Learning rate	Recurrent dropout	Total layers	Units per layer	Training MSE	Valid MSE
8	True	0.2	1×10^{-3}	0.3	3	90	1.5×10^{-5}	6.2×10^{-6}
8	True	0.3	1×10^{-3}	0.1	4	90	1.8×10^{-5}	7.8×10^{-6}
64	True	0	1×10^{-2}	0.4	3	60	9.1×10^{-6}	8.8×10^{-6}
32	True	0.1	1×10^{-2}	0	3	80	1.4×10^{-5}	9.3×10^{-6}
8	True	0.4	1×10^{-3}	0.5	3	50	2.6×10^{-5}	1.1×10^{-5}
16	False	0.4	1×10^{-3}	0.5	3	100	2.7×10^{-5}	1.2×10^{-5}
32	False	0	1×10^{-3}	0.3	5	70	1.9×10^{-5}	1.2×10^{-5}
64	True	0	1×10^{-2}	0.3	4	30	1.3×10^{-5}	1.4×10^{-5}
32	False	0.1	1×10^{-3}	0.4	2	80	2.2×10^{-5}	1.5×10^{-5}
64	True	0	1×10^{-2}	0.2	4	40	9.9×10^{-6}	1.6×10^{-5}

In general, all models produce very low MSE values (good performance) with lower batch sizes and learning rates combined with a high number of dense units resulting in better-performing models. It is also clear that shallow models, i.e. fewer stacked layers with BLSTM units in the hidden layers, result in overall lower MSE on the validation set. The best-performing model has the following architecture: one standard LSTM layer with 90 dense units, one bidirectional LSTM layer with 180 dense units, one standard LSTM layer with 90 dense units, and one output layer with 160 dense units.

7.6.4 Model evaluation

In this section, the accuracy of the BLSTM with the lowest validation MSE is discussed. This accuracy is determined by analysing the distribution of the MAE, calculated between the actual and predicted x and y coordinates for each airfoil in the hold-out dataset, as seen in Figure 7.13. From an analysis of the distribution of the MAE's descriptive statistics, it is clear that for most of the samples, the difference is negligible with an average testing set MAE of 0.0038 compared to a validation MAE of 0.0025. This negligible difference is also demonstrated in Figure 7.12, which displays a random sample of the real versus BLSTM-predicted airfoil shapes from the testing dataset.

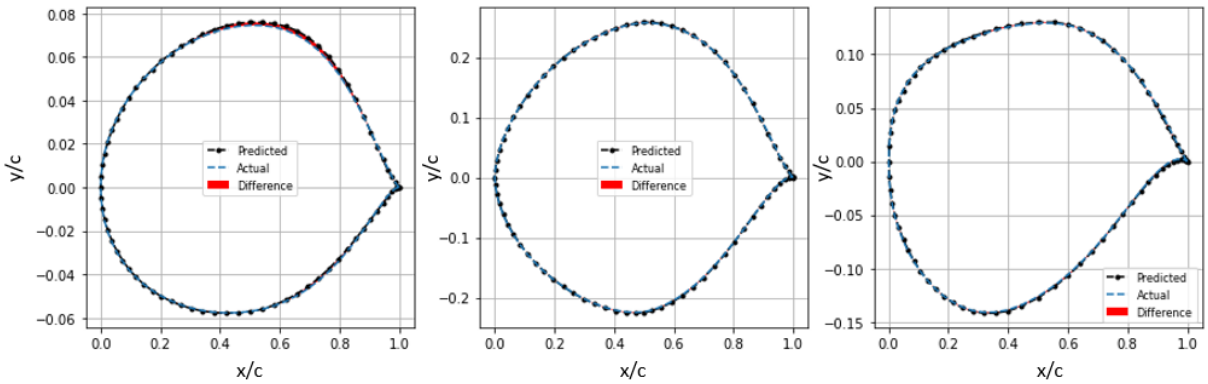


Figure 7.12: Example of BLSTM fitting error on practical airfoil shapes (MAE of 0.0031 compared to an average MAE on the test set of 0.0038)

From analysing the MAE distribution in Figure 7.13 there is, however, evidence of outliers which results in a skewed MAE distribution for the testing set. The long tail to the left indicates instances of highly accurate predictions whereas the short tail to the right indicates instances of highly inaccurate predictions. These highly inaccurate prediction outliers correspond to cases where the BLSTM model fails to accurately capture an airfoil's x and y coordinates, given the 17 shape and 6 structural variables.

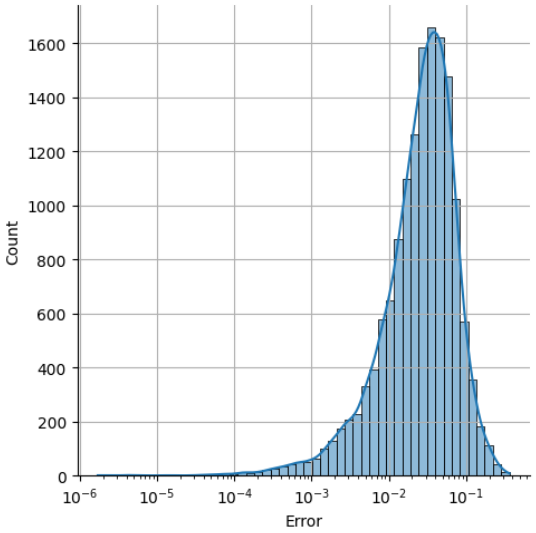


Figure 7.13: Distribution of the MAE as calculated on the testing dataset

To isolate the cases with high prediction inaccuracy, the 95th percentile is used as a

cut-off criterion. This means that all airfoils with an MAE above the 95th percentile are included in the outlier dataset to be used for analysis. This outlier dataset contains 549 airfoil samples. Upon closer inspection, it is clear that all airfoils in this dataset have impractical airfoil profiles. These impractical airfoil shapes, in general, have highly curved top and bottom surfaces usually with sharp nose radii. In Figure 7.14, a random sample of actual versus predicted airfoils present in the outlier dataset can be seen. From this figure, it is clear that these cases of high prediction error corresponded to impractical airfoil shapes.

The accuracy analysis of this section, therefore, highlights that the BLSTM model performs well on the task of predicting raw airfoil x and y coordinates given 23 parsimonious variables – *if* the airfoil has a practical shape. This practical concern is discussed in Sections 7.4.2 and 7.7.4, where it is shown that (1) conditioning on practical airfoil classes, and (2) constraining the latent vector as required by the specific cluster's means and covariance matrices, result in the generation of only practical airfoil profiles with desirable airfoil shape and structural characteristics.

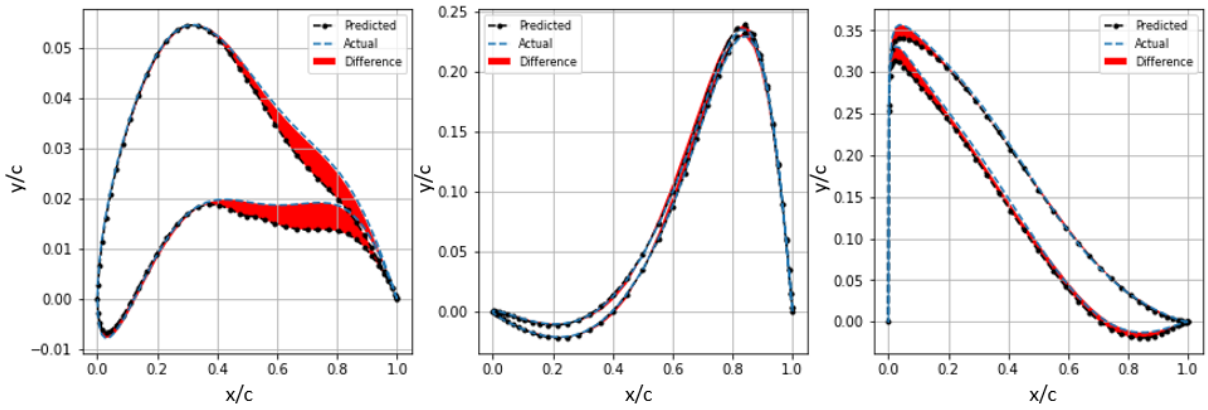


Figure 7.14: BLSTM fitting error on impractical airfoil shapes (MAE of 0.27 compared to an average MAE on the test set of 0.0038)

7.6.5 Module integration

The final module to be integrated into the TSM is the airfoil shape conversion module. As discussed in Section 7.6, the function of this module is to convert the BGMM-generated airfoil from the parsimonious variable domain to the raw x and y coordinate domain. This enables manual inspection of the quality of the airfoil produced and also verification of the DSMs predictions with the TSM cross-country results.

The shape conversion module is integrated with the TSM and has the inference process depicted in Figure 7.15. The inference process for airfoil generation with this module is as described in Algorithm 5. Here, the airfoil shape conversion module takes 23 parsimonious variables as input and predicts the corresponding airfoil profile in terms of raw x and y coordinates as output.

Algorithm 5 Airfoil Shape Conversion Module

Require: $\mathbf{z} \in \mathbb{R}^{23\times1}$ where $\mathbf{z}$ represents the airfoil parsimonious variable vector

Ensure: $\mathbf{c} \in \mathbb{R}^{160\times1}$ where $\mathbf{c}$ represents the high dimensional raw x and y coordinate vector

1: Scale $\mathbf{z}$ to generate vector $\mathbf{s} \in \mathbb{R}^{23\times1}$

2: Use $\mathbf{s}$ as input to the airfoil shape conversion module to generate $\mathbf{c} \in \mathbb{R}^{160\times1}$

3: **return c**

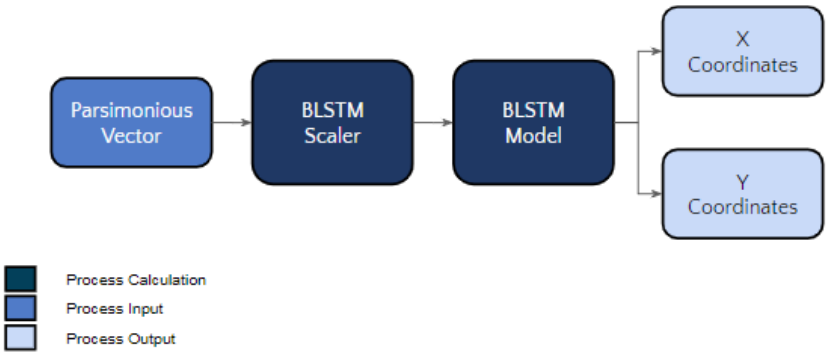


Figure 7.15: Depiction of the inference process of the airfoil shape conversion module

7.7 Development of the airfoil shape mapping module

In this section, the development of the airfoil shape mapping module is discussed. This is done in three successive steps. First, the process of linking the airfoil generator module and the airfoil shape conversion module to form the airfoil shape mapping module is discussed. Next, attention is given to the airfoil fitting process to understand how the airfoil shape mapping module is used to approximate any given airfoil profile. From here, the airfoil shape mapping module is used to fit 100 randomly sampled airfoils from the UIUC airfoil database. This is done to assess the fitting capacity of the proposed airfoil shape mapping module. Finally, the fitting error of the airfoils fitted in the previous step is analysed. This is done both from the perspective of profile fitting error and aerodynamic performance error calculation between the fitted and actual airfoils.

7.7.1 Overview

With both the airfoil generator as well as the airfoil shape conversion modules developed, the next step is to combine these two to form the airfoil shape mapping module. The purpose of the airfoil shape mapping module is to enable the efficient approximation of a supplied airfoil with a generated airfoil by either 23 parsimonious variable specifications or x and y coordinate definitions. That is, the airfoil shape mapping module is used to fit a supplied airfoil in order to have both its raw x and y coordinate definition as well as its parsimonious variable definition. This allows for the specification and translation of any given baseline design in the appropriate DSM and TSM formats, as seen in Sections 8.4 and D.3 respectively. Having such a combined system also allows for the final evaluation of the generalisation capacity and accuracy of the two underlying modules.

At a high level, the airfoil generation process, and hence the fitting process, as dictated by the airfoil shape mapping module, can be described as follows:

1. Sample a latent vector in the range -1 to 1 and a cluster choice in the range 0 to 49.
2. Given the airfoil cluster number, index for the appropriate covariance matrix of 49x1 and means matrix of 49x1 from the 49x49 means matrix and the 49x49 covariance matrix trained in the airfoil generator module.
3. Using the respective covariance and means matrices generate a new airfoil in terms of a 23-dimensional parsimonious vector.
4. Descale the 23 parsimonious vectors to arrive at the actual parsimonious vector with dimension 23x1.
5. The descaled parsimonious variables are then rescaled, this time with the BLSTM model scaler to form a new 23 by 1 parsimonious vector.
6. The BLSTM-scaled and transformed parsimonious vector is then used to generate the airfoil profile in terms of raw x and y coordinates with a dimensionality of 160x1.

7.7.2 Airfoil fitting

To evaluate the proposed airfoil shape mapping module's capacity to approximate a diverse set of out-of-sample airfoils, the model is used to fit the profiles of 100 randomly sampled airfoils taken from the UIUC airfoil database (M. Selig, 2022). The airfoil fitting is done by setting up an optimisation problem where the objective is to minimise the sum of the MAE difference between the generated airfoil x and y coordinates, and the target airfoil coordinates. In each iteration, the airfoil class and latent vector are updated by means of a GA, in the direction that would be most likely to minimise the objective function. It is found that the GA is able to reach acceptable fitting accuracy when a population size of 500 with 50 generations and a cross-over probability of 0.5 is used as optimisation model parameters.

In Figure 7.16 the average convergence for all 100 randomly sampled cases is shown. From this plot, it is observed that after 10 generations, feasible fitting tolerance is reached. For all 100 randomly sampled airfoils, the airfoil shape mapping module is able to approximate

each shape, with an average objective function value i.e. MSE of 2.08×10^{-8} and a standard deviation of 5.48×10^{-10}.

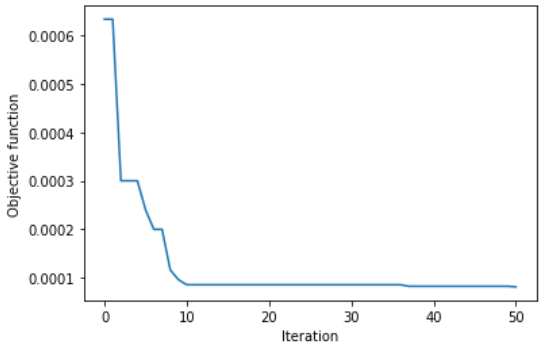
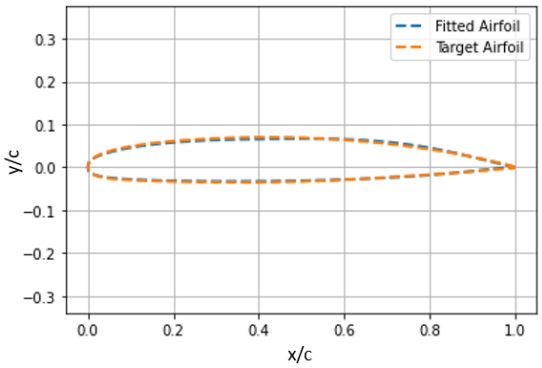


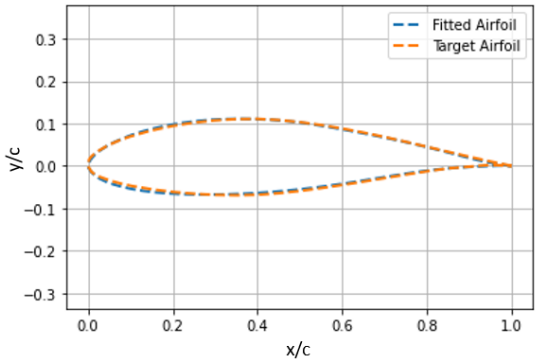
Figure 7.16: Average convergence rate of 100 fitted airfoils by means of the genetic algorithm, with the objective function measured as the mean absolute error between the fitted and actual airfoil x and y coordinates

7.7.3 Airfoil generalisation

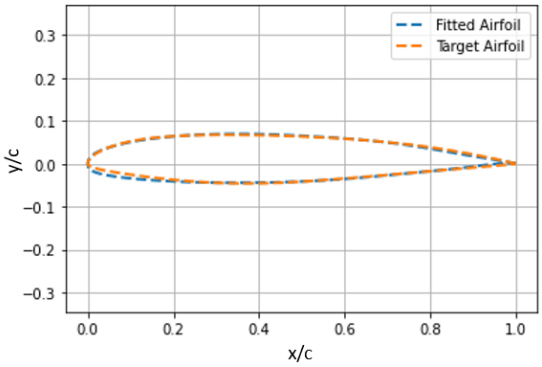
To understand the generalisation capability of the airfoil shape mapping module, the fitting errors are divided into three groups, i.e. low-fitting error, average-fitting error, and high-fitting error. With the model residuals normally distributed, a low-fitting error is defined as one standard deviation less than the average error. A high-fitting error is defined as one standard deviation higher than the average error. Figures 7.17a–7.18b depict the low-, average-, and high-fitting error airfoils used for this analysis.



(a) C141E Actual versus Generated Profiles (High accuracy)



(b) NACA 643418 Actual versus Generated Profiles (Avg. accuracy)



(c) Boeing-bacxxx Actual versus Generated Profiles (Low accuracy)

Figure 7.17: Varying degrees of BLSTM airfoil fitting error

These three fitted and actual airfoils are analysed in terms of their respective aerodynamic performance. This is done seeing that, even though small fitting errors are visible in Figures 7.17a–7.18b, it is known that small profile perturbations can cause large variances in aerodynamic performance. Therefore, if the shape mapping module has a high generalisation capacity, then the differences seen between the aerodynamic performance of the fitted and actual airfoils would be sufficiently small.

XFOIL (Drela, 1989) is used to determine the aerodynamic performance of each of the fitted and actual airfoils. The XFOIL simulation inputs were as follows:

1. N_{crit} set to 9,

2. Reynolds number of 2×10^6,

3. Mach number of 0, and

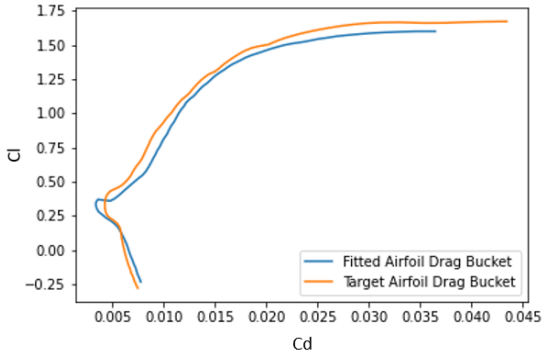
4. Angle of Attack ranging from 0 to 20 in 0.1 increments.

The results from this analysis can be seen in Figure 7.18. From this figure, it is observed that there is a more notable difference when comparing the aerodynamic performance of the fitted and actual airfoils compared to the differences seen in Figures 7.17a–7.18b. The differences in the analysed cases are, however, still within acceptable ranges with an average drag coefficient MAE of 0.0012, lift coefficient MAE of 0.0316, and an average adjusted R^2 of 0.97 for the case of the drag buckets. [11]

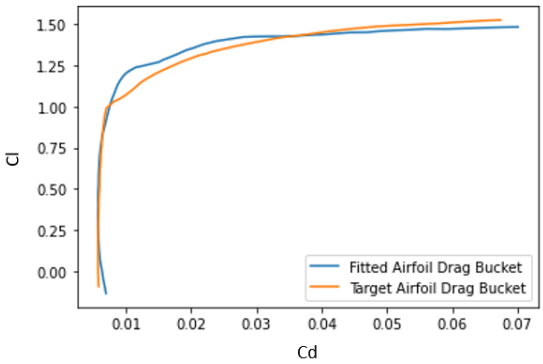
In Figures F.1 and F.2 a further sample of nine of the 100 fitted airfoils and the error between the actual 23 parsimonious variables and the airfoil shape generation model predicted values are shown respectively. From these figures, a low deviation between the fitted and the actual airfoils is apparent with an average fitting error rate of 0.69% observed across all 23 variables.

The minor differences between the fitted and actual airfoils alongside the minor difference between their respective aerodynamic performance are indicative of the accuracy of the airfoil shape mapping module, and hence the respective underlying sub-modules.

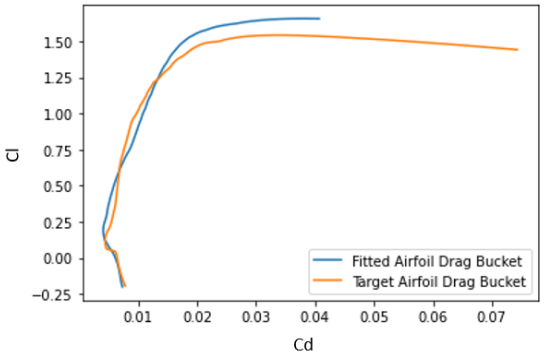
[11] Adjusted R-squared is a modified version of the R-squared statistic that takes into account the number of predictors or independent variables in a regression model. It is a measure of the proportion of the variation in the dependent variable that is explained by the independent variables, adjusted for the number of variables in the model. Adjusted R-squared penalises the addition of variables that do not improve the model's predictive power, unlike R-squared, which can increase even when adding insignificant variables. In general, a higher adjusted R-squared value indicates a better fit of the model to the data.



(a) C141E Actual versus Generated Profiles (High-accuracy)



(b) NACA 643418 Actual versus Generated Profiles (Avg. accuracy)



(c) Boeing-bacxxx Actual versus Generated Profiles (Low accuracy)

Figure 7.18: Aerodynamic performance differences for varying degrees of fitting errors

7.7.4 Padulo completeness

This section discusses the Padulo completeness of the airfoil shape mapping module.

Motivating the parsimony characteristics of the airfoil shape mapping module, readers can refer to Figure 7.19, where the profiles of different airfoils[12] are shown that are generated by perturbing the input space of the first four latent variables (out of the possible 29 variables) of the airfoil generator module in increments of 0.1, 0.25, 0.75, and 1 respectively. From this figure, it is observed that, when incrementing the input space of a given latent variable, predictable airfoil shape permutation results and that only one change to a single variable

[12]To generate the baseline airfoil the airfoil generator was conditioned on airfoil cluster one and all latent variables were set to 0.

is required to exert global shape control. This, therefore, highlights the parsimony of the airfoil shape mapping module where a limited number of parameters are required to introduce significant changes to the main geometrical features of the airfoil.

Figure 7.19 can also be used to address the flawlessness characteristic of the airfoil shape mapping module. In this figure, it is observed that, when exploiting the $X4$ latent variable to its upper bound, impractical airfoils start to form. This indicates that the module is not fully flawless and care should be taken on both the conditioning and sampling of the airfoil shape mapping module's input latent space to guarantee flawlessness whilst ensuring diversity of airfoil generation.

For comments on the completeness of the airfoil shape mapping module, reference is made to the analysis done in Section 4.5.2. Specifically, it is shown here that the parsimonious variable extraction framework upon which the airfoil shape mapping module is based, is designed with 23 variables to have near 80% coverage of the entire airfoil design space. In Section 7.1, further evidence of the coverage of the airfoil shape mapping module is given. Here it is shown that the module is able to approximate 100 randomly sampled airfoils from the UIUC airfoil database with a high degree of accuracy when measuring fitting accuracy as both profile deviation as well as aerodynamic performance deviation from the target airfoils.

The developed airfoil shape mapping module can also be shown to adhere to the intuitiveness criteria once again due to the design of the underlying parsimonious variable extraction framework. In Section 4.3 it is shown how the development of the entire parsimonious variable extraction framework is done under the requirement of parsimony. Every shape and structural variable chosen for inclusion in the framework is done so that all variables have physical meaning and hence map back to explicit global control over the airfoil characteristics in an intuitive way.

For comments on the final Padulo completeness criteria, i.e. orthogonality, reference is made to the underlying model of the airfoil generator module, the BGMM. With BGMMs, uniqueness is not always guaranteed. This is because a BGMM can model complex data

distributions with multiple modes, meaning that there can be multiple data points that have similar probability density values. When sampling from a BGMM, it is possible for different input values to have the same probability density values in the mixture model, leading to the possibility of producing the same output. However, the probability of two different inputs producing the same output is generally very low in practice, especially if the number of mixture components in the BGMM is sufficiently large and the mixture weights are well distributed. In other words, while uniqueness cannot be guaranteed in theory, the BGMM sampling algorithm can still be highly effective at generating diverse and representative samples from the modelled data distribution. With the large number of components that are well distributed in the case of our BGMM, although uniqueness is not guaranteed, it is highly unlikely to generate two different airfoils for the same input. Also, seeing that the mixture weights are well distributed, even if the same inputs result in a different airfoil output, the characteristics of these airfoils would be very similar and hence not have a detrimental impact on the optimisation and inference processes.

It can therefore be concluded that, although the airfoil shape mapping module does not fully satisfy all of Padulo's criteria, the most important requirements for this research, i.e. completeness, parsimony, and intuitiveness are adequately addressed.

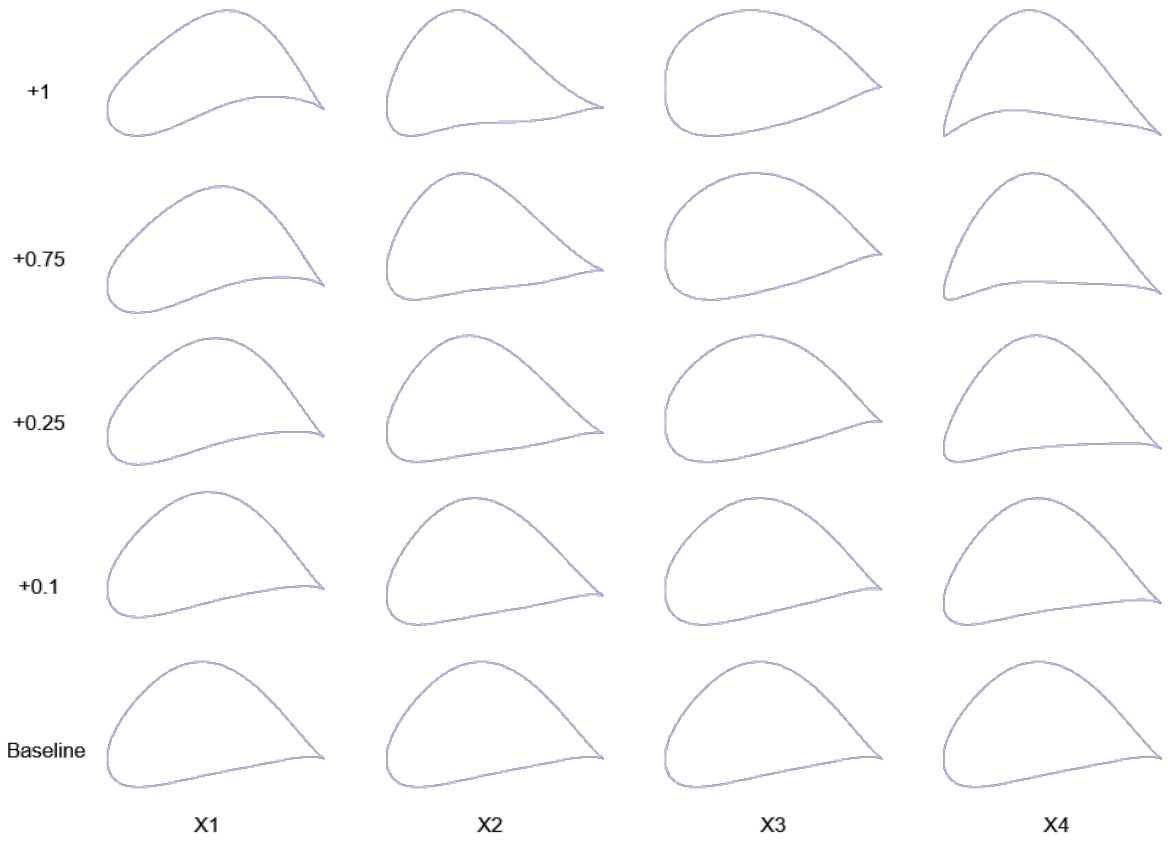


Figure 7.19: Depiction of different airfoils generated by exploiting the first four input dimensions of the airfoil shape mapping module

7.8 Conclusion

This chapter discusses the development of the airfoil generator module and the airfoil shape conversion module, which ultimately form the airfoil shape mapping module.

For the airfoil generator module, it is shown that a BGMM is effective both in generating an airfoil in terms of 23 parsimonious variables, and clustering airfoils in 50 segments based on their parametric coefficient and parsimonious variable descriptions. This module is also shown to be highly accurate in generating airfoils that have the same distribution, and global and local structure in their parsimonious variables as those found in the training dataset.

Also in this chapter, the airfoil shape conversion module development is discussed. This module is implemented using a BLSTM and is shown to be accurate as an airfoil constructive technique when tasked with generating airfoil profiles given the parsimonious variable definition of a practical airfoil profile.

Finally, these two modules are combined to form the airfoil shape mapping module. This module is motivated by three requirements: (1) having a single system that enables efficient mapping between different airfoil profile domains to enable both TSM and DSM calculations with the same airfoil specifications; (2) having a method of verifying the accuracy of the underlying sub-components; and (3) having the capability of acquiring the 23 parsimonious variables of any given airfoil in terms of its x and y coordinate definition.

In order to verify the accuracy of the airfoil shape mapping and hence the airfoil generator and airfoil shape conversion modules, 100 randomly sampled airfoils from the UIUC airfoil database are fitted. These airfoils did not form part of either the airfoil generator or airfoil shape conversion modules and hence are regarded as out-of-distribution data samples. The airfoil shape mapping module is able to fit this diverse group of 100 airfoils with a high degree of accuracy with an average MAE of 2.08×10^{-8} and a standard deviation of 5.48×10^{-10}. When analysing the aerodynamic performance of the fitted versus actual airfoils further favourable results are seen with an R^2 of 0.97.

From these analysis it can therefore be concluded that the developed airfoil shape mapping module has adequate accuracy and generalisation capacity for use in the development and validation of the optimisation module. As mentioned, for global cross-country performance optimisation of the JS4 sailplane the airfoil shape mapping module, the underlying sub-modules, and the shape dataset will be refined – see Section 10.2.2.

Chapter 8

Developing the deep learning sailplane model

8.1 Introduction

As detailed in Section 1.7, the DSM is the main component developed in this research that is an integration of all the modules developed up to this point, into the TSM. The development and evaluation of this DSM are therefore discussed in this chapter. First, in Section 8.2 an overview of the DSM is given with a specific focus on the inference process and the process of intuitive constraint setting using this module's input requirements. Then in Section 8.3 the methodology of how the accuracy of the DSM is verified is discussed. Specific attention to the validation case definitions is given in Section 8.4. Finally, in Section 8.5 the validation of the DSM is done and the accuracy as well as the validation case limitations are addressed.

8.2 DSM overview

In this section, the fully integrated DSM is discussed from the viewpoint of the individual component functioning and constraints setting. This gives insight into (1) the DSM functioning and input requirements, (2) how the individual modules contribute to the sailplane cross-country performance estimation, and (3) how the integrated modules assist in intuitive constraint setting on the airfoil profiles, 2D airfoil performance characteristics, and wing planform.

8.2.1 Inference process

In Figure 8.1 the integrated DSM is shown. This figure highlights the process of predicting the average cross-country performance of a given sailplane for a specified flight task and weather condition. As seen in this figure, there are a total of 15 steps involved when calculating the average cross-country performance of a sailplane with the DSM.

Steps (1)–(2) are concerned with generating an airfoil shape with the airfoil generator module, for each station along the wing span. Achieving this is accomplished using Algorithm 4. As seen in this algorithm, a specification of each airfoil in terms of its latent vector and airfoil cluster is required as input and as output, the 23 parsimonious variables for each airfoil at each station along the wing span results. These results are then passed to the PDLM in step (3).

Steps (4)–(5) involve the generation of the 2D airfoil sectional properties for each airfoil along the wing span, as required by the dynamic station LLT. This is accomplished by employing the PDLM to calculate the eight response variables for every airfoil along the wing span and then indexing the respective outputs to acquire the lift-curve slope and zero-lift Angle of Attack. This calculation is done by means of Algorithm 2.

In step (6) the results of step (5) are passed to the dynamic station LLT and a flight envelope is generated over which the calculation of the wing 3D lift coefficient, wing induced drag, planform efficiency, and lift distribution over the wing span are done.

These results are then passed to the IPDLM in step (7). In steps (8)–(9) the total profile drag for each station along the wing span is calculated, for every operating condition in the flight envelope. This is done by executing Algorithm 3 with the parsimonious variables of step (3) and the lift distribution of each operating condition in the flight envelope as input. These results are then passed to the sailplane module in step (10).

In step (11) the total sailplane performance is calculated in terms of the total drag and total lift performance for every operating condition present in the generated flight envelope. This is done by first calculating the winglet drag contribution according to Equation 3.9. Next in this step, is the calculation of the fuselage/tail drag. This is done by using the performance curve supplied by Jonker Sailplanes to interpolate the corresponding fuselage/tail drag for every airspeed present in the flight envelope. For the induced drag calculation, the appropriate results of step (7) are indexed. With the profile drag, the winglet drag, the induced drag, and the fuselage/tail drag coefficients known for the entire flight envelope, next, these coefficients are transformed into their dimensionalised forms. This is done using the wing geometry specification and allows for direct summation of the coefficients. To determine the total drag coefficient for the wing, the total dimensionalised drag is then converted back to its original non-dimensionalised form and Equation 3.9 is used. The final step is to also convert the lift coefficient for every condition present in the flight envelope to its dimensionalised form using the appropriate wing geometry and flight condition information.

The total sailplane performance for every flight condition over the flight envelope is then passed to the sailplane cross-country module in step (12). In steps (13)–(15), this module uses the sailplane performance and weather model as input, and employing the process discussed in Section 2.2, the average cross-country performance is given as output.

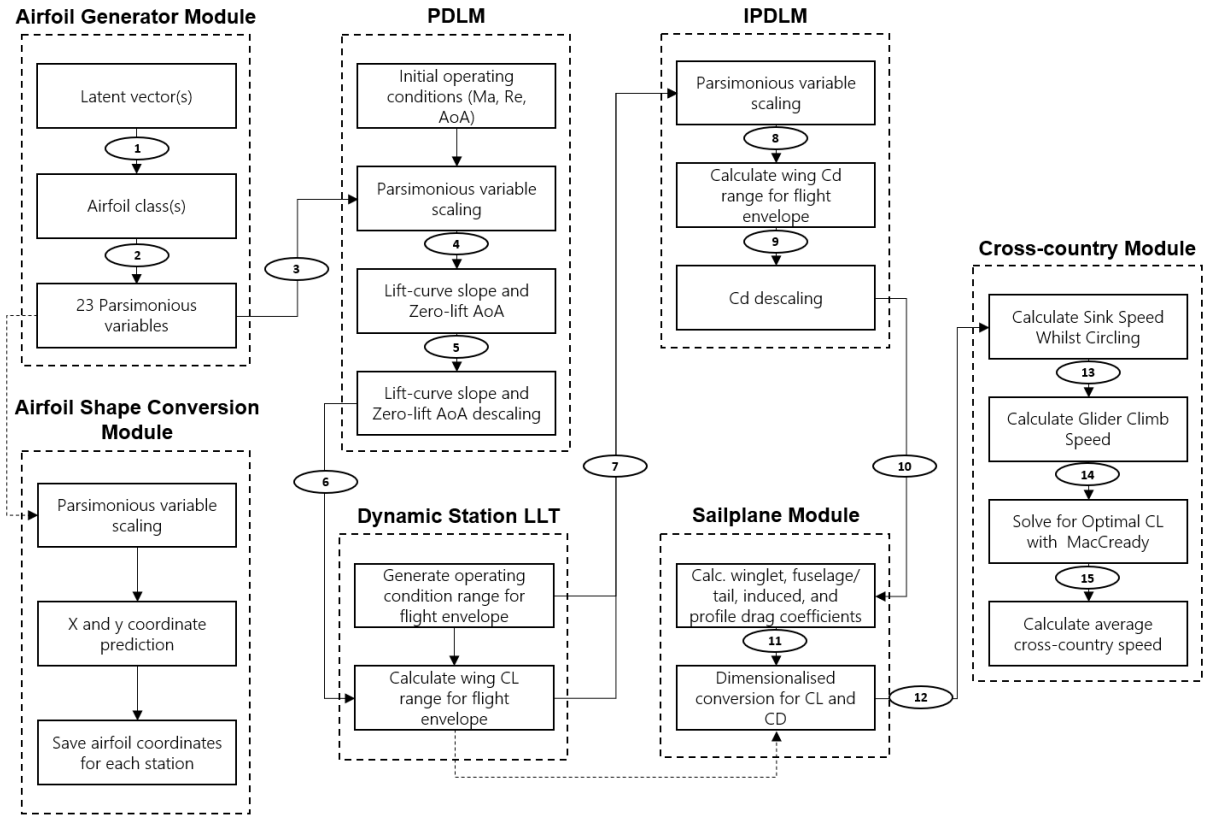


Figure 8.1: Depiction of the final inference process of DSM, highlighting the integration of the developed sub-modules

To highlight the efficiency improvements of the DSM when compared to the TSM, 200[1] airfoil/wing configurations are analysed by both models and the run times are logged for comparison. The distribution of the run times for both models can be seen in Figure 8.2. From this figure it is observed that the DSM has a significantly lower average run time across the 200 cases than the TSM, with the DSM average run time being 42.51 seconds and the TSM average run time being 262.84 seconds. This indicates that on average, the DSMs inference process is 6.18 times faster than the TSMs inference process over the 200 cases investigated. Also from this figure it is observed that the TSM has a higher spread

[1]The airfoils of these 200 configurations are generated by randomly perturbing the airfoil shape mapping modules input space to generate the 23 parsimonious variables as well as the raw x and y coordinates as required by the DSM, and TSM respectively. To ensure only realistic airfoils resulted, the airfoil generator was constrained to the following airfoil clusters: 21-22, 26-27, and 29. For the wing geometry, random wingspan increments and chords are uniformly sampled whilst ensuring realistic wing planforms. That is planforms with sequentially decreasing chord lengths. The total span was constrained to be between 15 and 20 meters.

indicating more variance between run times. This is caused by the convergence issues of XFOIL and is eliminated by the use of the PDLM, and IPDLM in the DSM, and therefore the lower variance between run times observed for this module.

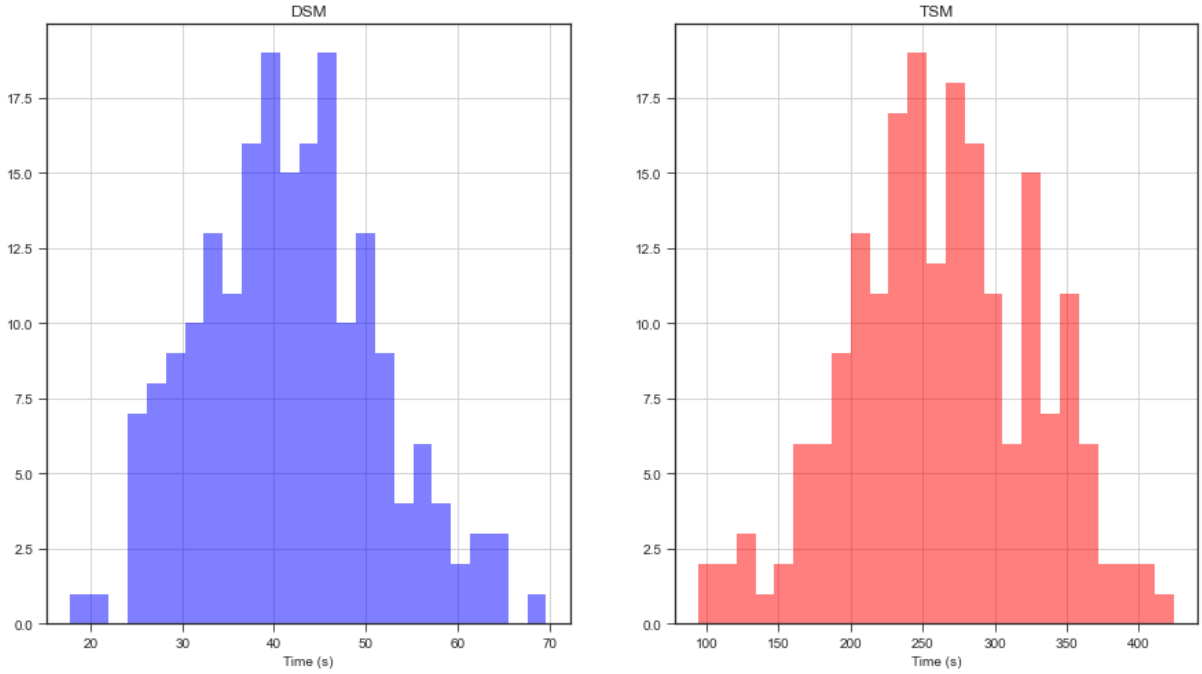


Figure 8.2: Comparison of the run times of the DSM and TSM models for 200 wing/airfoil configurations

8.2.2 Intuitive constraints setting

In the current DSM, the optimisation process can be constrained in three ways; at the airfoil generator module, the PDLM module, or the dynamic station LLT module.

The first to be discussed is the airfoil generator module. The DSM is developed in such a way that either every station on the wing can have its own individual airfoil definition, or the same airfoil definition can be used at every station. As noted in Section 1.7, in this research, only five sections per span are investigated and therefore, all expressions and definitions presented in this section are done under this global constraint. Specifying the airfoil present at each station with the airfoil generator module involves utilising Algorithm 4. Hence, for each station a vector $\mathbf{X} \in [-1, 1]^{5\times 29}$ will be generated, sampled

from a Normal distribution, and a vector $\mathbf{Y} \in [0, 49]^{5\times1}$ will be generated. After passing through the airfoil generator, a vector $\mathbf{P} \in \mathbb{R}^{5\times23}$ results, where the values of the 23 parsimonious variables lie between the values of the specific airfoil profile of the cluster chosen. At this point, constraints can be imposed on the vector P as follows:

$$\forall i, j : P_{i,j} \in [a_{i,j}, b_{i,j}] \tag{8.1}$$

where i and j are indices that range from 1 to 5 and 1 to 23 respectively, $P_{i,j}$ is the (i,j)-th element of the vector $\mathbf{P}$, and $[a_{i,j}, b_{i,j}]$ is the constrained interval of allowable values for that element. If the design constraints of the airfoil profile at each station are not met, penalties can be added to the objective function to guide the optimisation process to favourable designs. This penalty specification can be seen in Section 9.2.

The next module at which constraints can be set as input to the optimisation pipeline is the PDLM module. In this module, rather than constraining the geometry of the airfoil or the planform, the 2D performance characteristics of each airfoil can be constrained. In Table 5.1 the eight response variables can be seen. It is here where the motivation for the additional non-conventional 2D performance characteristics are highlighted, other than for their use in the dynamic station LLT model. With the PDLM, the eight 2D airfoil performance characteristics of each airfoil along the wing span can be calculated, hence resulting in $\mathbf{L} \in \mathbb{R}^{5\times8}$. Constraints can thus be imposed as input to the optimisation pipeline on $\mathbf{L}$ to ensure desirable 2D airfoil performance characteristics result at each wing station. These constraints can be expressed as:

$$\forall a \in A : \quad l_i \leq L_i(a) \leq u_i \tag{8.2}$$

where a represents an airfoil from the set A containing the 5 airfoils along the wing span, L_i the i-th response variable of each airfoil $a \in A$ with $i \in [1, 8]$, l_i the lower bound of the i-th response variable, and u_i the lower bound of the i-th response variable. With this, each individual airfoil can be uniquely constrained on any one or all of the eight response variables. For example, all airfoils can be constrained to have only laminar flow across the entire upper and lower surface, the tip airfoil can be constrained to have a specific maximum C_l, and the root airfoil can be constrained to have a specific zero-lift angle of

attack. If these upper and lower performance conditions are not met, once again a penalty function can be added or the iteration can be terminated and a new one started.

The final module at which constraints can be set as input to the optimisation pipeline is the dynamic station LLT module. Here, constraints can be set on the wing geometry, and if the constraints do not adhere, the simulation can again either be terminated or a specific penalty function added to the optimisation objective function. With five stations, each having a length, a width, and a respective constraint on each, the half-wing planform definition can be expressed as:

$$\forall s \in S : \begin{cases} l_l \leq L(s) \leq u_l \\ l_c \leq C(s) \leq u_c \end{cases} \tag{8.3}$$

where s a station from the set S containing the 5 stations along the wing span, $L(s)$ the length of station s constrained to be between the upper and lower bounds l_l and u_l respectively, and $C(s)$ the chord of station s constrained to be between the upper and lower bounds l_c and u_c respectively.

Later, in Section 9.4, it is shown how the DSM together with the intuitive constraint formulation derived in this section is used in an end-to-end sailplane cross-country performance optimisation routine.

8.3 Methodology of accuracy validation

In this section, the validation of the accuracy of the DSM is done by comparing this module's results with those of the TSM for a diverse range of wing/airfoil combinations.

Validating the DSM's performance across a broad spectrum of flight cases involves defining two objectives. In each objective, a total of five simulation cases are analysed, each with its own unique airfoil and wing definition. These 10 simulations are run with the DSM and the TSM and their respective average cross-country speed predictions are logged for comparison.

Although the wing and airfoil designs vary across each simulation, it is important that the cross-country task and weather conditions are kept constant. This allows for a direct comparison of the results for the same simulation definitions.

For the cross-country simulation, the Quasts weather model is used with: A1 (0.00%), A2 (56.67%), B1 (3.33%), and B2 (40.00%), and a 300 km flight task is specified. For the lifting line inputs, the wing is divided into 200 subdivisions, and a zero-degree twist angle is assumed.

Finally, for the sailplane module, the fuselage performance is assumed to vary as depicted in Figure D.2 – as supplied by Jonker Sailplanes. This figure shows the CFD simulated 3D drag performance of the wing/tail configuration of a JS3-based sailplane for different flight speeds. A sailplane weight of 539 kg is assumed, and the thermal radii are defined to vary between 30 metres and 400 metres in increments of 10 metres.

8.4 Validation case definition

In this section, the two validation objectives alongside the respective wing/airfoil configurations used for each of the five cases in each objective are discussed.

As mentioned earlier, for both objectives, a total of five simulation cases are defined. In objective one, each of the five cases has the same airfoil configuration for each station along the span and the wing planform is allowed to vary. In the second objective, each simulation case has different airfoils for each station along the span, and the wing planform is again allowed to vary.

For the two objectives, the airfoil specification, airfoil sequence, station chords, and wingspan increments are as depicted in Tables 8.1 – 8.2 respectively. This table shows how the same five base wing planforms are used in the two objectives, with each planform case having its own unique airfoil setting. That is, each case has a different airfoil shape and airfoil sequence. The aspect ratios and areas of the five planform cases can also be seen here, highlighting the diversity in validation designs used. In Figure 8.3, the five

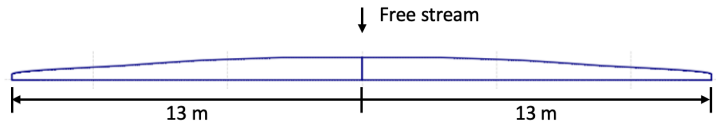
planforms resulting from the geometry defined in Tables 8.1 - 8.2 are shown.

Table 8.1: Objective 1: Airfoil sequence, wingspan increments, and chords defined per station

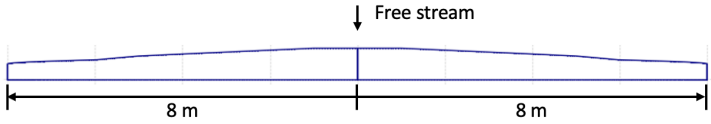
Case	Half-span location (m)						Chord length (m)						A.R	Area	Airfoil Configuration
	1	2	3	4	5	6	1	2	3	4	5	6			
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	42.5	15.9	1,1,1,1,1,1
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	25.3	10.1	2,2,2,2,2,2
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	17.5	11.2	3,3,3,3,3,3
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	69.7	9.7	4,4,4,4,4,4
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	32	12.5	5,5,5,5,5,5

Table 8.2: Objective 2: Airfoil sequence, wingspan increments, and chords defined per station

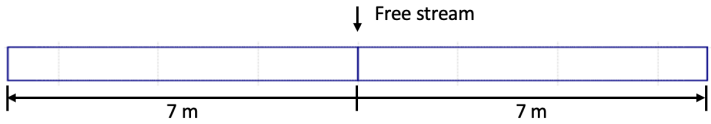
Case	Half-span location (m)						Chord length (m)						A.R	Area	Airfoil Configuration
	1	2	3	4	5	6	1	2	3	4	5	6			
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	42.5	15.9	1,2,3,4,5
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	25.3	10.1	2,3,4,5,6
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	17.5	11.2	5,5,5,5,6
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	69.7	9.7	3,3,4,5,6
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	32	12.5	1,2,3,4,5



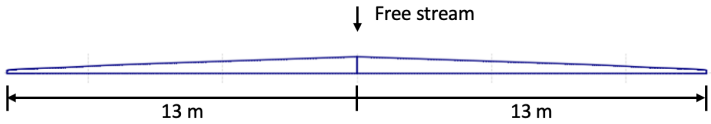
(a) Planform for case 1



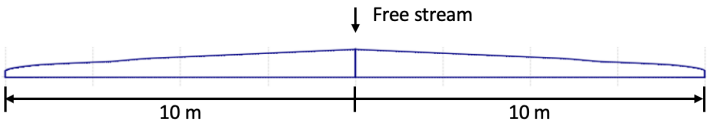
(b) Planform for case 2



(c) Planform for case 3



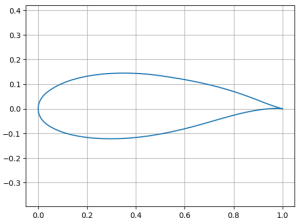
(d) Planform for case 4



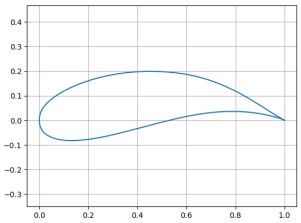
(e) Planform for case 5

Figure 8.3: Five respective planforms used in the validation of the DSMs average cross-country performance prediction accuracy

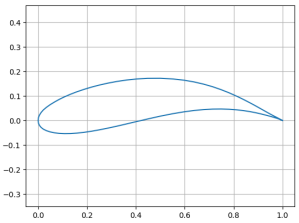
The airfoil profiles corresponding to the depicted station configurations in Tables 8.1 – 8.2 can be seen in Figure 8.4. These six airfoils are generated by randomly perturbing the airfoil shape mapping module's input latent space. Using the airfoil shape mapping module means that both the parsimonious variable representations and the raw x and y coordinate definitions are available at the time of generation. From Figure 8.4 it can be seen that the airfoils randomly generated have diverse profiles with varying degrees of camber, curvature, and thickness distribution.



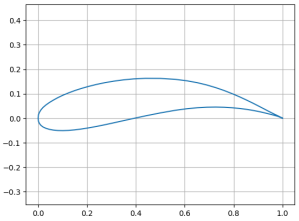
(a) Airfoil 1



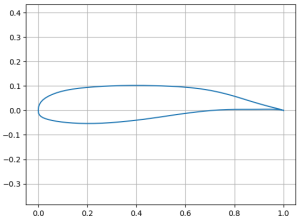
(b) Airfoil 2



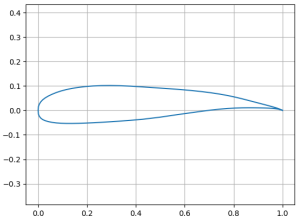
(c) Airfoil 3



(d) Airfoil 4



(e) Airfoil 5



(f) Airfoil 6

Figure 8.4: Randomly generated airfoils used to analyse the polar plots generated by the 2D Performance Deep Learning Module

Each of the 10 validation cases defined, across the two objectives, together with the specified flight tasks, are used in both the DSM and TSM to predict the respective average cross-country speeds. The results from these simulations and the analyses and interpretation thereof can be seen in the following section.

8.5 DSM performance validation

In this section, the results for objectives one and two are given. These results are discussed to understand the accuracy of the DSM for a diverse range of wing/airfoil configurations. Differences between the TSM and the DSM are also explained. Finally, 25 additional experimental results are discussed to ensure that the conclusions drawn from the 10 simulation runs in objectives one and two are significant.

8.5.1 Base case validation

The results from the respective simulations for the two objectives can be seen in Tables 8.3 and 8.4 respectively. Here, the MAPE is measured between the results of the TSM and the DSM simulations, using the TSM results as a baseline. For objective one, the average of the MAPE for all five cases is 4.89%, and the R^2 is 0.9778. The largest error for objective one is seen in case one and equates to 8.81%, i.e. the thickest airfoil with the least amount of camber. The lowest error for objective one sees a MAPE of 3.50% and is seen in case five, i.e. the thinnest airfoil with slight camber near the TE. This can be attributed to the fact that the majority of the airfoil profiles in the training dataset are more similar to the profiles depicted in Figures 8.4c, 8.4e, and 8.4f. It is interesting to note that, for all the cases of objective one, the DSM predicts higher average cross-country speeds. This is mainly due to the phenomenon discussed in Section 6.7.1 where the PDLM overpredicts the lift coefficient. This overprediction is, however, very slight and the aggregate effect over all the cross-country simulations on average is less than 5%.

For objective two a similar average MAPE of 4.93% is observed across all five cases with an average R^2 of 0.9081. The slightly lower accuracy and R^2 here can be attributed to the fact that more complex wing/airfoil combinations are used. Although there is a lower R^2 for this objective, it is still close to 1, indicating a strong positive correlation. Again here the highest degree of inaccuracy is seen for case one and is 7.72%. However, in this objective, the highest level of accuracy is seen for case three and has a MAPE of 0.78%. Case three of objective two mostly has the same airfoil configuration as seen in case five in objective one. The only difference here is a different airfoil used at the wing tip and a non-tapered wing planform. Seeing that both cases have the highest level of accuracy but different wing planform definitions, the previous conclusion for this case is strengthened. That is, airfoils 3, 5, and 6 share the same profiles as most of the airfoil profiles in the training dataset and hence more accurate predictions for these cases.

With the exclusion of case one of objective two, again here the DSM results are higher than those predicted with the TSM. Besides the reason already mentioned, another cause of this slight overprediction is attributed to the differences between the airfoil profile

definitions used in the respective models. The DSM makes use of the low dimensional 23 parsimonious variables, whereas the TSM makes use of the high dimensional raw x and y coordinates and the airfoil shape mapping module is used to convert from one dimension to the other. The airfoil shape mapping module is shown to be accurate to an acceptable degree when mapping the 23 parsimonious variables to raw x and y coordinates. However, as shown in Section 7.7.3, slight differences in profile definitions between the two domains can cause deviations in predicted aerodynamic performance. However, the differences here between the TSM and DSM are sufficiently small with a less than 5% average error rate for the complex wing/airfoil validation cases.

Table 8.3: Objective 1: Cross-country performance comparison of the traditional and deep learning integrated model

Case	Traditional Model	Integrated DL Model	MAPE
	Avg. Cross-Country Speed	Avg. Cross-Country Speed	
1	98.19	106.84	8.81%
2	87.21	91.16	4.54%
3	81.75	84.79	3.72%
4	99.38	103.25	3.89%
5	104.36	108.01	3.50%

Table 8.4: Objective 2: Cross-country performance comparison of the traditional and deep learning integrated model

Case	Traditional Model	Integrated DL Model	MAPE
	Avg. Cross-Country Speed	Avg. Cross-Country Speed	
1	81.19	74.92	7.72%
2	87.27	91.40	4.73%
3	93.29	94.02	0.78%
4	94.80	98.33	3.72%
5	69.37	74.72	7.71%

8.5.2 Additional case validation

To ensure that the findings from objectives one and two are significant, 25 additional experiments are run where the wing planforms of cases one to five are each simulated

with airfoils one to five. In each case, the MAPE between the TSM and DSM simulations are compared. Table 8.5 shows the results of this analysis.

It is noted that for all 25 additional cases investigated, an average MAPE of 4.69% and a R^2 of 0.9489 is observed. Also, across all cases, the highest degree of accuracy is seen for the box wing planform, i.e. the wing planform of Figure 8.3c. The most accurate predictions are for the combination of the box wing coupled with either airfoil 3, 4, or 5 and equated to 3.57%, 0.21%, and 0.51% respectively. These findings are directly aligned with the findings of objectives one and two. This, therefore, confirms that the favourable model performance observed is not a result of chance from the small number of experiments that were run.

Table 8.5: Additional experiments to validate the accuracy of the DSM's average cross-country speed prediction

Case	Half-span location (m)						Chord length (m)						Airfoil	Traditional	Deep learning	MAPE
	1	2	3	4	5	6	1	2	3	4	5	6	Sequence	model	model	
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	1,1,1,1,1	98.19	106.84	8.81
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	1,1,1,1,1	88.09	95.20	8.08
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	1,1,1,1,1	83.52	90.16	7.95
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	1,1,1,1,1	91.99	102.33	11.23
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	1,1,1,1,1	94.92	103.45	8.99
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	2,2,2,2,2	93.49	96.88	3.63
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	2,2,2,2,2	87.22	91.16	4.52
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	2,2,2,2,2	78.72	82.00	4.17
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	2,2,2,2,2	97.63	101.41	3.87
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	2,2,2,2,2	92.70	96.50	4.09
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	3,3,3,3,3	95.73	99.91	4.36
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	3,3,3,3,3	90.61	94.88	4.70
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	3,3,3,3,3	81.88	84.80	3.57
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	3,3,3,3,3	101.56	107.18	5.52
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	3,3,3,3,3	95.72	101.74	6.29
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	4,4,4,4,4	96.71	99.16	2.53
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	4,4,4,4,4	88.87	93.33	5.02
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	4,4,4,4,4	83.30	83.47	0.21
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	4,4,4,4,4	100.47	103.25	2.77
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	4,4,4,4,4	96.66	98.69	2.10
1	0	3	6	9	11	13	0.8	0.8	0.7	0.5	0.4	0.2	5,5,5,5,5	104.20	109.62	5.21
2	0	1	3	5	6	8	0.8	0.8	0.7	0.6	0.5	0.4	5,5,5,5,5	99.74	101.51	1.78
3	0	1	3	5	6	7	0.8	0.8	0.8	0.8	0.8	0.8	5,5,5,5,5	93.28	93.14	0.15
4	0	3	6	9	11	13	0.6	0.5	0.4	0.3	0.2	0.1	5,5,5,5,5	106.92	111.36	4.15
5	0	4	6	7	9	10	0.9	0.7	0.6	0.5	0.4	0.2	5,5,5,5,5	104.21	108.00	3.64

8.5.3 Validation discussion

It is important to note that all validation is done under certain constraints, specifically: (1) the fuselage performance at multiple flight speeds is fixed; (2) a 300km cross-country task is specified; (3) no wing twist is specified; (4) the thermal radii are fixed; and (5) the strength and allocation to the A1, A2, B1, B2 thermal models are fixed.

Changing these simulation settings will however not affect any of the deep learning module calculations but only influence a subsection of the calculations of the dynamic station LLT and the cross-country models. It can, therefore, be shown that changing any or all of these constants and re-running the defined experiments will yield similar correlation and MAPE results when comparing the TSM and DSM's results.

It is also important to note that the reported accuracy of 4.84% and an R^2 of 0.95 is adequate only for the DSM's use in the development and validation of the optimisation module, as seen in Chapter 9. This is because this chapter is focused on developing the optimisation module, validating its performance and decisions made during the optimisation process, and understanding possible discrepancies in the model architecture and predictions. All validation is performed merely as an exercise to understand the module's functioning and is done against simple baseline models. The objective here is therefore the development of a functioning optimisation module that makes use of the DSM and allows for explicit constraint definition as input to the optimisation pipeline, and not optimisation accuracy. The insights from this chapter is then applied in Chapter 10 where all sub-modules, and hence also the DSM, is refined such to allow for full optimisation of a SOTA standard class sailplane.

8.6 Conclusion

In this chapter, the integrated DSM is discussed. Each sub-element and its individual role in the final architecture is described. In addition, the way in which the DSM can be used to set intuitive constraints on the airfoil profiles, wing planform, and the sectional

2D airfoil performance characteristics is discussed. Finally, the accuracy of the DSM is validated with the verified TSM, as detailed in Section 8.5.

The DSM is shown to have adequate accuracy under diverse wing and airfoil combinations with high positive correlation and low MAPE across all cases investigated. Specifically, across all simulations the DSM is shown to have an average error rate of 4.84% and an average R^2 of 0.95.

In Sections 9.2 to 9.3 this DSM is embedded into an optimisation module which in turn is used in various case studies to verify the efficacy and usability of the designed optimisation module. Then, in Section 10.3, a refined version of the DSM and optimisation module is used to achieve explicitly constrained global optimisation of the JS4 sailplane.

Chapter 9

Optimisation module development and verification

9.1 Introduction

This chapter is centred around the development and verification of the optimisation module such that, after respective sub-module refinement, it is fit for use in SOTA glider optimisation – as discussed later in Section 10.3. The optimisation module should therefore be developed to facilitate multi-objective sailplane cross-country performance optimisation whilst allowing for intuitive constraints setting as input to the optimisation process.

In Section 9.2, an overview of the functioning of the optimisation module is given with specific focus on the optimisation process and the user-defined objective function penalisation process. Next, in Section 9.3, the optimisation module is verified under two optimisation cases. Here, the two cases have different airfoil settings along the span of the wing and the optimisation algorithm must find the optimal planform within the defined constraints. An analysis is performed for both optimisation cases to ensure that the optimisation module makes the correct planform design decisions while adhering to the user-defined constraints. In Section 9.4 the optimisation module is used to improve the average cross-country performance of a basic baseline sailplane design, with explicit

constraints imposed on the airfoil parsimonious shape variables, the airfoil parsimonious structural variables, and the wing geometry. This is done to ascertain if the optimisation module functions correctly in explicitly multi-constrained optimisation settings. Finally, in Section 9.5, the optimised design is analysed in order to understand: (1) how the design compares to the baseline design, (2) why the optimisation module made certain design decisions, and (3) whether all constraints are adhered to in the optimisation process. Conclusions are then drawn on the robustness and efficacy of the optimisation module, the discrepancies identified, and the potential areas of improvement of the module before downstream use in JS4 sailplane optimisation.

9.2 Optimisation module overview

The optimisation module developed in this section has a single objective: to utilise the DSM to enable efficient sailplane cross-country performance optimisation with intuitive user input for wing and airfoil design constraints. This optimisation module is depicted in Figure 9.1. Here it can be seen that the core part of this optimisation module is the DSM, as discussed in Section 8.2.1. Also from this figure, the ability to intuitively set constraints as input to the DSM is clearly depicted. This process was thoroughly discussed in Section 8.2.2.

In this optimisation module, the chosen optimisation algorithm is the GA (Bozorg-Haddad et al., 2017) as discussed in Section 2.7.1. However, the DSM architecture is designed in such a way that any gradient-free optimisation algorithm can be leveraged.

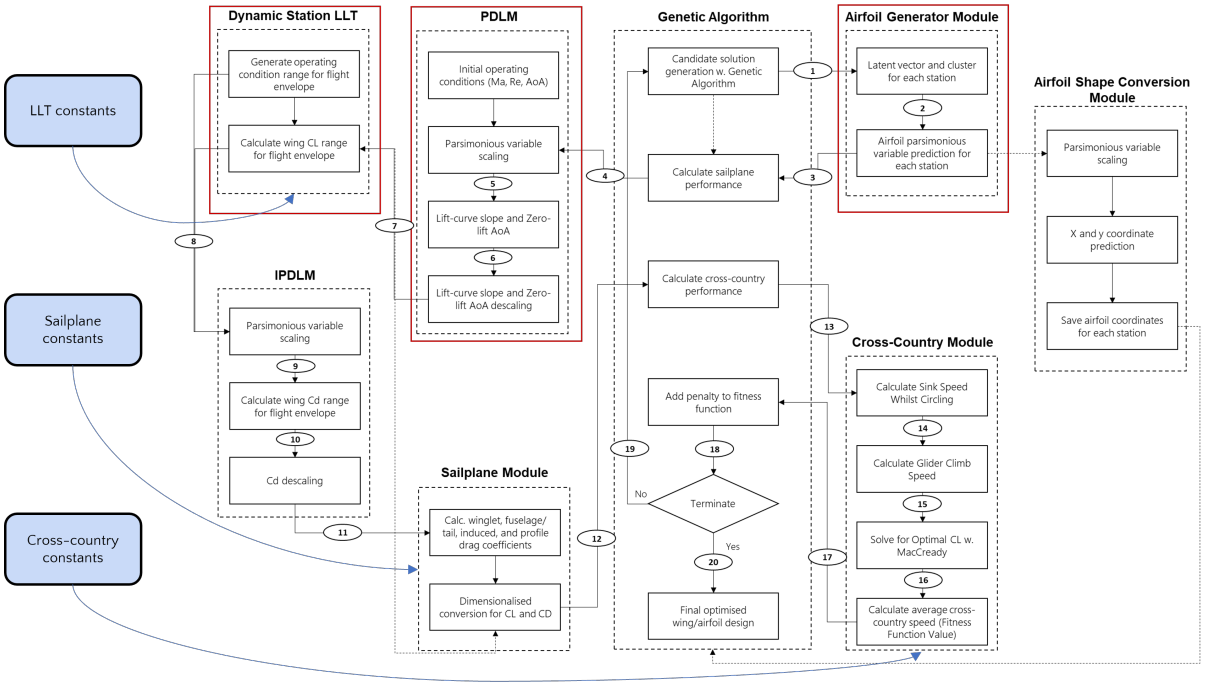


Figure 9.1: Process map of the end-to-end cross-country performance optimisation process with the DSM. Red boxes indicate the modules upon which individual intuitive constraints can be imposed to control the airfoil profiles, sectional 2D airfoil performance characteristics, and wing geometry of each station.

9.2.1 Optimisation process

Using the GA-based approach depicted in Figure 9.1, the optimisation module's process flow can be summarised as follow:

1. **Problem definition:**

   (a) **Sailplane constants (Section 3.4.3)**:

   - Define a sailplane weight.[1]
   - Specify a fuselage/tail configuration performance curve.

   (b) **Dynamic station LLT constants (Section 3.4.1)**:

[1]It will later be shown how this parameter is used to ensure that wing loadings between the optimised design and the baseline design are equivalent to facilitate performance comparison. This is referred to in this research as wing loading normalisation.

- Set the number of wing subdivisions.
- Define the wing twist angle.

(c) **Cross-country constants (Section 2.2.5)**:

- Specify the thermal radii matrix.
- Define a cross-country distance.
- Specify the number of thermals and the thermal types.
- Define the flight proportion allocation per thermal.

2. **Wing design constraints:**

(a) For every station along the wing span and every parsimonious variable vector of interest, define the airfoil lower and upper constraints according to Equation 8.1.

(b) For every airfoil along the wing span and every response variable of interest, define the 2D performance lower and upper constraints according to Equation 8.2.

(c) For every station along the wing span each with its own station geometry, define the respective lower and upper airfoil chord and length constraints according to Equation 8.3.

3. **Solution generation:**

(a) Using the GA, generate $\mathbf{X} \in [-1, 1]^{N \times 29}$, and $\mathbf{Y} \in [0, 49]^{N \times 1}$, defining the airfoil profiles at each of the station N stations along the wing span, with X the latent vector for the airfoil generator module, and Y the airfoil cluster upon which airfoil generation for each station is conditioned.

(b) Using the GA, generate $\mathbf{L} \in (0, \infty)^{N \times 1}$, and $\mathbf{C}^{N \times 1} \in (0, \infty)$, defining the individual wingspan increments and chords respectively, at each of the N stations along the wing span.

4. **Solution penalisation:**

(a) Add an airfoil geometry, wing geometry, or 2D airfoil performance penalty score if any of the constraints are breached after respective evaluation with the airfoil generator module, the dynamic station LLT, or the PDLM (see Section 9.2.2).

5. **Solution evaluation:**

   (a) Offload the generated candidate solution and the problem definition to the DSM for calculation of the objective function, i.e. average cross-country performance.

   (b) If applicable, add the respective airfoil geometry, wing geometry, or 2D airfoil performance penalty scores to the objective function value.

6. **Solution termination:**

   (a) If convergence is met, as defined in Table 9.1, then terminate the optimisation process.

   (b) Save:

   - The final optimised wing design as generated with the GA.
   - The parsimonious variables of the optimised run, as generated by the airfoil generator module.
   - The raw airfoil x and y coordinates of the optimised run, as generated by the airfoil shape conversion module.

   (c) If convergence is not met, iterate steps (1)–(6).

In the optimisation module, the GA parameter settings of Table 9.1 are used. These parameters are determined by iteratively exploring a wide range of input combinations while monitoring the convergence speed, quality of solutions, and computational resources required to maximise solution diversity and balance exploration versus exploitation.

Table 9.1: GA parameters used in the optimisation module

Parameter	Discussion	Value
Maximum number of generations	The maximum number of generations that the genetic algorithm will run before terminating.	50
Total population size	The total number of candidate solutions, (i.e. chromosomes) in each generation.	500
Mutation probability	The probability that a gene in a chromosome will mutate during the reproduction process.	0.4
Elite ratio	The proportion of the population that will be preserved from one generation to the next without being subjected to genetic operations.	0.01
Crossover probability	The probability that a crossover operation will be applied during reproduction to create new candidate solutions.	0.6
Parents proportion	The proportion of the population that will be selected as parents for the reproduction process.	0.3
Crossover type	The type of crossover operation that will be applied during reproduction. In this case, the uniform crossover is used, which selects genes randomly from the parents to create new offspring.	Uniform
Early stopping	The number of iterations without any improvement in the fitness of the population that will trigger termination of the algorithm.	5

9.2.2 Optimisation penalties

As mentioned earlier in Section 9.2.1, penalties are added to the objective function if the solutions' respective airfoil parsimonious variables, wing geometry, or sectional 2D airfoil performance do not adhere to the upfront defined constraints. In the case that any of the constraints are breached, Equations 9.1–9.3 are used to calculate the respective penalty values.

In Equation 9.1, the total normalised airfoil parsimonious variables penalty, P_f, is shown. Here $f_{g_{i,j}}$ is the generated j^{th} parsimonious variable for the i^{th} airfoil, and $f_{a_{i,j}}$ and $f_{b_{i,j}}$ represent the lower and upper bounds respectively. The variable λ_{pi} represents a constant that typically takes the value of 1 but can be adjusted to place more emphasis on a specific constraint for a particular airfoil. Here, the penalty is normalised to ensure the value of λ_{pi} scales relatively across all penalty functions.

$$P_f = \sum_{i=1}^{5} \sum_{j=1}^{23} \begin{cases} \left| \frac{f_{a_{i,j}} - f_{g_{i,j}}}{f_{g_{i,j}}} \right|^2 \cdot \lambda_{p,i} & \text{if } f_{g_{i,j}} < f_{a_{i,j}} \\ \left| \frac{f_{b_{i,j}} - f_{g_{i,j}}}{f_{g_{i,j}}} \right|^2 \cdot \lambda_{p,i} & \text{if } f_{g_{i,j}} > f_{b_{i,j}} \\ 0 & \text{if } f_{a_{i,j}} \leq f_{g_{i,j}} \leq f_{b_{i,j}} \end{cases} \tag{9.1}$$

In Equation 9.2, the total normalised penalty for the airfoil 2D performance characteris-

tics, P_s, is shown. Here $s_{g_{i,j}}$ is the generated j^{th} 2D performance characteristic for the i^{th} airfoil, $s_{l_{i,j}}$ and $s_{u_{i,j}}$ represent the lower and upper bounds respectively, and λ_{si} a constant to place more importance on the 2D performance constraints of specific airfoils.

$$P_s = \sum_{i}^{5}\sum_{j}^{8}\begin{cases}\left|\frac{s_{l_{i,j}}-s_{g_{i,j}}}{s_{g_{i,j}}}\right|^2 \cdot \lambda_{s,i} & \text{if } s_{g_{i,j}} < s_{l_{i,j}} \\ \left|\frac{s_{u_{i,j}}-s_{g_{i,j}}}{s_{g_{i,j}}}\right|^2 \cdot \lambda_{s,i} & \text{if } s_{g_{i,j}} > s_{u_{i,j}} \\ 0 & \text{if } s_{l_{i,j}} \leq s_{g_{i,j}} \leq s_{u_{i,j}}\end{cases} \tag{9.2}$$

Finally, Equation 9.3 represents the wing normalised geometry penalty P_w. Here $w(l)$ denotes the wing length and $w(c)$ denotes the wing chord and hence, $w(l)_{g_{i,j}}$ and $w(c)_{g_{i,j}}$ represents the generated j^{th} length and chord respectively for the i^{th} wing station. Similar to Equation 9.2 $l_{i,j}$, and $u_{i,j}$ represent the lower and upper constraints respectively. However, here there is a lower and upper constraint defined for each $w(l)_{g_{i,j}}$ and $w(c)_{g_{i,j}}$. Also in this equation, λ_{wi} is a constant value used to place emphasis on a specific wing's station characteristics.

$$\begin{aligned} P_w = &\sum_{i}^{5}\sum_{j}^{5}\begin{cases}\left|\frac{w(l)_{l_{i,j}}-w(l)_{g_{i,j}}}{w(l)_{g_{i,j}}}\right|^2 \cdot \lambda_{s,i} & \text{if } w(l)_{g_{i,j}} < w(l)_{l_{i,j}} \\ \left|\frac{w(l)_{u_{i,j}}-w(l)_{g_{i,j}}}{w(l)_{g_{i,j}}}\right|^2 \cdot \lambda_{s,i} & \text{if } w(l)_{g_{i,j}} > w(l)_{u_{i,j}} \\ 0 & \text{if } w(l)_{l_{i,j}} \leq w(l)_{g_{i,j}} \leq w(l)_{u_{i,j}}\end{cases} \\ &+\sum_{i}^{5}\sum_{j}^{5}\begin{cases}\left|\frac{w(c)_{l_{i,j}}-w(c)_{g_{i,j}}}{w(c)_{g_{i,j}}}\right|^2 \cdot \lambda_{s,i} & \text{if } w(c)_{g_{i,j}} < w(c)_{l_{i,j}} \\ \left|\frac{w(c)_{u_{i,j}}-w(c)_{g_{i,j}}}{w(c)_{g_{i,j}}}\right|^2 \cdot \lambda_{s,i} & \text{if } w(c)_{g_{i,j}} > w(c)_{u_{i,j}} \\ 0 & \text{if } w(c)_{l_{i,j}} \leq w(c)_{g_{i,j}} \leq w(c)_{u_{i,j}}\end{cases}\end{aligned} \tag{9.3}$$

With Equations 9.1–9.3, the further away the candidate solution value is from the defined constraint, the larger the penalty function. Here the penalties are normalised according to their respective generated solution values, meaning that the λ parameters can be set between 0 and 1 to place greater emphasis on the deviation of a specific constraint. Seeing that deviation from the desired airfoil profiles is of greater importance than wing geometry or 2D airfoil performance deviation, λ_{pi} is set to 1 for all cases, and λ_{si} and λ_{wi} are set to 0.1. It is important to note that, with this constraint definition, if a variable is to be unbounded in the upper or lower limit, a sufficiently large or small number should be defined to ensure no constraint is breached. The total constraint penalty can therefore be

expressed as:

$$P_T = P_f + P_s + P_w \tag{9.4}$$

Seeing that the GA is free to produce wing planforms in any random form, impractical wing geometry can result. Therefore, a penalty P_P is added to guide the optimisation module in delivering practical wing planforms. This function is expressed so that cases are penalised according to Equation 9.5 when successive chords are larger than the previous chords.

$$P_P = \begin{cases} \left[\sum_i^n x_i - x_{i-1} \right] * \lambda_i & if \quad (x_i - x_{i-1}) < 0 \\ 0 & if \quad (x_i - x_{i-1}) \geq 0 \end{cases} \tag{9.5}$$

where x_i refers to the i^{th} station along the wing span, starting at the fuselage and proceeding to the wing tip, and λ_i refers to a constant that dictates the importance of the specific constraint. In this case, the λ_i value is chosen to be 0.5 seeing that impractical wing planforms are of great importance to the optimisation module.

The final objective function is therefore defined as:

$$\zeta = -1 * \left[\nu + P_T + P_p \right] \tag{9.6}$$

where ν is the average cross-country speed. Here, the objective function is negated seeing that the optimisation module frames the optimisation task as a minimisation problem.

9.3 Optimisation module verification

In this section, the efficacy of the optimisation module detailed in Section 9.2 is evaluated. This is done by evaluating the optimisation outputs of two cases, namely, a thick airfoil case and a thin airfoil case. For both the thick and the thin airfoil optimisation cases the following elements are fixed: (1) the sailplane constants, (2) the dynamic station LLT constants, and (3) the cross-country constants. Thus, the only difference between the two cases are in the airfoils used at each station. In case one, a thick airfoil is used at each station, and in case two a thin airfoil is used at each station.

Therefore, for the optimisation module verification using these two airfoil cases, only the wing geometry is allowed to vary in an attempt to find improvements in cross-country performance. This performance increase is measured relative to a baseline design as discussed in the following section.

9.3.1 Optimisation case definition

Before the functioning of the optimisation module can be verified, a baseline design is required for both the thick and thin airfoil cases. This baseline design forms the foundation of comparison for the optimisation outputs and facilitates the setting of the upper and lower limits of the wing planform constraints.

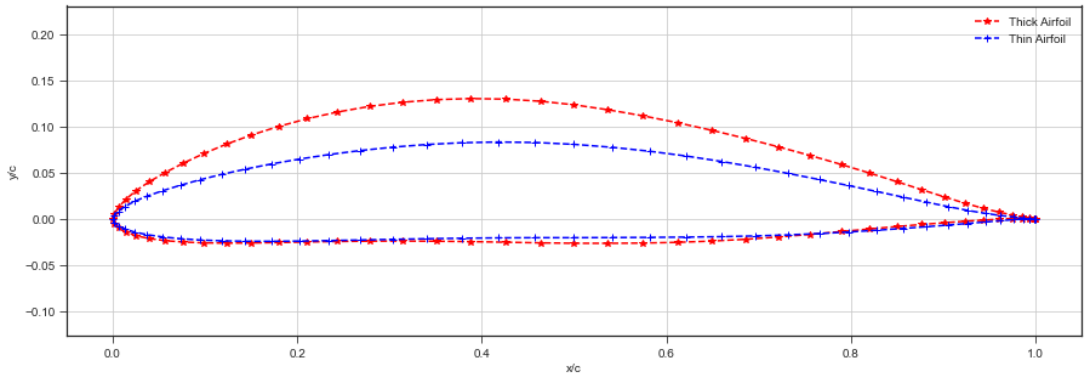
In this section, a derivation of the JS3 sailplane is used as the baseline design. For this design we make use of the Althaus AH 80-129 airfoil at each wing station. As mentioned in Section D.1, the JS3 sailplane has proven to be very effective at the world gliding championships, taking the first six podium positions in 2022. The actual JS3 design is therefore a well-established sailplane, superior in its class, and it is expected that it will be difficult to improve on it. However, due to intellectual property the actual JS3 sailplane design specifications for the wing and airfoil cannot be used. Therefore, a derivation of the actual JS3 sailplane's wing planform and airfoil profile is used in this research. This derivation is somewhat of a simplification but follows the same design patterns and therefore has similar, favourable, cross-country performance characteristics. This baseline design, the translation of the baseline design to be compatible with both the TSM and DSM models, the definition of the cross-country flight task, and the simulation constants can be seen in Appendix D.

As mentioned, two distinctively different airfoils, derived from the Althaus AH 80-129 airfoil, are used to define the thick and thin airfoil cases, respectively. For this airfoil derivation, the geometric design subroutine of XFOIL is used.

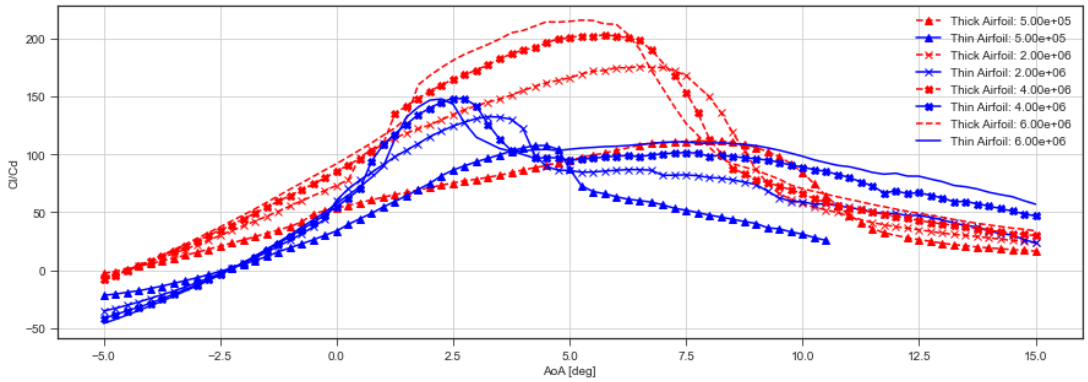
For the thick airfoil case, the thickness and camber ratios are set to 1.25 respectively. This means that the derived thick airfoil has a maximum thickness of 0.1597 at $x = 0.391$

and a maximum camber of 0.0528 at $x = 0.428$. For the thin airfoil case, the thickness and camber ratios are set to 0.85 and 0.75 respectively. The thin airfoil, therefore, has a maximum thickness of 0.1086 at $x = 0.391$ and a maximum camber of 0.0316 at $x = 0.428$.

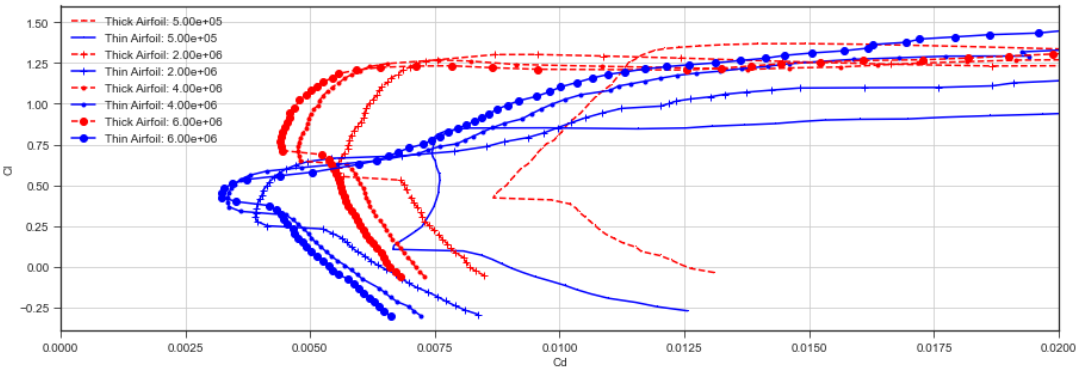
The airfoil profiles, drag polars, and the respective lift-to-drag curves resulting from XFOIL simulations with Reynolds numbers between 0.5×10^6 and 6×10^6, a Mach number of 0.3, and an AoA range of -5° to 15°, can be seen in Figure 9.2. From this figure, it can be observed that: for the majority of the AoA and Reynolds number simulations, the thick airfoil has higher lift-to-drag ratios than the thin airfoil. This is especially true for the AoA cases between minus five and eight degrees. It is also noted from this figure that the thick airfoil has delayed stall conditions when compared to the thin airfoil case.



(a) Airfoil profiles for the thick and thin airfoils derived from the seed (Althaus AH 80-129 airfoil) airfoil



(b) Lift-to-drag ratios over multiple angles of attack and Reynolds numbers for the thick and thin airfoil profiles



(c) C_l vs C_d over multiple angles of attack and Reynolds numbers for the thick and thin airfoil profiles

Figure 9.2: High and low lift airfoil profiles and their corresponding lift-to-drag performance curves and polar plots

With the airfoils defined, the next step is to establish a baseline performance for these two airfoil cases. These baselines set the performance benchmark against which the optimisation results are measured. Here the baseline results are gathered by means of two TSM cross-country simulations; one for each airfoil case. In each simulation, the flight task definition and constants as defined in Section D.4 and the JS3 baseline design are used – the only difference being in the airfoil definition used for each station along the wing span.

The drag results for each respective baseline cross-country simulation can be seen in Figure 9.3. From this figure, it is observed that there is a section between 85 km/h and 120 km/h where the thick airfoil with the baseline wing outperforms the thin airfoil with the baseline wing. From this point onward the thin airfoil case outperforms the thick airfoil case in terms of total drag for the flight envelope analysed. This lower drag aids in the cross-country climbing process between phases and hence we see a higher average cross-country speed for the thin airfoil case as compared to the thick airfoil case. Specifically, for the thick baseline TSM simulation, the average cross-country speed is 97.09 km/h; for the thin baseline TSM simulation, the average cross-country speed is 101.13 km/h.

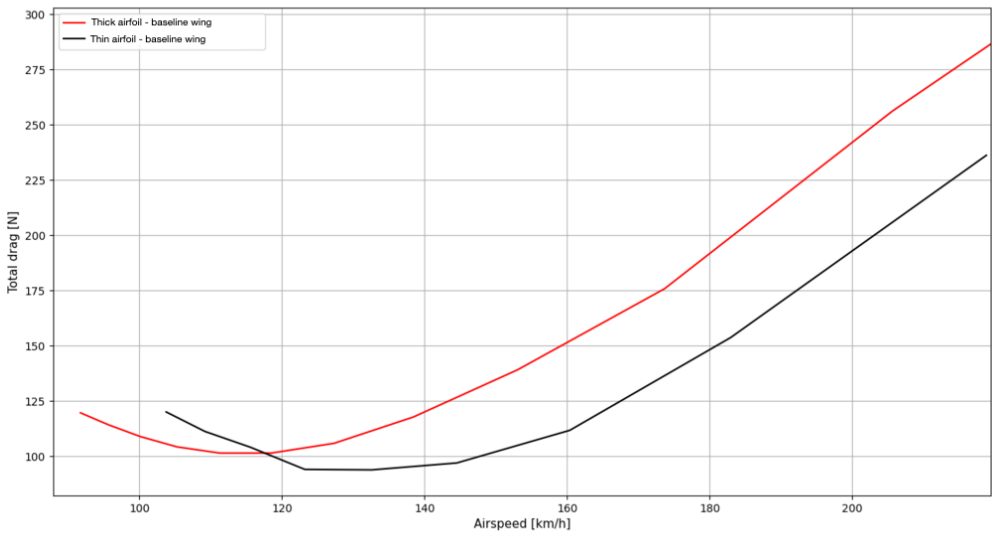


Figure 9.3: Total drag for the thick- and thin airfoils with the baseline planform

For the DSM optimisation routine for the respective airfoil cases, the thick and thin airfoils are translated to the DSM model format by means of the same processes detailed in Section D.3. This ensures that the same airfoil design used in the baseline TSM simulations are also set up in the optimisation module, when converting from the parsimonious domain to the x and y coordinate domain.

The constraints set on the wing geometry as input to the optimisation module are as seen in Table 9.2. Here, any deviance from the wing upper and lower constraints are penalised as in Equation 9.3 and the impractical wing shapes are penalised as in Equation 9.5. Seeing that the objective here is to optimise the planform only, whilst keeping the airfoils fixed, no penalties are required for 2D airfoil performance or deviance from the desired airfoil parsimonious features.

Table 9.2: Upper and lower bounds for station chords and wingspan increments, for the fixed airfoil, varying planform optimisation objective

Station	Chord length (m)			Wingspan increment (m)		
	Baseline	Lower bound	Upper bound	Baseline	Lower bound	Upper bound
1	0.75	0.30	1.25	1.60	1.00	2.75
2	0.72	0.25	1.00	2.60	0.75	2.50
3	0.62	0.25	0.80	1.80	0.75	2.25
4	0.48	0.20	0.80	2.17	0.50	1.75
5	0.32	0.15	0.75	0.89	0.50	1.50
6	0.20	0.15	0.75	N/A	N/A	N/A

As a search and optimisation method for both validation cases, a metaheuristic evolutionary algorithm as described in Section 9.2.1 is used. For both cases, the same parameter settings are used for the GA and are defined as in Table 9.1. The optimisation results for the two cases and their related discussions are detailed in the following section.

9.3.2 Optimisation results and discussion

The optimisation progression, over the subsequent generations, for both airfoil cases can be seen in Figure 9.4. Here, it is observed that optimisation is terminated earlier for the thick airfoil case with only 418 of the generated candidate solutions having feasible

wing planforms. This is compared to 1 006 of the generated candidate solutions having practical wing planforms for the thin airfoil optimisation case. Also in this figure, it can be seen that the thick airfoil optimisation process has a more rapid ascent to the local optimal with significant improvement in cross-country performance seen after 200 iterations and the final solution found 200 iterations thereafter. In the case of the thin airfoil optimisation, there is a more gradual exploration versus exploitation process in play, and, although the same incline in performance improvement can be seen after 200 iterations, it takes another 800 iterations to find the local optima. From this figure it can also be observed that it takes more than 400 iterations in both optimisation cases to find, similar, or superior performance to the respective baseline designs.

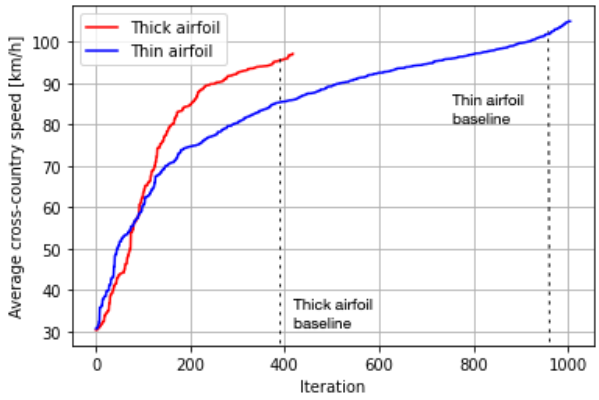


Figure 9.4: DSM optimisation results for the thick and thin airfoil objectives, with specific mention of the respective baseline cross-country performance

To ensure accurate comparisons are made, the cross-country performance of the most optimal designs for both the thick and thin airfoil cases are simulated with the TSM. All further comparisons made in this section are therefore based on the TSM results rather than the DSM results.

In the case of the thin airfoil, the TSM model predicts a 104.4 km/h average cross-country speed compared to the DSM prediction of 110.68 km/h. This means the absolute percentage error between the two models is 6.02%. It can be seen here that both the TSM results and the DSM results for the thin airfoil case are superior to the relevant baseline cross-country performance. For the TSM cross-country performance of the optimised thin

airfoil case, a total cross-country increase of 3.27 km/h is realised when compared to the baseline.

In the case of the thick airfoil, the DSM optimised cross-country speed is 98.57 km/h. When simulated with the TSM, this value equates to 97.44 km/h, meaning that there is a 1.13 km/h difference between the two models. This means, for the thick airfoil case, the optimisation module was able to achieve a 1.16% increase in cross-country performance when compared to the baseline performance. In both the thin airfoil and the thick airfoil cases, the DSM optimised results are superior to the baseline results when simulated with the TSM.

An important note here is in the discrepancy between the DSM predicted performance and the TSM predicted performance. It is true that in both airfoil cases the optimisation module is able to improve baseline performance when validated with the TSM, however, the impact of the DSM error rate can be seen here. Specifically in the thick airfoil case where, once validated, the perceived improvements are significantly lower than the predicted improvement during the optimisation phase. When optimising SOTA standard class glider designs, this error rate of the underlying DSM needs to be lowered to ensure that the optimisation module predicted performance increase translates to similar performance increases when validated with the TSM. This lower error rate will also ensure more efficient optimisation seeing that the optimisation module will explore and exploit designs that actually lead to improved performance.

Furthermore, the cross-country findings can be explained by the lift-to-drag ratio curves of the thick airfoil and thin airfoil optimised planforms and their respective baselines, as seen in Figure 9.5. From this figure, it is observed that the thin airfoil case outperforms the baseline case for the speed ranges of 100 km/h up to 137 km/h and has similar performance from thereon out. For the case of the thick airfoil, here it is observed that in both baseline and optimised cases, similar performance is seen with the optimised case being slightly outperformed by the baseline case.

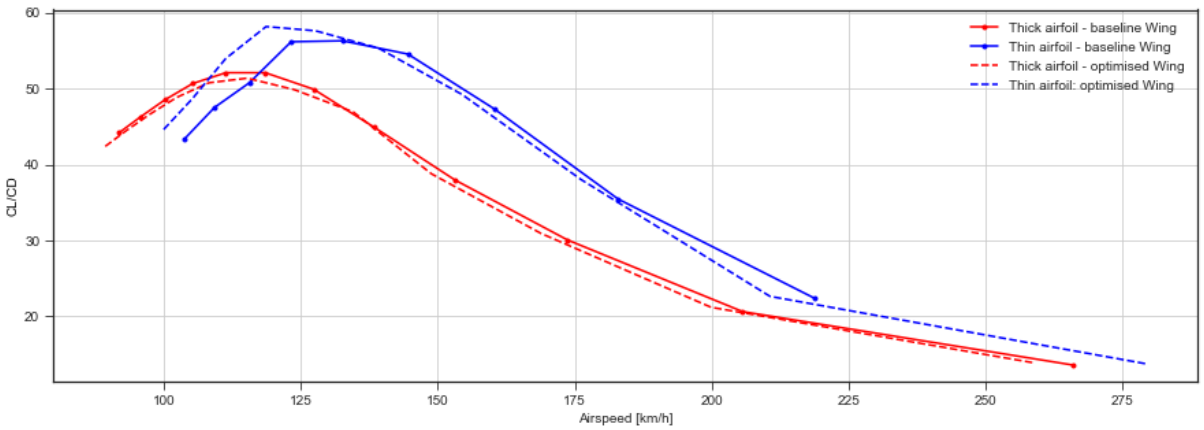


Figure 9.5: Lift-to-drag ratios for the thick and thin airfoil optimisation cases

Table 9.3 shows the thermal performance of the optimised thick and thin airfoil cases along with their respective benchmarks. Here, it is seen that the thin airfoil has, on average, 20.55% less drag and 12.12% higher lift-to-drag performance across the 4 thermal models. It is also seen that the thin optimised design, on average, climbs 5.4% faster than the thin airfoil baseline. For the thick airfoil case, in Table 9.3 it is observed that, although the optimal design has on average 3.49% less drag, it has on average 2.09% lower lift-to-drag ratio. This means more height is required and therefore a slightly longer time for each phase. However the differences here are minor, and therefore there is similar cross-country performance between the optimised thick airfoil case and its baseline design.

Table 9.3: Optimisation results, for the thick and thin airfoil optimisation cases, broken down by thermal properties

Thin airfoil - optimised						
Thermal	Speed (km/h)	Lift/Drag	C_D	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	130.33	56.48	0.0117	0.25	0.00	0.00
A2	188.25	32.10	0.0099	1.86	6382.98	6133.65
B1	155.29	50.03	0.0093	0.67	246.40	477.97
B2	199.04	28.06	0.0101	2.38	4885.89	3966.78
Thin airfoil - baseline						
Thermal	Speed (km/h)	Lift/Drag	C_D	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	129.55	49.00	0.0136	0.38	0.00	0.00
A2	179.75	27.92	0.0124	2.06	6088.27	6360.14
B1	146.04	41.21	0.0128	0.81	242.44	545.23
B2	188.41	25.03	0.0126	2.51	4794.49	4205.95
Thin airfoil - percentage difference						
Thermal	Speed (km/h)	Lift/Drag	C_D	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	0.60	0.00	-14.29	-34.10	0.00	0.00
A2	4.73	14.94	-20.68	-9.69	4.84	-3.56
B1	6.33	21.42	-27.16	-17.64	1.63	-12.34
B2	5.65	12.12	-20.08	-5.02	1.91	-5.69
Average	4.33	12.12	-20.55	-16.61	2.09	-5.40
Thick airfoil - optimised						
Thermal	Speed (km/h)	Lift/Drag	C_D	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	135.06	46.61	0.0124	0.51	0.00	0.00
A2	180.38	26.62	0.0122	2.19	6387.74	6309.26
B1	145.38	40.53	0.0123	0.83	246.45	545.95
B2	186.70	24.51	0.0124	2.53	4895.23	4248.39
Thick airfoil - baseline						
Thermal	Speed (km/h)	Lift/Drag	C_D	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	129.55	49.00	0.0136	0.38	0.00	0.00
A2	179.75	27.92	0.0124	2.06	6088.27	6360.14
B1	146.04	41.21	0.0128	0.81	242.44	545.23
B2	188.41	25.03	0.0126	2.51	4794.49	4205.95
Thick airfoil - percentage difference						
Thermal	Speed (km/h)	Lift/Drag	C_D	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	4.26	0.00	-8.82	34.27	0.00	0.00
A2	0.35	-4.69	-1.76	6.32	4.92	-0.80
B1	-0.45	-1.63	-3.27	1.80	1.66	0.13
B2	-0.91	-2.06	-1.96	0.97	2.10	1.01
Average	0.81	-2.09	-3.95	10.84	2.17	0.08

In Table 9.4 the respective wing planforms can be seen compared to the baseline planform and respective upper and lower constraints. Here it is noted that for both the thick and thin airfoil optimisation cases, the wing chords and wingspan increments are within the set upper and lower bounds. This indicates that the optimisation module is effectively able to find wing planforms that yield desirable performance whilst adhering to upfront defined constraints. Seeing that no constraints are set on the total span length, it is interesting to note that the optimisation module tended to exploit this fact and yielded wing designs with individual wingspan increments close to the upper bounds. This is

aligned with known sailplane design patterns where an increase in span generally results in an increase in cross-country performance. The fact that the model learned this behavior implicitly instills confidence in the inner mechanics of the optimisation module. In both the thick and thin airfoil cases the total wing length is more than 18 metres with 18.2 m and 19.2 m respectively. The aspect ratio for the thin airfoil optimisation case is 34.23 with a wing area of 10.76 m^2. For the thick airfoil case, the aspect ratio is 31.24 and the total wing area is 10.58 m^2.

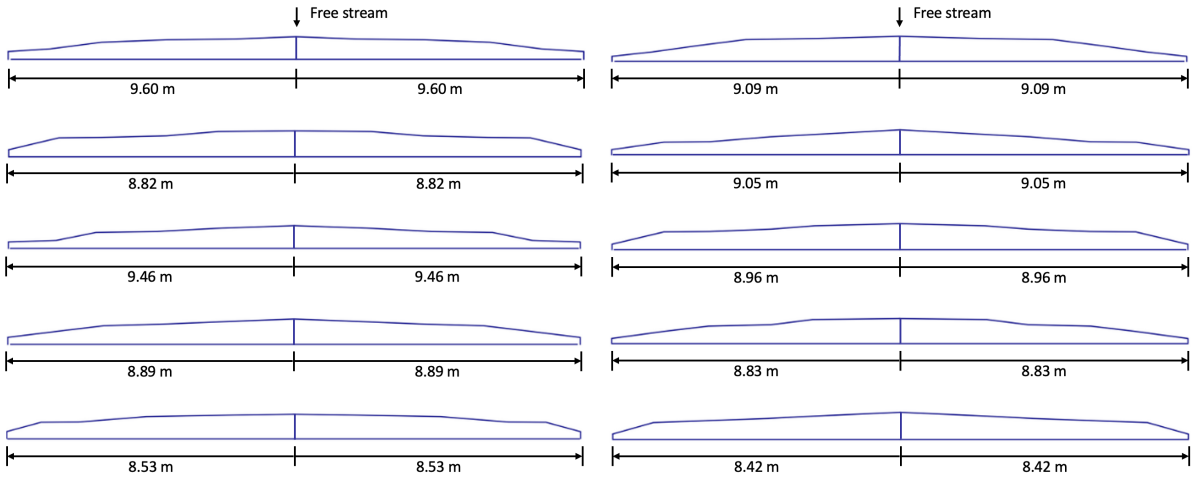
In Figure 9.6 the five respective planforms for the top-performing thick and thin airfoil cases are shown. For the thin airfoil case, the average length and aspect ratio across these five cases is 18.11 m and 31.67 respectively. For the thick airfoil, these values are 17.74 and 31.57 respectively. Here it is noted that, although the top-performing planforms shared similar aspect ratios, the thick airfoil planforms tended to be shorter with less surface area (10.01 m^2 on average) compared to the thin airfoil case which had longer spans with larger surface areas (10.41 m^2 on average). This is an interesting finding seeing that it again is aligned with priors where airfoils that generate more lift require less area. From Figure 9.2 it is clear that the thick airfoil generates more lift across all cases investigated which explains the slightly reduced average area requirement. The fact that the optimisation module is able to map this complex relationship therefore again gives confidence in its constrained optimisation capability.

Table 9.4: Planform designs of the optimal thick and thin optimisation cases compared to the respective upper and lower bounds and the baseline design planform

Station	Chord length (m)					Wingspan increment (m)				
	Baseline	Lower bound	Upper bound	Thick-opt.	Thin-opt	Baseline	Lower bound	Upper bound	Thick-opt.	Thin-opt
1	0.75	0.30	1.25	0.79	0.74	1.60	1.00	2.75	2.56	2.07
2	0.72	0.25	1.00	0.72	0.66	2.60	0.75	2.50	2.30	2.21
3	0.62	0.25	0.80	0.69	0.64	1.80	0.75	2.25	1.48	2.20
4	0.48	0.20	0.80	0.50	0.56	2.17	0.50	1.75	1.47	1.71
5	0.32	0.15	0.75	0.29	0.34	0.89	0.50	1.50	1.28	1.41
6	0.20	0.15	0.75	0.15	0.25	N/A	N/A	N/A	N/A	N/A

From the findings in this section, it can therefore be concluded that the proposed optimisation module enables efficient constrained optimisation of a given wing/airfoil combination. This module is therefore deemed fit for use in a more complex multi-objective constrained optimisation problem, where both the planform and the airfoils are allowed to vary with

constraints set on both. This case study can be seen in the following section.



(a) Optimised wing geometry for the thin airfoil optimisation case

(b) Optimised wing geometry for the thick airfoil case

Figure 9.6: Top 5 respective wing planforms as optimised for by the DSM model for both the thick and thin airfoil cases

9.4 Constrained optimisation module verification

In this section, the proposed optimisation modules' ability to perform multi-objective optimisation, with intuitive constraints set on both the individual airfoil profiles as well as the wing planform, is validated. First, the optimisation problem itself is described in Section 9.4.1, before the wing and airfoil constraints are defined. Next, in Section 9.4.2, the results of this optimisation routine are discussed with specific mention of the optimisation progress through the successive iterations and the adherence of the optimised result to the upfront defined wing and airfoil design constraints. All of this is done to ascertain if the optimisation module: (1) is functioning as intended, (2) makes the appropriate design decisions, and (3) adheres to all upfront user-defined constraints.

9.4.1 Optimisation problem definition

For the multi-objective optimisation problem, the JS3-based model is once again chosen as the baseline design. However, in this case, instead of fixing the airfoil the optimisation module is allowed to optimise the airfoil alongside the wing planform – with multiple constraints set on the optimised airfoil profile and wing planform. Readers are again referred to Appendix D for the specification of the baseline sailplane design and cross-country definition as this is the setup once again used here. For the constraint definitions for the upper and lower wing chords and lengths, again the definitions in Table 9.2 are used. For the penalty of deviation from our desirable station wise chords and lengths, Equation 9.3 is used, and for impractical wing planforms the penalty expression of Equation 9.5 is used.

For the airfoil definition the same airfoil is used at each station along the wing span. That is, every airfoil used at each of the five stations had exactly the same 23 parsimonious variables, as generated with the airfoil generator module. Also, the same constraints are imposed on each of the airfoils along the span and Equation 9.1 is used to penalise any deviance from the desired properties. Here, no constraints are imposed on the sectional 2D airfoil performance characteristics and hence no penalty term is required for the 2D performance deviation case. Using the same airfoil at each station means that the size of the optimisation space decreases from 161 variables (5×29 latent vectors, 5×1 airfoil cluster specifications, 6×1 chords, 5×1 lengths) to 41 variables (1×29 latent vectors, 1×1 airfoil cluster specifications, 6×1 chords, 5×1 lengths). This choice is made here seeing that understanding the optimisation modules decisions in the lower dimensional space is simpler and hence it is easier to identify any discrepancies or deviation from the desired optimisation performance. Later, in Section 10.3, a full optimisation case study is done where all airfoil stations are allowed to have different airfoil settings and constraints.

Seeing that the baseline design makes use of the Althaus AH 80-129 airfoil, this airfoil's specifications are used in the airfoil constraint setting. The constraints are chosen such that: (1) an airfoil with a slightly higher camber situated closer to the TE and, (2) an airfoil with a higher thickness ratio situated closer to the center will be the end result. Exercising such specific control on the optimisation pipeline demonstrates the flexibility and

the simplicity of setting intuitive constraints on airfoil shapes as input to the optimisation module. The chosen constraints for both the shape and the structural characteristics are as in Table 9.5.

Table 9.5: Upper and lower bounds on airfoil shape and structural features for the full end-to-end optimisation process with the developed optimisation module.

Feature	Lower bound	Upper bound
Maximum thickness	0.130	0.150
Maximum thickness location	0.410	0.550
Maximum camber	0.000	0.055
Maximum camber location	0.350	0.500
Maximum top magnitude	0.008	0.135
Maximum top magnitude location	0.410	0.650
Airfoil area	0.080	0.125
Airfoil perimeter	2.000	2.200
Bending inertia about the x-axis	0.004	0.007

Finally, for efficient exploration of the design space, the GA is once again used with the parameter settings as defined in Table 9.1.

9.4.2 Optimisation results and discussion

The optimisation pipeline discussed in the previous section is executed and allowed to run till either convergence is met or early stopping is triggered. In Figure 9.7 the optimisation iterative progress can be seen for this case. From this figure, it can be observed that:

1. Optimisation is initially slow, for the first 95 iterations, with a sharp increase in performance thereafter. This is because, for the initial 95 iterations, the system has not yet gained the required experience on how practical wing designs, specific wing planforms, and airfoil profiles should be defined to avoid penalties. Once this problem is overcome we see a gradual increase as exploration versus exploitation is balanced in the search process to arrive at the local optimum after 800 iterations.

2. Only incremental improvement is seen between 600 and 800 iterations as this is the point where the airfoil designs and wing designs start to stabilise and minor adjust-

ments are made to these respective geometries to gain incremental improvements whilst the constraints are adhered to.

3. After 33 generations i.e. 776 iterations, early stopping is triggered.

4. The optimisation module is able to surpass the performance of the baseline design after 483 iterations.

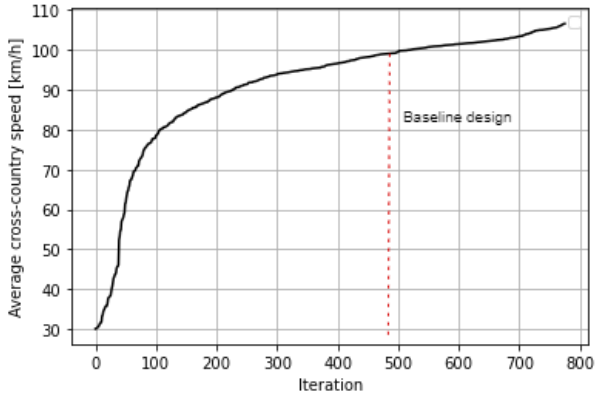


Figure 9.7: Optimisation iteration results for the full multi-objective optimisation process

From Figure 9.7 it is however clear that there are multiple designs that have similar average cross-country performance. Simply selecting the wing/airfoil combination that yields the highest average cross-country speed for the single cross-country task might, from this pool of candidate solutions, mean a suboptimal design is chosen for further downstream analysis. For this reason, a second objective is weighed into the decision of the optimal design choice. This second objective is the average lift-to-drag ratio of the sailplane configuration over the flight envelope present in the cross-country task defined. The DSM is used to generate these metrics separately, seeing that they are not calculated and brought into consideration at the time of optimisation. The results of this analysis can be seen in Figure 9.8. From this figure, it is observed that there are multiple configurations with average cross-country speeds greater than 100 km/h but with variance in the average lift-to-drag ratios from 27 to 60. The optimal choice from this analysis is therefore observed to be the one with a DSM calculated average cross-country speed of 106.59 km/h and an average lift-to-drag ratio of 60.25.

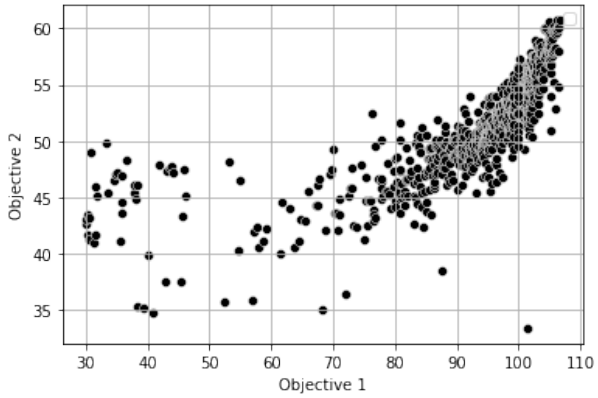


Figure 9.8: Depiction of the Pareto front for objective 1, i.e. average cross-country speed, and objective 2, i.e. average lift-to-drag ratio over the flight envelope

This optimal design is next subjected to an analysis of its average cross-country speed, as compared to the baseline design. All analyses are done with the TSM to ensure that the comparisons and conclusions are accurately drawn. For the optimised case, the DSM calculated an average cross-country speed that is 106.59 km/h whereas the TSM calculates this value to be 102.35 km/h – an absolute difference of 4.29 km/h. However, in both cases, the optimised design achieves superior performance compared to the baseline design with a TSM simulated speed of 97.32 km/h. In Table 9.6, the optimisation results for the baseline and optimised designs are depicted, segregated by thermal type. Here it can be seen that, although the baseline design has higher lift-to-drag ratios for the specified thermals and their respective distance allocations, the optimised design has on average a 2.03% lower time per phase. This is due to the favourable performance seen across the maximum climb speed for the optimal design where it has a significantly higher average climb speed for all four thermal types. It is important to note that the big increase in the A1 thermal type climb speed for the optimised design does not have any influence here seeing that 0% flight time is allocated to this thermal type. From this table, an indication of the superior performance of the optimised design when compared to the baseline design for the specific task at hand is therefore evident.

Table 9.6: Results for the full multi-objective optimisation process, broken down by thermal properties

Optimised design						
Thermal	Speed (km/h)	Lift/Drag	CD	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	136.5	48.417	0.0117	0.589	0.00	0.00
A2	185.9	27.467	0.0111	2.293	6189.66	5992.52
B1	145.7	43.792	0.0113	0.820	228.12	525.07
B2	190.8	26.051	0.0111	2.540	4606.28	4077.16
Baseline design						
Thermal	Speed (km/h)	Lift/Drag	CD	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	129.5	53.829	0.0124	0.288	0.00	0.00
A2	181.3	30.054	0.0113	1.898	5656.73	6355.53
B1	152.3	44.957	0.0107	0.756	222.21	530.20
B2	193.4	25.620	0.0117	2.461	4683.81	4137.25
Percentage difference						
Thermal	Speed (km/h)	Lift/Drag	CD	Maximum climb speed (m/s)	Height Required (m)	Time for phase (s)
A1	5.42	0.00	-5.98	104.42	0.00	0.00
A2	2.51	-8.61	-2.14	20.79	9.42	-5.71
B1	-4.36	-2.59	5.48	8.53	2.66	-0.97
B2	-1.32	1.68	-5.08	3.23	-1.66	-1.45
Average	0.56	-2.38	-1.93	34.24	2.61	-2.03

Next, the sink speed curves of the optimised and baseline designs are analysed to make an informed decision on optimality. Here, the sailplane weight of the optimised design is adjusted to ensure that the same wing loading is perceived in both the baseline and the optimised design TSM simulations. The results of this analysis can be seen in Figure 9.9. From this figure, it can be observed that the optimised design outperforms the baseline design at low speeds, i.e. between 105 km/h to 135 km/h, whereas the baseline design outperforms the optimised design for higher speeds, i.e. 135 km/h to 180 km/h.

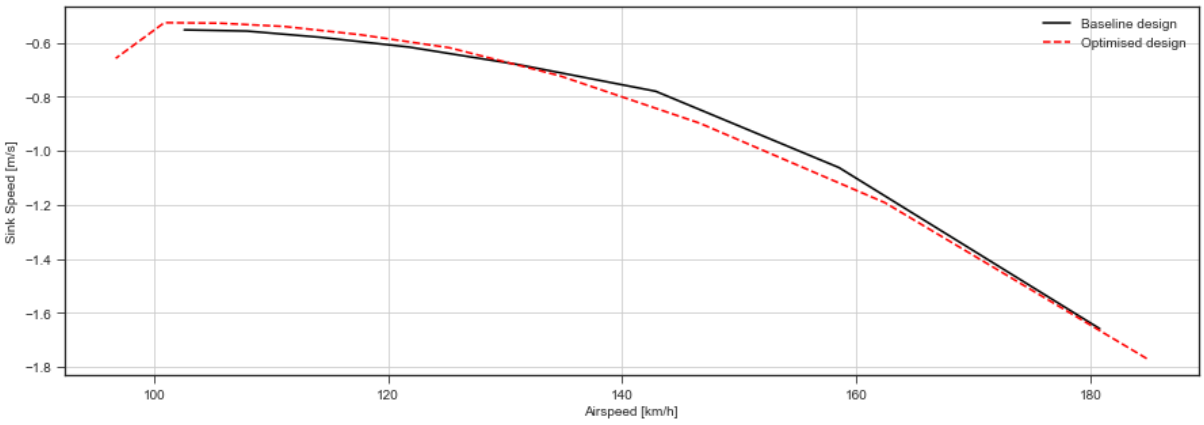


Figure 9.9: Sink speed of the optimised design versus the baseline design

9.5 Constrained optimisation results discussion

To verify the efficacy of the optimisation module is to understand if the design resulting from the optimisation module in Section 9.4.2 is indeed an improvement upon the baseline design whilst adhering to the defined constraints. Therefore, two further validation checks are required. The first is to establish if the airfoil/wing resultant adheres to the defined constraints. The second validation check is to compare the average cross-country speed of the baseline design to that of the optimised design for diverse cross-country flight tasks. These two additional evaluations can be seen in the subsequent sections.

9.5.1 Constraint evaluation

In this section, the airfoil and wing designs are investigated to determine whether they adhere to the upper and lower constraints defined as the input to the optimisation pipeline. This is done by numerically analysing the shape and structural characteristics of the airfoil as well as the individual wingspan increments and chords. Also in this section, the optimisation module's design decisions regarding the wing planform and airfoil designs are commented on.

The first constraint adherence analysis done is for the optimised airfoil. This is done by inspecting the shape and structural characteristics of the optimised airfoil. In this endeavour, the XFOIL-GDES and XFOIL-BEND subroutines are used where possible, otherwise, the calculations in Section 4.4 are employed. The results of this analysis can be seen in Table 9.7 and the final airfoil design, as generated with the airfoil shape mapping module, is depicted in Figure 9.10a. This analysis indicates that the chosen parsimonious airfoil shape and structural properties, that are constrained as input to the optimisation module, have feature values that are within the desired upper and lower bounds.

Table 9.7: Upper and lower parsimonious variable constraints compared to the values of the final optimised airfoil

Feature	Lower bound	Upper bound	Optimised airfoil value
Maximum thickness	0.130	0.150	0.132
Maximum thickness location	0.410	0.550	0.422
Maximum camber	0.000	0.055	0.051
Maximum camber location	0.350	0.500	0.461
Maximum top magnitude	0.008	0.135	0.117
Maximum top magnitude location	0.410	0.650	0.461
Airfoil area	0.080	0.125	0.090
Airfoil perimeter	2.000	2.200	2.047
Bending inertia about the x-axis	0.000	0.007	0.005

Next is the analysis of the constraint adherence of the optimised wing profile. For this analysis, a basic comparison is done between the individual station geometric features and the defined upper and lower bound constraints. The results from this analysis are summarised in Table 9.8. It is also observed from this analysis that both the wing's individual wingspan increments and chords fall within the upper and lower constraints. As can be derived from this table, the optimised wing has an aspect ratio of 37.14, a surface area of 10.62 m^2, and a total length of 19.86 m. This is compared to the baseline wing with an aspect ratio of 32.90, a surface area of 9.98 m^2, and a total length of 18.12 m. Detailed comments about the comparison between these two wing geometries are made later in this section.

Table 9.8: Planform geometry of the optimised design compared to the respective upper and lower bounds and the baseline design planform

Station	Chord length (m)				Wingspan increment (m)			
	Baseline	Lower bound	Upper bound	Optimised	Baseline	Lower bound	Upper bound	Optimised
1	0.75	0.30	1.25	0.74	1.60	1.00	2.75	2.59
2	0.72	0.25	1.00	0.66	2.60	0.75	2.50	2.34
3	0.62	0.25	0.80	0.60	1.80	0.75	2.25	2.22
4	0.48	0.20	0.80	0.40	2.17	0.50	1.75	1.42
5	0.32	0.15	0.75	0.35	0.89	0.50	1.50	1.36
6	0.20	0.15	0.75	0.24	N/A	N/A	N/A	N/A

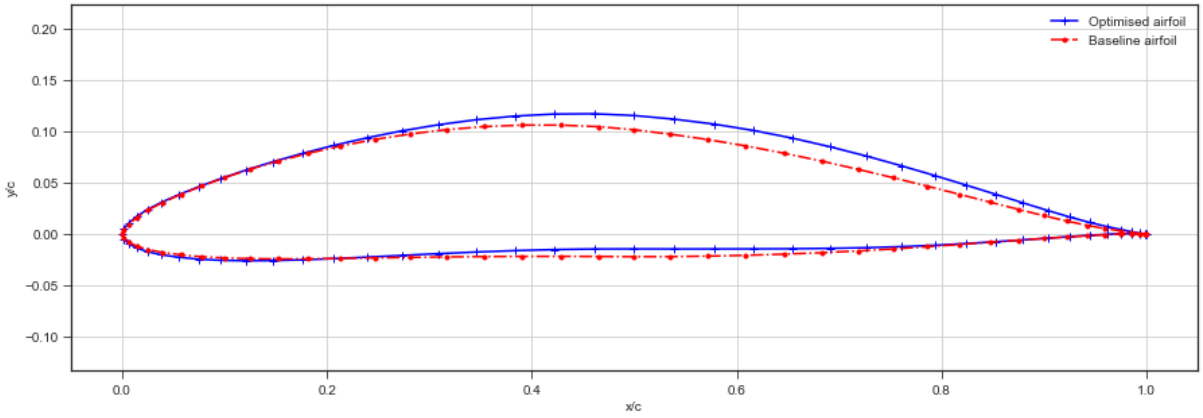
With the optimisation module shown to be able to adhere to all design constraints imposed on the airfoil profiles as well as the wing geometry, the optimisation module's decisions

behind the respective geometric choices are investigated next. First is the analysis of the airfoil; specifically, the 2D airfoil performance characteristics. In this specific cross-country task, the sailplane is subject to Reynolds numbers ranging from 0.5×10^6 to 4×10^6. To gain a full understanding of the respective 2D airfoil performance, four XFOIL simulations are executed for each airfoil varying between $0.5\times10^6 \leq Re < 6\times10^6$ and $-5^\circ \leq AoA < 15^\circ$, with a Mach number of 0.3 in each simulation. The polar curves and the lift-to-drag ratio curves for the optimised airfoil and the baseline airfoil can be seen in Figures 9.10b and 9.10c respectively. From these curves, it is clear that the optimised airfoil outperforms the baseline airfoil for the majority of the flight envelope. This is especially true in the ranges $-5^\circ \leq AoA < 5^\circ$. The optimised airfoil is observed to have similar drag performance over the ranges $-5^\circ \leq AoA < 5^\circ$, with higher lift production, but worse drag performance for the ranges $6^\circ \leq AoA < 15^\circ$ for the same lift production. This is the reason why lower lift-to-drag ratios are seen for this airfoil in the ranges $6^\circ \leq AoA < 15^\circ$ Overall, these simulations highlight the fact that, at an individual airfoil level, the optimised airfoil outperforms the baseline airfoil for the major part of the flight envelope investigated.

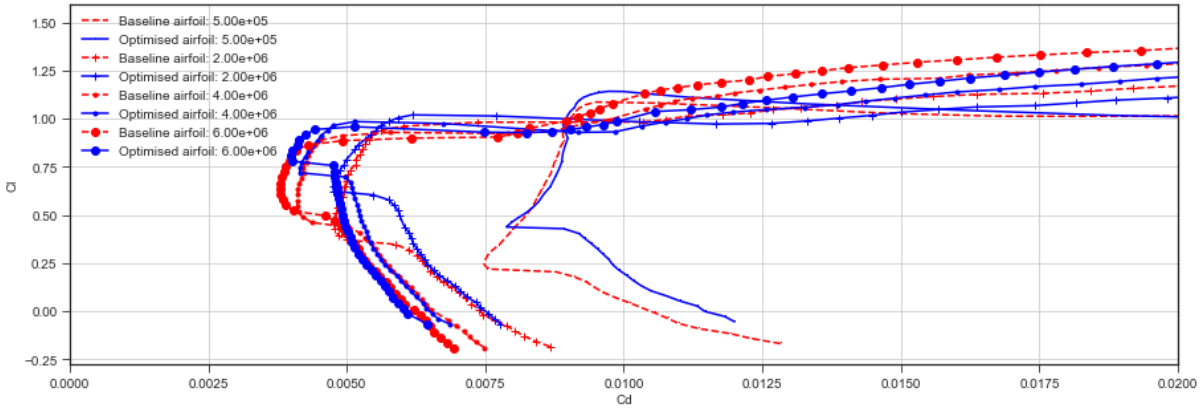
Next is the investigation of the wing planform. From Table 9.8, it is noted that the optimisation module tended to yield wingspan increments that are situated closer to the upper bound extremes. This means the algorithm exploited the well-known fact that having greater span lengths generally will result in better cross-country performance. From this table, it is also observed that the chord lengths are better chosen at the mid to lower end of the constraints. This highlights the fact that the optimisation module tried to balance aspect ratio and area increase – two contrastive objectives. It is also clear that the optimisation module correctly increased the aspect ratio of the optimised planform, seeing that the optimised airfoil produces more lift than the baseline airfoil. This higher aspect ratio planform compared to the baseline planform can be seen in Figure 9.11. From this figure, it is noted that the optimised planform had a similar chord distribution relative to the baseline design and hence a somewhat similar planform shape results, however with differing areas, and aspect ratios. From this figure, it is also clear that the chords are distributed in a successively decreasing sequence resulting in a realistic wing planform.

From the findings in this section, it can be concluded that the optimisation module is able

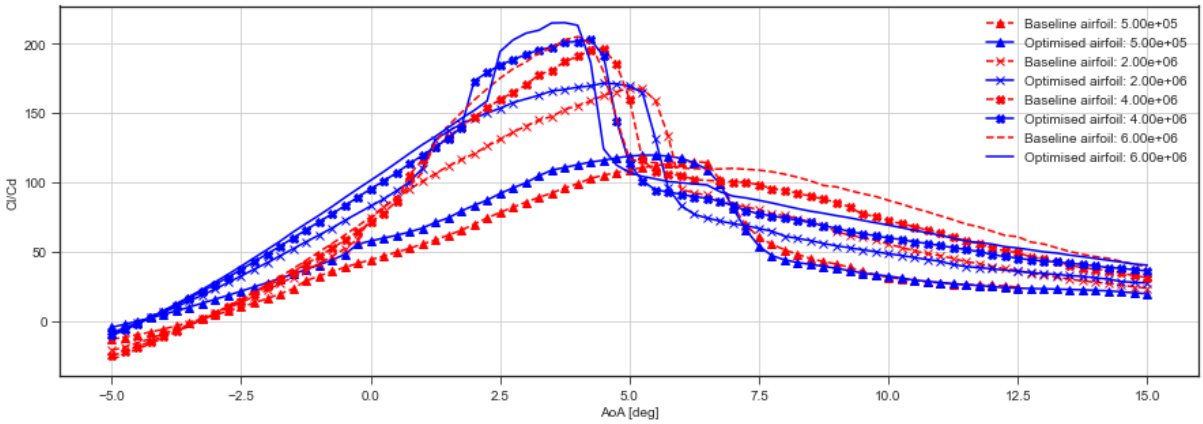
to perform cross-country performance optimisation while adhering to predefined airfoil profile and wing geometry constraints. Also, it is observed that the optimisation module's design decisions are aligned with priors of best practice in sailplane cross-country design.



(a) Baseline and optimised airfoil profile comparison



(b) Polar curves for the baseline and optimised airfoils over $0.5 \times 10^6 \leq Re < 6 \times 10^6$ and $-5° \leq AoA < 15°$, with a Mach number of 0.3

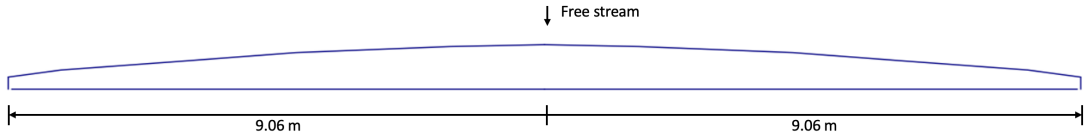


(c) Lift-to-drag ratio for baseline and optimised airfoils over $0.5 \times 10^6 \leq Re < 6 \times 10^6$ and $-5° \leq AoA < 15°$, with a Mach number of 0.3

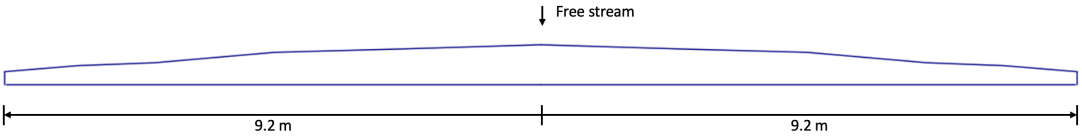
Figure 9.10: Airfoil profile and performance comparison for the optimised and baseline airfoil shapes

Table 9.9: Comparison between the optimised planform and the baseline platform, together with the explicit station geometry upper and lower constraints

Station	Chord length (m)				Wingspan increment (m)			
	Baseline	Lower bound	Upper bound	Optimised	Baseline	Lower bound	Upper bound	Optimised
1	0.75	0.50	0.90	0.87	0.00	0.00	0.00	0.00
2	0.72	0.45	0.85	0.83	1.60	1.40	2.00	1.87
3	0.62	0.40	0.80	0.67	2.60	2.10	2.80	2.57
4	0.48	0.30	0.60	0.52	1.80	1.40	2.00	1.55
5	0.32	0.20	0.50	0.42	2.17	1.60	2.40	2.10
6	0.20	0.15	0.40	0.33	0.89	0.50	1.20	1.11



(a) Baseline wing geometry



(b) Optimised wing geometry

Figure 9.11: Final optimised wing planform compared to the baseline wing planform.

9.5.2 Cross-country performance validation

In this section, the optimised design is subject to more rigorous testing to further comment on optimality and hence verify the efficacy of the proposed optimisation module. The objective here is twofold: (1) establish if the optimised design indeed performs better than the baseline design for the cross-country task defined in the optimisation module, and (2) if so, if the performance increase is only for the baseline optimisation flight task or if the optimised design performs better over a wider range of cross-country tasks.

This is achieved by generating a set of 15 evaluation cases, each with a flight task randomly sampled between 100 km and 800 km, and a chosen percentage allocation for an A1, A2,

B1, and B2 thermal type to ensure a comprehensive blend of strong and weak gliding conditions are present. In the thermal type percentage allocation, only one to four different thermal types can be chosen for a specific case and the percentage allocation has to sum to 100%. The thermal type definitions are as in Table 9.10.

Table 9.10: Thermal radius definitions for the end-to-end TSM model validation

Thermal	Core radius (m)	Vertical Speed (m/s)	Gradient (1/s)
A1	60	1.750	0.025
A2	60	3.500	0.032
B1	60	1.750	0.005
B2	60	3.500	0.006

For each of the 15 validation cases, the TSM is used to determine the average cross-country performance for both the baseline design as well as the optimised design, and a comparison is done to understand how the performance varies between the two designs. In this comparison, the cross-country speed is calculated for the:

1. Baseline design denoted by BW-BF representing baseline wing – baseline airfoil.
2. Optimised design, denoted by OW-OF representing optimised wing – optimised airfoil.
3. Optimised design with normalised wing loading[2] abbreviated as OW-OF(N) representing optimised wing – optimise airfoil(normalised).
4. Baseline wing with the optimised airfoil abbreviated as BW-OF for baseline wing – optimised airfoil.
5. Optimised wing with normalised wing loading and baseline airfoil, denoted by OW-BF(N).

For the analysis to be conducted, the optimised design is translated into the acceptable TSM format using the airfoil shape mapping module.

[2]Here the normalised case refers to where the sailplane's input weight is adjusted to ensure both the optimised (normalised) TSM simulations and the baseline simulations have the same wing loading.

In Table 9.11, the results of the 15 cross-country simulations are depicted for all of the aforementioned design permutations. Also here the characterisation of the thermal combination can be seen and is classified as strong, weak, and mixed, indicating strong, weak, or mixed flight conditions for the specific simulation case. From this table, the following is observed:

1. Across all 15 simulations, the average cross-country speed of the OW-OF design is on average 14.73% higher than the BW-BF design. For the OW-OF(N) design, this value is 10.97%.

2. Seeing that the optimised wing planform has a larger area than the baseline design, it performs significantly better in weak flight conditions. For the OW-OF design, the average cross-country speed for weak flight conditions is on average 25.4% higher than the BW-BF design, whereas this value is 17.9% for the OW-OF(N) design.

3. The worst performance of the OW-OF(N) design is for strong thermalling conditions; however, this wing design outperforms the BW-BF design, on average by 6.1%.

4. When using the baseline wing with the optimised airfoil i.e. the BW-OF design, a similar performance to the OW-OF, and OW-OF(N) design is seen. Specifically, this design outperforms the BW-BF design by 2.9% for strong thermalling conditions, 11.0% for weak thermalling conditions, and 8.7% for mixed thermalling conditions, respectively. This is aligned with the findings depicted in Figure 9.10c, where the optimised airfoil outperforms the baseline airfoil over most of the flight conditions present in the simulated cross-country tasks.

5. For the BW-OF design, the average cross-country performance over all 15 cases is on average 6.75% greater.

6. For the OW-BF(N) design, overall similar, but slightly worse, performance is seen across all 15 cases, with the OW-BF(N) having a 0.66% lower average cross-country speed than the BW-BF design.

From these findings, it can therefore be concluded that the optimised design outperforms

the baseline design for a wide range of cross-country tasks. It is important to note that relative performance increase is more easily achieved when comparing spans of different lengths. However, it is shown that using the optimised wing with increased span in combination with the baseline airfoil (the OW-BF(N) design) yields a similar performance to that seen in the baseline case. This highlights that the superior performance found by the optimisation module is a combination of both the optimised airfoil and the optimised wing. However, repeating this analysis whilst constraining the wing geometry to not exceed or drop below a certain span length would give further insight into the behaviour of the optimisation module and may be considered for further study.

Table 9.11: Results of 15 cross-country case studies when comparing the average cross-country performance of the BW-BF, OW-OF, OW-OF(N), BW-OF, OW-BF(N) designs

Case	Distance (Km)	A1 (%)	A2 (%)	B1 (%)	B2 (%)	Baseline (km/h)	OW-OF (km/h)	OW-OF(N) (km/h)	BW-BF (km/h)	OW-BF(N) (km/h)	Case category	OW-OF diff (%)	OW-OF(N) diff (%)	BW-BF diff (%)	OW-BF(N) diff (%)
1	300	0	57	3	40	97.3	102.4	101.7	100.1	98.8	Strong	5.24	4.52	2.88	1.53
2	500	40	30	20	10	58.0	73.4	69.6	66.0	55.8	Weak	26.55	20.00	13.79	-3.76
3	400	10	20	30	40	80.1	84.3	83.2	80.5	77.3	Strong	5.24	3.81	0.54	-3.51
4	300	100	0	0	0	39.4	58.6	53.3	49.1	36.3	Weak	48.73	35.28	24.67	-7.97
5	100	70	10	10	10	46.8	65.3	60.4	56.4	44.2	Weak	39.53	29.06	20.53	-5.58
6	200	0	50	0	50	100.2	104.0	103.7	102.1	101.2	Strong	3.79	3.52	1.92	1.01
7	500	25	25	25	25	66.7	78.2	75.8	73.5	64.6	Mixed	17.24	13.64	10.25	-3.16
8	100	50	50	0	0	56.0	74.5	69.9	65.8	67.0	Mixed	33.04	24.82	17.43	19.64
9	600	0	100	0	0	96.4	102.1	101.1	99.5	96.2	Strong	5.91	4.92	3.20	-0.20
10	400	50	0	50	0	56.7	63.2	59.8	56.4	47.5	Weak	11.46	5.43	-0.58	-16.17
11	100	0	0	0	100	104.5	106.0	105.9	104.9	106.2	Strong	1.44	1.34	0.39	1.64
12	700	10	60	20	10	78.9	87.4	85.5	82.9	78.1	Strong	10.77	8.38	5.07	-0.96
13	200	0	0	100	0	68.1	68.5	68.0	65.8	69.1	Weak	0.59	-0.15	-3.36	1.51
14	400	0	0	50	50	82.3	83.2	83.1	80.9	83.8	Mixed	1.09	0.97	-1.72	1.79
15	100	5	25	25	45	81.0	89.3	88.3	86.1	84.5	Strong	10.25	9.01	6.25	4.30

9.6 Conclusion

In this chapter, an optimisation module is designed in such a manner to enable efficient and robust sailplane cross-country performance optimisation with intuitive constraints set on the wing geometry, 2D airfoil performance characteristics, and airfoil profile as input to the process. Given the architectural design of the DSM, the optimisation module is developed such that it could be used by any gradient-free optimisation algorithm. For

this research, however, a variation of the GA is used.

The central theme of this chapter is therefore the validation of the optimisation module in terms of its capacity to optimise a given sailplane design whilst adhering to multiple constraints defined for 2D performance, wing planform design , and airfoil profile design.

To evaluate the efficacy of the optimisation module, a baseline design similar to the JS3 sailplane is specified and used to define two optimisation cases. For these two optimisation cases, two different airfoils are used; i.e. a thick and a thin airfoil and in each iteration, these foils are fixed and only the wing planforms are allowed to vary. These thick and thin airfoils are derived from the baseline airfoil (Althaus AH 80-129) using the geometric design subroutine of XFOIL. In both cases, the optimisation module is able to find a wing planform design that result in higher sailplane cross-country speeds whilst adhering to the constraints imposed on the wingspan increments and chords.

As a final verification step, the optimisation module is used to perform constrained sailplane cross-country performance optimisation where both the planform and the airfoils are allowed to vary, both with a set of upper and lower geometric constraints. After the optimisation simulation converged, the outputs are analysed in two ways. First it is verified whether the optimisation module conforms to the upper and lower bounds of the geometric wing and airfoil profile constraints. Analysis proves that in both cases all constraints are adhered to. The second analysis is performed to determine whether: (1) the optimised design indeed outperforms the baseline design, and (2) the optimised design outperforms the baseline design for more than just the baseline cross-country task. In order to achieve this, the optimised design is first converted to the TSM format so that all comparisons can be done with this model. It is shown that the optimised design outperforms the baseline design for not only the baseline cross-country task but also for a wide range of diverse cross-country tasks with an average increase of 10.97% in cross-country performance when normalising the wing span loadings to ensure accurate comparisons are made. This is especially true when weak thermalling conditions dominate the cross-country task, seeing that the optimised design has a larger area. Here it is also shown that the optimised design i.e using the optimised wing and optimised airfoil, outperforms

the designs using (1) the optimised airfoil and the baseline wing, and (2) the baseline airfoil and the optimised wing. This indicates that the optimised design attributes the performance increase to the combined alterations to the wing planform and airfoil profile.

Given these results, it could therefore be concluded that the optimisation module achieves the goal of being a single integrated system that can simulate and optimise the cross-country performance of a given sailplane design while allowing for the setting of intuitive constraints on the wing/airfoil combination.

However, two minor discrepancies need to be addressed before the optimisation module is fit for use in global optimisation of a SOTA glider. The first is the accuracy of the DSM and hence the underlying modules. The second is the application of the optimisation module to cases where each station has a unique airfoil with its own constraints setting. Both of these problems are addressed in the next section where it is shown how the respective sub-modules of the DSM are refined to have a higher degree of accuracy both in terms of standard class sailplane airfoil generation and 2D airfoil performance prediction over the desired sailplane operating conditions.

Chapter 10

JS4 sailplane cross-country performance optimisation

10.1 Introduction

In this chapter, the cross-country optimisation framework developed throughout this research is applied to optimise the performance of the JS4 sailplane model, which currently is the world's best-performing 15m standard class sailplane. Refining the deep learning modules developed thus far to facilitate this optimisation process involves focusing on two main objectives: (1) to ensure 2D airfoil performance prediction accuracy in the operating range of standard class sailplanes and (2) to ensure that only typical standard class airfoils are produced and evaluated during the optimisation process.

In Section 10.2 the main focus is on this module refinement, where first the data mining pipeline is discussed that strictly ensures standard class sailplane airfoils are mined for downstream module refinement. From here the refined sailplane performance data set is used to train the airfoil generator module, the airfoil shape conversion module and hence the airfoil shape mapping module, and finally the 2D airfoil performance module. Then in Section 10.3 the refined modules are integrated with the TSM to form the DSM which is used in end-to-end optimisation of the cross-country performance of the JS4 standard

class sailplane.

Optimisation is done in two cases, an unconstrained and a constrained optimisation case. In the former, the optimisation algorithm is allowed to alter both the planform and the airfoils at each station as opposed to the latter case where the algorithm is only allowed to adjust the airfoils at each station whilst the platform design is fixed. Also in this section, the performance of both optimisation cases is compared with the baseline JS4 design. The most optimal of the two optimised designs is then chosen for deeper subsequent analysis of the individual airfoils at each station as well as the overall cross-country performance. This is done to understand the origination of possible sources of enhanced cross-country performance.

10.2 Module refinement

This section is concerned with the refinement of the core modules required by the DSM to perform accurate and efficient cross-country performance optimisation of the JS4 standard class sailplane. First, in Section 10.2.1, a curated data set is developed containing the parsimonious variables, profile definitions, and 2D aerodynamic performance of airfoils typically used in standard class sailplanes. Then, in Section 10.2.2, using this curated data set, the airfoil shape mapping module is refined such that it is only capable of generating a diverse set of airfoil profiles typically used in standard class sailplanes. Finally Section 10.2.3 is focused on refining the 2D airfoil performance module to ensure that it is accurate to a high degree in the prediction of an airfoils' 2D aerodynamic performance, specifically in the module's lift and drag coefficient prediction capacity.

10.2.1 Data mining

To refine the respective modules, data must be mined that represents typical standard class sailplane airfoil profiles. For this task, a total of nine profiles are sourced from Jonker Sailplanes that include the following airfoils: (1) FX S 02 196, (2) LS 7, (3) LS8,

(4) BS1, (5) FX61 163, (6) Hyb1, (7) ST1, (8) OPT110, and (9) Tip23. These nine airfoil profiles serve as the basis for determining the feasible upper and lower limits of the 23 parsimonious variables of typical standard class sailplane airfoils. To determine the parsimonious variables of these nine airfoils, the framework developed and discussed in Chapter 4 is used. The resulting minimum, maximum, and median descriptive statistics for each of the 23 parsimonious variables can be seen in Table 10.1.

Table 10.1: Ranges of the shape and structural characteristics of typical standard class sailplane airfoil profiles

Parsimonious characteristic	Airfoil name									Summative statistics		
	Tip23	Opt110	ST1	Hyb1	FX61-163	BS1	LS8	LS7	FX S 02 196	Min	Max	Median
Max top thickness	1.02×10^{-1}	1.15×10^{-1}	1.10×10^{-1}	1.06×10^{-1}	1.08×10^{-1}	1.08×10^{-1}	9.81×10^{-2}	1.01×10^{-1}	1.34×10^{-1}	9.81×10^{-2}	1.34×10^{-1}	1.08×10^{-1}
Max top thickness location	3.97×10^{-1}	4.20×10^{-1}	4.20×10^{-1}	4.13×10^{-1}	3.67×10^{-1}	4.08×10^{-1}	4.07×10^{-1}	4.11×10^{-1}	3.78×10^{-1}	3.67×10^{-1}	4.20×10^{-1}	4.13×10^{-1}
Min bottom thickness	-2.88×10^{-2}	-4.44×10^{-2}	-3.88×10^{-2}	-3.55×10^{-2}	-5.67×10^{-2}	-3.30×10^{-2}	-3.69×10^{-2}	-4.74×10^{-2}	-6.15×10^{-2}	-6.15×10^{-2}	-2.88×10^{-2}	-3.88×10^{-2}
Min bottom thickness location	2.90×10^{-1}	3.93×10^{-1}	4.10×10^{-1}	4.10×10^{-1}	3.76×10^{-1}	4.10×10^{-1}	3.86×10^{-1}	4.72×10^{-1}	3.46×10^{-1}	2.90×10^{-1}	4.72×10^{-1}	3.93×10^{-1}
Max bottom thickness	3.37×10^{-3}	5.53×10^{-3}	2.06×10^{-3}	7.00×10^{-4}	7.65×10^{-3}	-6.60×10^{-4}	1.12×10^{-3}	4.99×10^{-3}	5.10×10^{-3}	-6.60×10^{-4}	7.65×10^{-3}	3.37×10^{-3}
Max bottom thickness location	9.40×10^{-1}	9.40×10^{-1}	9.53×10^{-1}	9.73×10^{-1}	9.22×10^{-1}	9.94×10^{-1}	9.55×10^{-1}	9.36×10^{-1}	9.38×10^{-1}	9.22×10^{-1}	9.94×10^{-1}	9.40×10^{-1}
Max camber	3.80×10^{-2}	3.53×10^{-2}	3.55×10^{-2}	3.55×10^{-2}	2.55×10^{-2}	3.74×10^{-2}	3.07×10^{-2}	2.73×10^{-2}	3.71×10^{-2}	2.55×10^{-2}	3.80×10^{-2}	3.55×10^{-2}
Max camber location	4.60×10^{-1}	4.37×10^{-1}	4.23×10^{-1}	4.17×10^{-1}	3.51×10^{-1}	4.07×10^{-1}	4.22×10^{-1}	3.74×10^{-1}	4.72×10^{-1}	3.51×10^{-1}	4.72×10^{-1}	4.23×10^{-1}
Max total thickness	1.29×10^{-1}	1.58×10^{-1}	1.47×10^{-1}	1.41×10^{-1}	1.64×10^{-1}	1.39×10^{-1}	1.34×10^{-1}	1.47×10^{-1}	1.94×10^{-1}	1.29×10^{-1}	1.94×10^{-1}	1.47×10^{-1}
Max total thickness location	3.42×10^{-1}	4.03×10^{-1}	4.15×10^{-1}	4.06×10^{-1}	3.42×10^{-1}	3.91×10^{-1}	3.91×10^{-1}	4.21×10^{-1}	3.37×10^{-1}	3.37×10^{-1}	4.21×10^{-1}	4.03×10^{-1}
Trailing endge angle	6.42×10^{-1}	5.82×10^{-1}	5.72×10^{-1}	6.19×10^{-1}	9.98×10^{-2}	1.25×10^{-1}	1.15×10^{-1}	9.68×10^{-2}	2.59×10^{-1}	9.68×10^{-2}	6.42×10^{-1}	5.82×10^{-1}
Leading edge angle	3.93×10^{0}	3.57×10^{0}	3.52×10^{0}	3.51×10^{0}	2.73×10^{0}	4.13×10^{0}	3.43×10^{0}	2.93×10^{0}	4.23×10^{0}	2.73×10^{0}	4.23×10^{0}	3.52×10^{0}
Cambe angle	4.58×10^{0}	4.15×10^{0}	4.10×10^{0}	4.13×10^{0}	2.83×10^{0}	4.25×10^{0}	3.55×10^{0}	3.03×10^{0}	4.49×10^{0}	2.83×10^{0}	4.58×10^{0}	4.13×10^{0}
Top curvature	7.78×10^{-1}	8.28×10^{-1}	8.98×10^{-1}	7.90×10^{-1}	1.07×10^{0}	8.28×10^{-1}	7.54×10^{-1}	6.65×10^{-1}	1.41×10^{0}	6.65×10^{-1}	1.41×10^{0}	8.28×10^{-1}
Min bottom curvature	1.73×10^{-1}	2.98×10^{-1}	3.04×10^{-1}	2.74×10^{-1}	7.33×10^{-1}	2.82×10^{-1}	3.19×10^{-1}	4.24×10^{-1}	1.00×10^{0}	1.73×10^{-1}	1.00×10^{0}	2.98×10^{-1}
Max bottom curvature	1.92×10^{0}	3.33×10^{0}	3.02×10^{0}	2.32×10^{0}	1.88×10^{0}	0.00×10^{0}	2.48×10^{0}	2.98×10^{0}	2.00×10^{0}	0.00×10^{0}	3.33×10^{0}	2.32×10^{0}
Nose radius	1.02×10^{-2}	1.68×10^{-2}	6.60×10^{-3}	5.81×10^{-3}	6.13×10^{-3}	1.65×10^{-2}	1.57×10^{-2}	5.81×10^{-3}	1.65×10^{-2}	5.81×10^{-3}	1.68×10^{-2}	1.02×10^{-2}
Total area	8.70×10^{-2}	1.05×10^{-1}	9.91×10^{-2}	9.57×10^{-2}	1.01×10^{-1}	9.60×10^{-2}	8.85×10^{-2}	1.03×10^{-1}	1.18×10^{-1}	8.70×10^{-2}	1.18×10^{-1}	9.91×10^{-2}
Perimeter	2.03×10^{0}	2.04×10^{0}	2.04×10^{0}	2.03×10^{0}	2.05×10^{0}	2.04×10^{0}	2.03×10^{0}	2.05×10^{0}	2.06×10^{0}	2.03×10^{0}	2.06×10^{0}	2.04×10^{0}
Y centroid	3.12×10^{-2}	2.89×10^{-2}	2.83×10^{-2}	2.81×10^{-2}	2.26×10^{-2}	2.90×10^{-2}	2.51×10^{-2}	2.34×10^{-2}	3.15×10^{-2}	2.26×10^{-2}	3.15×10^{-2}	2.83×10^{-2}
X centroid	4.24×10^{-1}	4.24×10^{-1}	4.37×10^{-1}	4.36×10^{-1}	4.08×10^{-1}	4.38×10^{-1}	4.32×10^{-1}	4.39×10^{-1}	4.05×10^{-1}	4.05×10^{-1}	4.39×10^{-1}	4.24×10^{-1}
Bending inertia y	9.00×10^{-5}	1.63×10^{-4}	1.33×10^{-4}	1.19×10^{-4}	1.53×10^{-4}	1.19×10^{-4}	9.70×10^{-5}	1.37×10^{-4}	2.50×10^{-4}	9.00×10^{-5}	2.50×10^{-4}	1.33×10^{-4}
Bending inertia x	4.40×10^{-3}	4.95×10^{-3}	4.89×10^{-3}	4.79×10^{-3}	4.57×10^{-3}	4.94×10^{-3}	4.27×10^{-3}	5.31×10^{-3}	5.34×10^{-3}	4.27×10^{-3}	5.34×10^{-3}	4.79×10^{-3}

With the upper and lower limits known, the data mining pipeline of Chapter 5 can be executed however, additional filters are added to the validation step to ensure that only the 2D airfoil performance and airfoil profiles are logged of those that fall within the upper and lower limits as defined in Table 10.1. The pipeline is executed to mine 2D

performance, parsimonious, and profile data for a total of 18 614 airfoils. This data set is referred to as the refined sailplane performance data set. In Figure 10.1, 20 randomly sampled airfoils from the refined sailplane performance data set are depicted. Here the diversity in airfoil profiles can be seen, specifically in the variation of (1) the position and magnitude of maximum camber, (2) the position and magnitude of the thickness distribution on the top and bottom surfaces, and (3) the reflex and curvature present at the trailing edge.

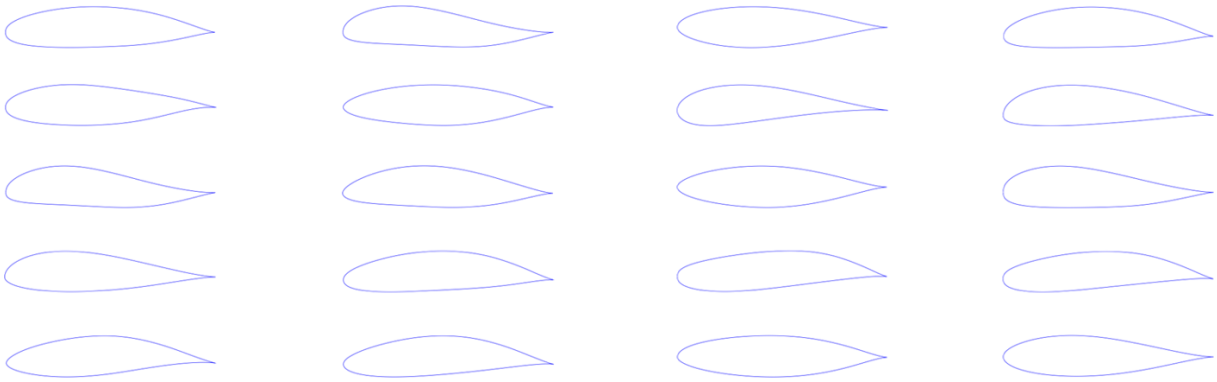


Figure 10.1: Randomly sampled airfoils form the refined sailplane performance data set

As will be shown in the subsequent sections, this refined sailplane performance data set will form the basis of the refinement of the airfoil shape mapping module, and its sub-modules, as well as the 2D airfoil performance module refinement.

10.2.2 Airfoil shape mapping module refinement

This section is concerned with the refinement of the airfoil shape mapping module to ensure that it is able to generate typical standard class sailplane airfoils during the optimisation of the JS4 sailplane. Recall from Chapter 7 that the airfoil shape mapping module is comprised of two sub-modules, the airfoil generator module and the airfoil shape conversion module. Both of these sub-modules, therefore, need to be optimised individually using the refined sailplane performance data set before they are integrated to form the newly refined airfoil shape mapping module. The respective modules are optimised following the same procedures outlined in Sections 7.3 and 7.6, ensuring that

the optimal model parameters are selected given the data and computational constraints. These optimised models are then integrated as outlined in Section 7.7 to form the refined airfoil shape mapping module.

In order to showcase the standard class airfoil generative capacity of the refined airfoil shape mapping module, 20 airfoils are generated with the module by means of randomly perturbing the input latent space of the BGMM for various airfoil cluster selections. These 20 airfoil profiles can be seen in Figure 10.2. When comparing this figure with Figure 7.19, the propensity of the refined airfoil shape mapping module to generate workable standard class airfoils is clear, irrespective of the random latent variable and cluster selection, which previously was not the case.

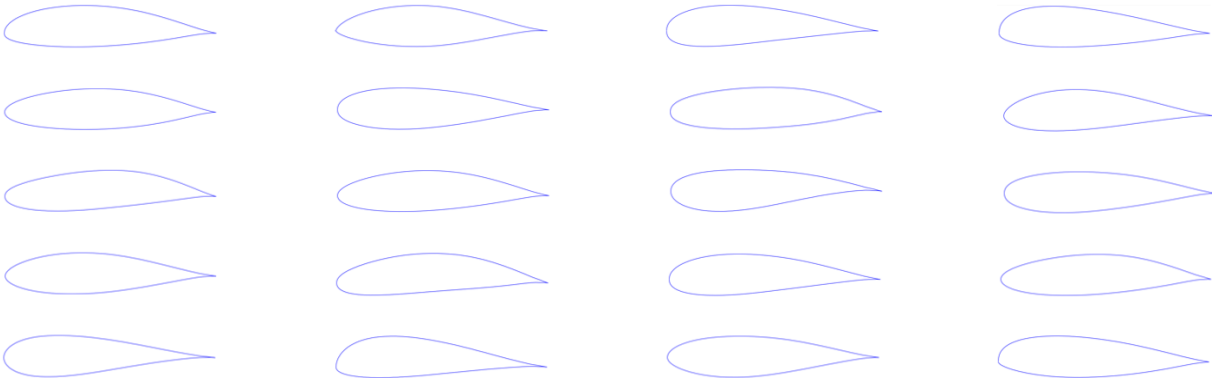


Figure 10.2: Randomly generated airfoils with the refined airfoil shape mapping module

For validation of the refined airfoil shape mapping module's generative capacity, the ST1 airfoil profile is fitted with the procedure outlined in Section 7.7.2. The module is able to fit the ST1 airfoil profile with an average fitting error between the y-coordinates of the actual and generated airfoil of 1.6×10^{-6}. The profiles of the actual and generated airfoil can be seen in Figure 10.3. From this figure, it is observed that there is near-perfect fitting for the top surface and the majority of the bottom surface. However, because the parameterisation framework derived in Section 4.3 allows for limited control over the trailing edge, from a curvature, reflex, thickness, and trailing edge gap perspective, the refined airfoil shape mapping module is not able to accurately fit the ST1 airfoil profile near the trailing edge.

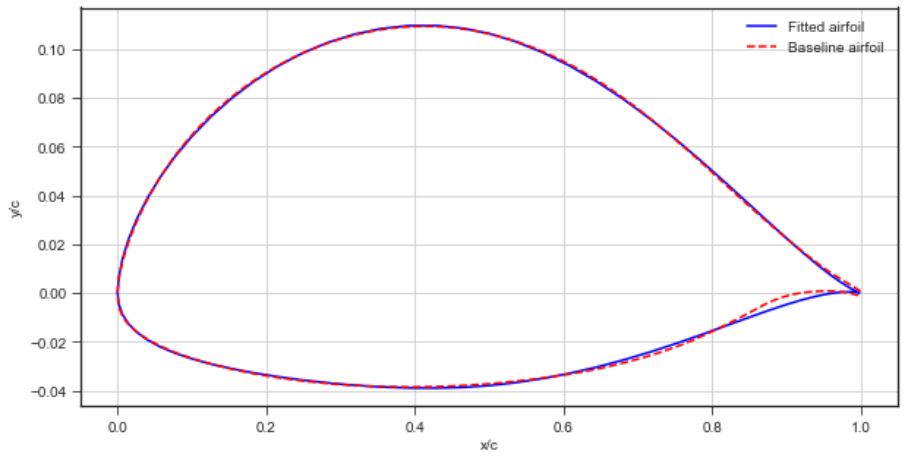
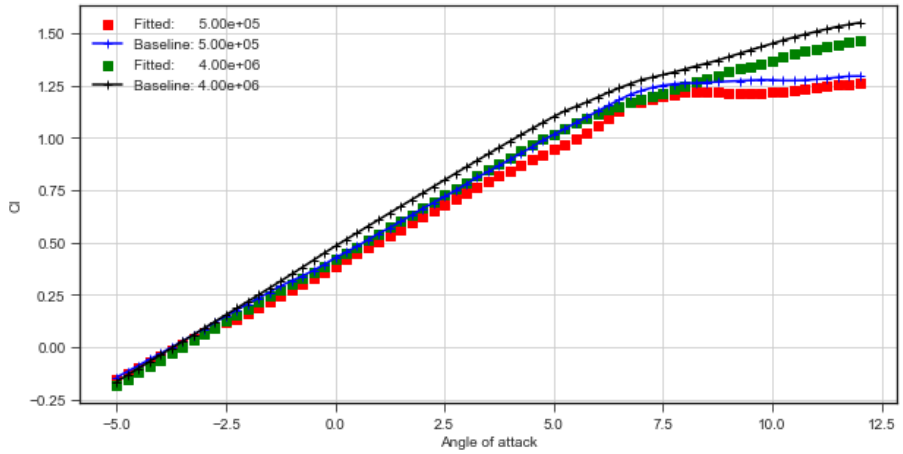


Figure 10.3: Actual and approximated ST1 airfoil profile fitted with the refined airfoil shape mapping module

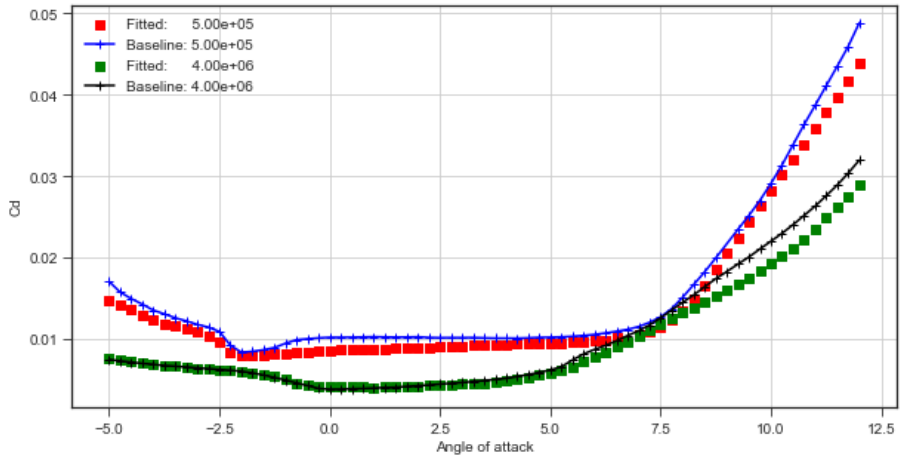
With the visible differences between the fitted and actual profiles of the ST1 airfoil, especially near the trailing edge, a performance analysis is done to understand the effects these differences have on the aerodynamic performance of the respective profiles. Here, the aerodynamic performance of the actual and fitted ST1 airfoil profiles are analysed at two Reynolds numbers i.e. 0.5×10^6 and 4×10^6, which are the upper and lower limits typically present during sailplane cross-country flight near the root and tip. The analysis is done with XFOIL with the Ncrit value set to 9, the Mach number set to 0.3 and the 2D lift and drag performance simulated between -5° and 12°. The results of this analysis can be seen in Figure 10.4. As can be seen from this figure, the differences between the lift and drag performance of the fitted and actual ST1 airfoils are material with the average drag coefficient MAE for the two cases equating to 0.001 ($R^2 = 0.97$), and an average lift coefficient MAE of 0.053 ($R^2 = 0.99$). These differences are a direct result of the absence of low-level control of the trailing edge.

However, as later will be shown, the absence of this low-level trailing edge control did not hinder the optimisation of the JS4 sailplane, and the optimisation module is able to optimise the sailplane performance in two separate instances. This case study however does highlight the need for more refined control over the airfoil trailing edge and hence it is recommended that future studies focus on further extending the refined airfoil shape

mapping module in this regard.



(a) Lift performance of the actual and fitted ST1 airfoil profiles for Reynolds numbers of 0.5×10^6 and 4×10^6 respectively



(b) Drag performance of the actual and fitted ST1 airfoil profiles for Reynolds numbers of 0.5×10^6 and 4×10^6 respectively

Figure 10.4: Lift and drag performance of the fitted and actual ST1 airfoil profiles simulated for the upper and lower extreme Reynolds numbers present during typical cross-country flight

10.2.3 2D airfoil performance deep learning module refinement

The final module that needs to be refined to enable full optimisation of the JS4 sailplane is the 2D airfoil performance deep learning module. Once again this is done using the refined sailplane performance data set. To arrive at the deep learning model with the best parameters that minimises the validation loss, the procedures outlined in Section 6.4 is followed. Training is executed for 500 epochs with L1 and L2 regularisation set to 1×10^{-6}. Early stopping is invoked at 467 epochs after no training improvement, based on the validation loss, is realised for the past 25 epochs. The best-performing model has 10 layers, 512 neurons per layer with 2 664 968 parameters of which 2 654 728 are trainable and 10 240 are non-trainable. This model resulted in a training loss of 6.14×10^{-4} and a validation loss of 3.42×10^{-4} which is a significant improvement over the more general performance module developed in Section 6.4.2.

Analysing the performance of the refined 2D airfoil performance deep learning module on out-of-sample data involves utilising the module to predict the performance of the airfoils in the hold-out data set and calculating the respective performance metrics. The results from this analysis can be seen in Table 10.2. From the mean and standard deviation predictions seen in this table, it is clear that the spread and variability of the data are captured accurately by the module. Furthermore, the average MAPE for all eight response variables equates to 1.960, meaning on average the module is 98% accurate. The overall adjusted R^2 calculated on the hold-out data set equates to 0.995, further indicating the refined modules' capacity to accurately model the variation in the eight 2D performance response variables. Analysing the lift and drag MAPE, respective values of 2.159 and 3.391 are observed.

These metrics are aligned with similar work, specifically, the work of Peter Sharp (Sharpe, 2023), where a drag error of 3.9% was observed for a 5-layer 512 neuron, physics-informed neural network trained on data mined from XFOIL using MIT's Supercloud – a high-performance computing cluster operated by the MIT Lincoln Laboratory Supercomputing Center (LLSC). The model of Peter Sharpe used millions of airfoils compared to the limited set of our work and also had the availability of significantly larger compute instances.

The improved drag performance observed in our model compared to Peter Sharp's NeuralFoil SOTA model demonstrates the effectiveness of our approach in parameterisation, data mining, data sampling, and model training.

Furthermore, to improve the inference times of the model, the refined module is quantised. This results in a model with a smaller memory footprint, which maintains the overall module accuracy. The quantised module is capable of running batch inference for a single airfoil for a total of 120 airfoil operating conditions in under 100 milliseconds – which will further improve the cross-country performance optimisation efficacy.

The overall improved accuracy in 2D aerodynamic performance prediction for workable standard class sailplane airfoils, the high R^2, and the low batch inference times result in a refined module that can now be utilised for JS4 cross-country performance optimisation.

Table 10.2: Analysis of the refined 2D airfoil performance deep learning module on the hold-out dataset

Response variable	Prediction		Actual		Model performance metrics			
	Mean	Std.	Mean	Std.	R^2	MSE	MAE	MAPE
Maximum C_l	1.394	0.229	1.394	0.229	0.995	2.5×10^{-4}	1.2×10^{-2}	0.655
C_l	0.046	0.889	0.049	0.889	0.999	6.8×10^{-4}	1.9×10^{-2}	2.159
C_d	0.016	0.013	0.015	0.013	0.989	2.0×10^{-6}	7.9×10^{-4}	3.391
C_m	-0.046	0.056	-0.046	0.056	0.997	9.2×10^{-6}	2.0×10^{-3}	3.334
Top transition location	0.526	0.336	0.527	0.337	0.998	1.2×10^{-4}	7.0×10^{-3}	1.261
Bottom transition location	0.433	0.391	0.433	0.392	0.999	1.5×10^{-4}	8.0×10^{-3}	2.616
Zero-lift AoA	-2.28	1.433	-2.270	1.431	0.998	3.0×10^{-3}	3.7×10^{-2}	1.298
Lift-curve slope	4.166	0.609	4.166	0.614	0.988	4.4×10^{-3}	5.1×10^{-2}	0.969

10.3 JS4 sailplane optimisation

Cross-country performance optimisation of the JS4 sailplane is enabled by integration of the refined modules into the DSM and ultimately into the optimisation module to form the final refined optimisation module. This module therefore forms the basis for the optimisation done in the remainder of this chapter.

Two cases are considered in the optimisation of the JS4 cross-country performance: (1) an

unconstrained platform case, and (2) a constrained planform case. For the unconstrained planform case, each station chord length is allowed to vary between 0.1 and 0.9 meters and the optimisation algorithm adjusts both the chord lengths and the airfoil profiles of each station to optimise the cross-country performance. To ensure the genetic algorithm learns to generate realistic wing planforms Equation 9.5 is used. Also, seeing that the JS4 standard class sailplane has a wing span of 15 meters, the wingspan increments are sinusoidally distributed along the six stations meaning, the wingspan increments get shorter as you near the wing tip.

For the constrained planform case, the planform is fixed to be exactly the same as the JS4 design. This means that no additional penalty is required to teach the optimisation algorithm to generate practical planform designs and all the focus is on finding the optimal airfoil profile combinations.

Seeing that full-scale sailplane cross-country optimisation is significantly more complex than the case study presented in Section 9.4, a further step is taken to ensure the optimisation algorithm is in the best possible position for overall performance optimisation. In this step, a set of 2 500 initial candidate solutions are generated upfront for downstream use in the optimisation algorithm. These 2 500 candidate solutions are generated such that the airfoil profiles in each solution fall within the desirable shape characteristic constraints. This means less time is spent by the optimisation algorithm to generate and penalise airfoils with undesirable thickness, camber, curvature, etc. properties, and more time is spent on finding the optimal shapes and airfoil combinations along the wing span. These initial candidate solutions are iteratively generated by randomly sampling the latent space of the airfoil shape mapping module and then only including the airfoils that have the desired shape characteristics. The upper and lower limits of the chosen airfoil shape characteristics can be seen in Table 10.3.

With the optimisation algorithm having the ability to permute the latent codes of the initial airfoil candidate solution population, airfoils with characteristics outside those defined in Table 10.3 may result. These cases are penalised according to Equation 9.1. However, because the initial population all have acceptable characteristics, the algorithm

is less likely to produce shapes outside the desired bounds, especially as more optimisation iterations are executed.

Table 10.3: Upper and lower parsimonious variable constraints for JS4 cross-country performance optimisation

Parsimonious feature	Lower bound	Upper bound
Maximum thickness magnitude	0.100	0.220
Maximum camber magnitude	0.000	0.100
Maximum top magnitude	0.080	0.180
Maximum top location	0.150	0.650
Maximum bottom magnitude	-0.065	-0.030

In both optimisation cases, the GA is used as optimisation algorithm and the cross-country flight task is as defined in Section D.4 – with the baseline weight set to 485 kg. The GA is executed for 25 generations, and in each generation, the airfoil profile designs (and in the case of the unconstrained planform, the chord lengths) of each station are adjusted by the GA in the direction that most likely would result in a reduction of the objective function. In this case, the objective function is similar to Equation 9.6 with the respective error corrections added.

10.3.1 Optimisation results

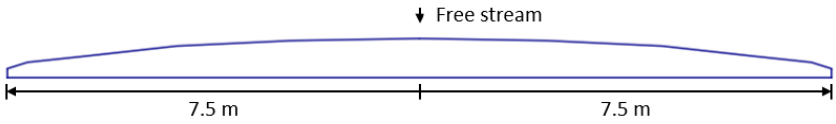
In both optimisation cases, the algorithm ran for all 25 generations with no early stopping invoked – meaning improvement in cross-country performance was realised throughout all optimisation iterations. The optimal airfoils at each station as well as the corresponding planform can be seen in Figures 10.5 and 10.6.

As can be seen from these two figures, the area for the unconstrained planform case is smaller than for the constrained planform case. Specifically, the area of the unconstrained optimisation planform is 8.58 m^2 and the aspect ratio is 26.4. For the constrained optimisation case the area is 10.03 m^2 and the aspect ratio is 22.7.

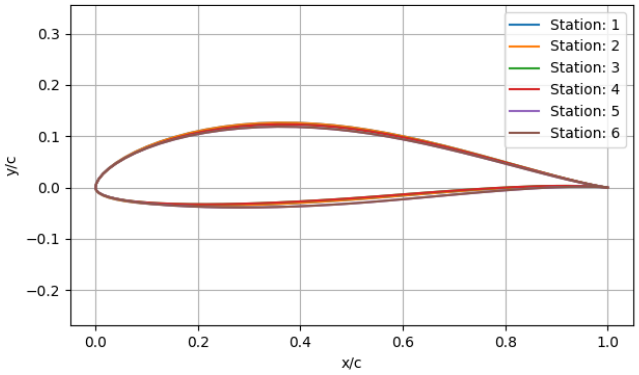
Furthermore, from Figure 10.5, the following remarks can be made regarding the optimised

results of the unconstrained case:

- The total airfoil thickness decreases as we near the wing tip. This is expected behaviour given the Reynolds number distribution from the root to the tip. The GA therefore accurately learned some general design patterns for increased performance.
- All the airfoils have shape characteristics that fall within the desired upper and lower boundaries, meaning all upfront defined constraints are adhered to.
- The optimisation algorithm set the airfoils at stations 1 and 2 equal, stations 3 and 4 equal, and stations 5 and 6 equal. This is likely due to the GA getting trapped in a local optimum. This can be circumvented by adding more diversity into the parents' mating pool parameter as well as increasing the number of generations. However, overall cross-country performance increase was still achieved as will be seen later in this section.
- The optimised unconstrained wing design has a small area that will not adhere to the minimum stall requirements for standard class sailplanes. In future work, a penalty should be added to the loss function to ensure that the GA does not allow such solutions to be exploited in the optimisation process.



(a) Planform for the unconstrained optimisation case

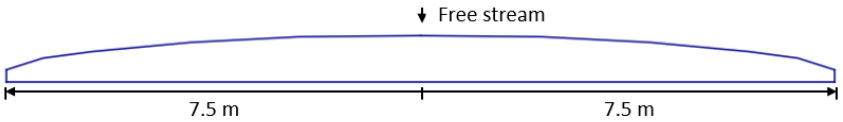


(b) Airfoils for the unconstrained optimisation case

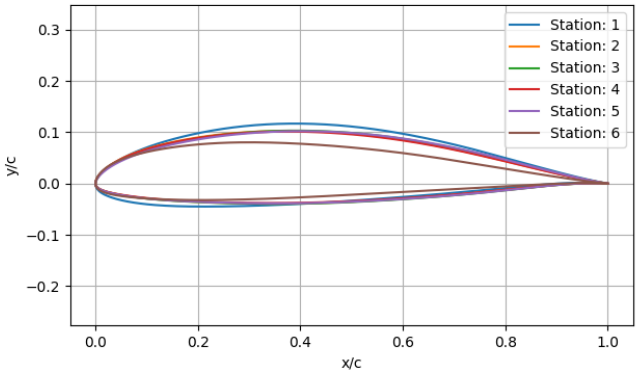
Figure 10.5: Final wing and airfoil design resulting from the unconstrained planform optimisation case

From Figure 10.6, the following remarks can be made regarding the optimised results of the constrained case:

- Once again here, the GA learned to decrease the total airfoil thickness distribution from the root to the tip to compensate for the change in Reynolds number.
- The airfoils at stations 2, 3, 4, and 5 share a very similar design with only minor profile differences to compensate for the different Reynolds and Mach numbers present over the flight envelope. This means that the GA found that the general design used here has the best performance for the Mach and Reynolds numbers present at these stations, whilst adhering to the shape constraints.
- All the optimised airfoils adhere to the upfront defined shape constraints – supplied as input to the optimisation pipeline.



(a) Planform for the constrained optimisation case



(b) Airfoils for the constrained optimisation case

Figure 10.6: Final wing and airfoil design resulting from the constrained planform optimisation case

To analyse and compare the overall performance of the respective optimised designs with the baseline design, the JSCode is used. Specifically, the JSCode is used to simulate the sink speed for the baseline design, the constrained design, and the unconstrained design at a weight of 485 kg for free stream speeds ranging from 90 km/h to 220 km/h. The percentage improvement as well as the respective sink speed performance can be seen in Figures 10.7 and 10.8

When analysing Figure 10.7, it is clear that the unconstrained design has similar or slightly better performance than the baseline design for the climbing phase i.e. the free stream speeds ranging from 90 km/h to 120 km/h, with an average increase in performance of 2.4% over this range. From 120 km/h to 180 km/h the baseline design outperforms the unconstrained design with an average relative decrease in performance of -2.8%. Then in the high speed or gliding phase i.e. the free stream speeds ranging from 180 km/h to 220 km/h, the unconstrained design outperforms the baseline design with an average performance increase of 3.7% in this range. Overall, the unconstrained design has an

average performance increase of 0.36% when compared to the baseline design.

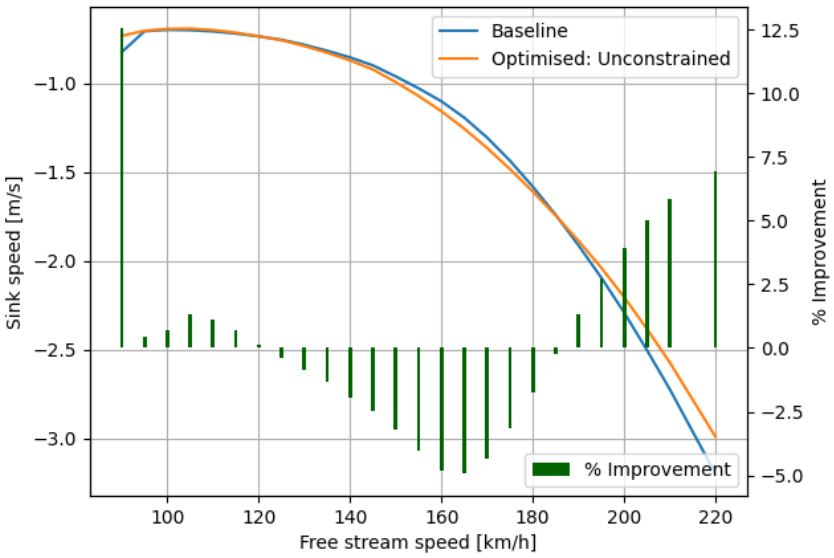


Figure 10.7: Sink speed performance and performance improvement of the unconstrained planform design over the baseline JS4 design for a range of free-stream speeds. Simulations are executed at a weight of 485 kg.

When comparing the sink speed performance of the baseline design with the constrained design, as seen in Figure 10.8, the same remarks can be made on the performance increase in the gliding phase. Here, however, the performance increase of the constrained design over the baseline design is more prominent compared to the unconstrained case, with an average increase of 5.5% in the range of 180 km/h to 220 km/h. In contrast with the unconstrained design, the baseline design outperforms the constrained design in the range of 90 km/h to 180 km/h, however, this is very slight with an average outperformance of 0.9%. This means the GA was able to find a design that either had very similar performance to the baseline design or had superior performance. Overall, the constrained design outperformed the baseline design with an average performance increase of 0.8%.

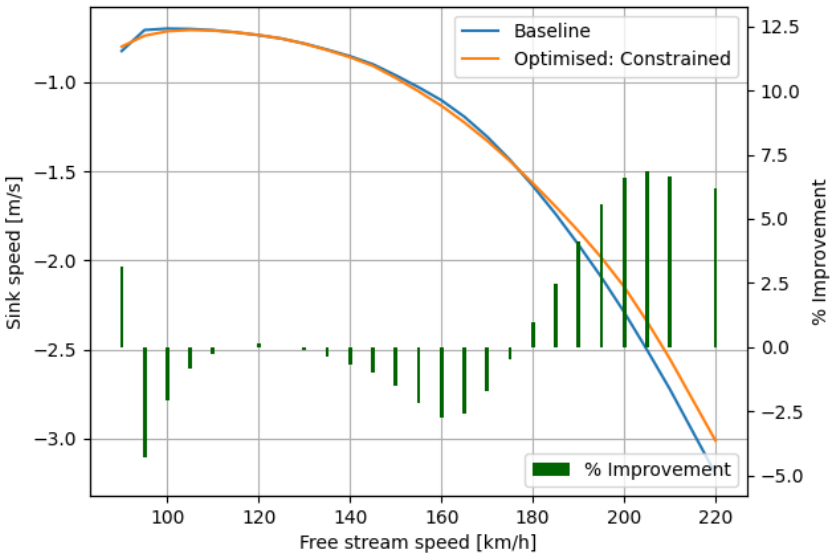


Figure 10.8: Sink speed performance and performance improvement of the constrained planform design over the baseline JS4 design for a range of free-stream speeds. Simulations are executed at a weight of 485 kg.

Considering the improved performance of both the unconstrained and constrained designs when compared to the baseline design, and noting that the constrained design outperforms the unconstrained design, the constrained design is selected for further analysis. This selected optimal design will be subject to an in-depth exploration in Section 10.3.2 to gain a deeper understanding of the sources of cross-country performance improvement.

10.3.2 Optimal design analyses

To understand the sources of performance improvement of the constrained i.e. optimal design over the baseline design, two analyses are conducted. The first is an analysis of the individual airfoils. Here, each airfoil is analysed at the station's respective maximum and minimum Reynolds number and the performance is compared with the baseline design over the same operating conditions. This is done to understand where the optimised airfoils have improved performance over the baseline design. The second is an analysis of

the cross-country performance at different weights for different cross-country tasks. Here, both the optimal and baseline designs are used to simulate the average cross-country speeds over a range of flight tasks for different sailplane weights to understand the overall optimality of the optimised design.

For the first analysis XFOIL is used to simulate the performance of both the baseline airfoil and the optimised airfoil of each station at their respective upper and lower Reynolds numbers. For these simulations the Ncrit value is set to nine and a Mach number of 0 is used. Also, the simulations are executed for the angles of attack ranging from -7.50° to 10.00°. The main results that are used for this discussion can be seen in Figure 10.9, which compares the drag count difference of the baseline and the optimised airfoils for each respective operating condition. It is important to note that in this analysis, the baseline airfoil is used as the denominator and hence a value below zero means improved drag performance. Further supplementary analysis to support the findings of this section can be viewed in Appendix G and is not included here for the sake of brevity.

From Figure 10.9, it can be seen that for station one, for the low angles of attack i.e. -7.50° to -1.75° and the high angles of attack i.e. 5.00° to 10.00° the optimised design outperforms the baseline design. Overall for station one, the optimised design has an average decrease of 8.22 drag counts for the operating conditions analysed, specifically with an average drag count reduction of 10.58 for the Reynolds number range of 1 500 000 and an average drag count reduction of 5.85 for the Reynolds number range of 3 500 000. Furthermore, the optimised design's root airfoil and the baseline design's root airfoil have very similar lift production for the same operating conditions (see Figure G.1a) with the optimised design having a slight average increase of 0.01 over the respective operating conditions. This means the optimised design, on average, has a higher lift-to-drag ratio than the baseline design – which equates to an average increase of 5.57 in the lift-to-drag ratio count.

From the result analysis of station two as depicted in Figure 10.9, it is clear that the optimised design has slightly lower drag performance in the angle of attack range between -7.50° and -2.00°. Here it is also noted that, for the Reynolds number case of 1 500

000, the optimised airfoil either has similar performance to the baseline airfoil or has a significantly lower drag performance. However, in both Reynolds number instances, the optimised airfoil has a significantly lower drag coefficient in the angle of attack range between 7.00° to 10.00°. Here, the optimised design has, on average, a 7.71 drag count decrease over the operating conditions simulated. In this case, the lift production of the optimised design is lower than the baseline design (see Figure G.1b) meaning there is an average decrease of 6.12 in the C_l/C_d over the cases simulated. However, the optimised design's enhanced drag performance for the angle of attack range between 7.00° to 10.00° means the optimised design has a significant increase in the lift-to-drag performance over this region with an average increase in of 21.40 lift-to-drag counts. In the cases of station 1 and station 2, the optimised design's airfoils are seen to have smoother stall properties and lower drag performance at higher angles of attack meaning that for both of these cases, the optimised design has more favorable lift-to-drag ratio performance at these conditions – see Figures G.2a and G.2b.

As seen in Figure G.3, the airfoils present at stations 2-4 share notable similarities with one another and also with the baseline airfoils present at these respective stations. Similar remarks on the performance of optimised airfoils 3 and 4 can therefore be made than were previously made for optimised airfoil 2. Specifically, optimised airfoils 3 and 4 have, in general, lower drag performance for the same operating conditions with a significant reduction in drag for the angles of attack range between 7.00° to 10.00°. Once again here the smoother stall of airfoils 3 and 4 and the reduction in drag at higher angles of attack means that the optimised airfoils achieve superior lift-to-drag performance in the angle of attack ranges of 7.00° to 10.00° – see Figure G.2.

Although optimised airfoil 5 shares a similar profile to optimised airfoils 2-4, there is a significant difference in the respective drag performance when compared to the baseline airfoil. This is because here, the baseline design makes use of an airfoil that differs significantly from the optimised airfoil, as seen in Figure G.3e. In terms of drag performance, optimised airfoil 5 outperforms the baseline airfoil present at station 5 for the majority of angles of attack simulated. However, from Figure G.1e, it is clear that optimised airfoil 5 has significantly lower lift production than baseline airfoil 5 for all angles of attack

simulated. This results in optimised airfoil 5 having significantly lower lift-to-drag performance than baseline airfoil 5, specifically in the angle of attack ranges between -5 to 2.5 – as shown in Figure G.2e.

Finally, when comparing the performance of optimised airfoil 6 with baseline airfoil 6, the optimised airfoil either has similar or significantly lower drag performance over the operating conditions simulated. However, as seen in Figures G.1f and G.2f, the lower lift production of the optimised airfoil result in a significant relative decrease in the lift-to-drag ratio over the analysed set of operating conditions. It is also interesting to note that for the tip case i.e. station 6, the optimised airfoil once again has favourable stall characteristics with smooth lift-to-drag performance at high angles of attack.

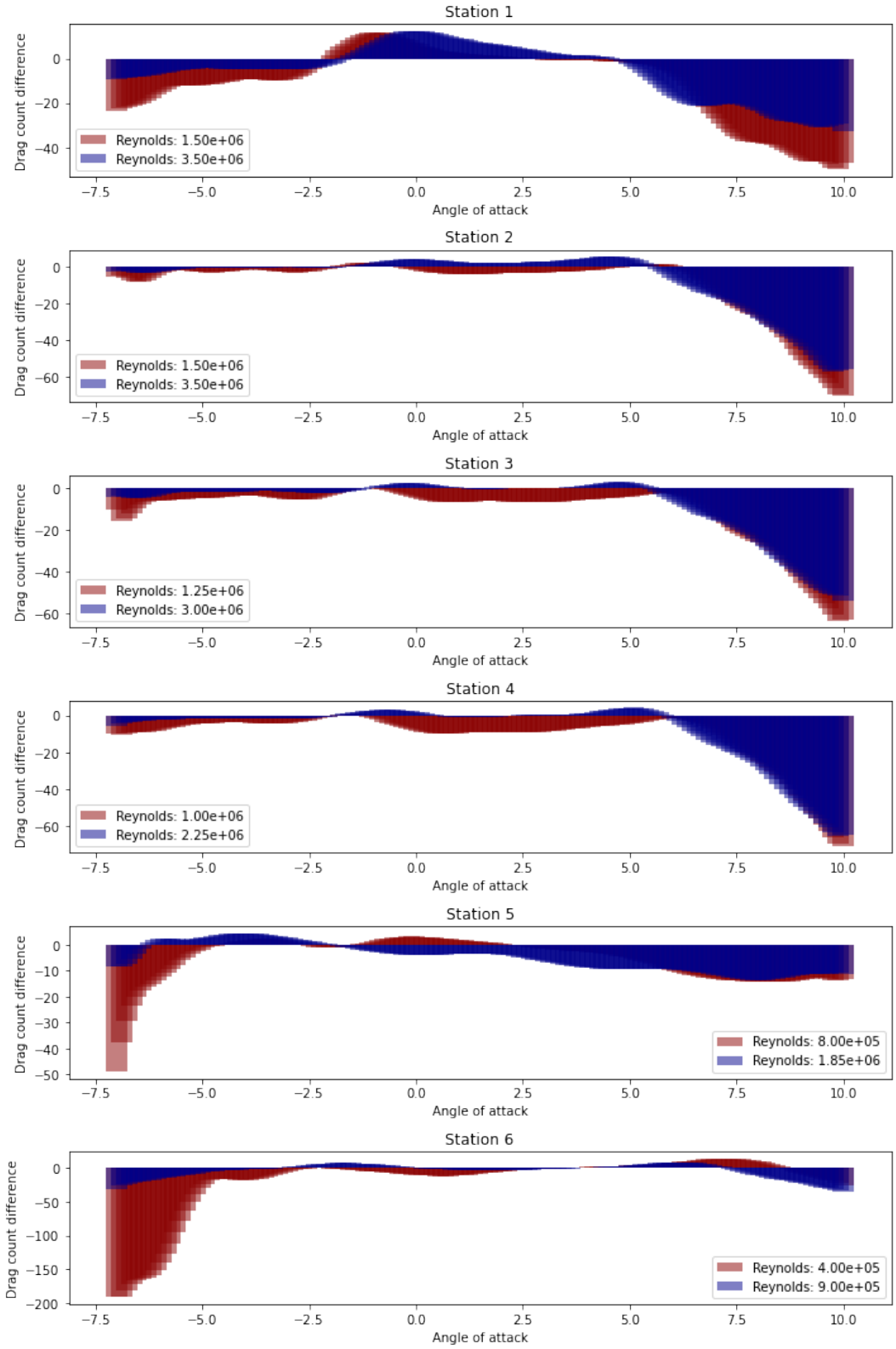


Figure 10.9: Difference in drag performance, measured in drag counts, of the optimised and baseline airfoils at each station and their respective upper and lower Reynolds number operating ranges

The next and final analysis compares the average cross-country speeds of the baseline design and the optimised design for different sailplane weights and cross-country flight conditions. This analysis is done with the JSCode and a total of four cross-country flight conditions are investigated. In each of the four flight conditions, the thermal gradient is set to 0.025 $1/s$ with a varying thermal vertical speed. The thermal vertical speeds, defined at a thermal radius of 60 m, for the four respective cases, are defined as follows:

1. Case 1: 3 m/s,
2. Case 2: 4 m/s,
3. Case 3: 5 m/s, and
4. Case 4: 6 m/s.

With the four flight cases defined, the average cross-country speed of the baseline and the optimised designs are determined with the JSCode at sailplane weights of 450 kg, 485 kg, and 525 kg, and the maximum speed of each respective simulation is plotted, as seen in Figure 10.10. From this figure, it is clear that the optimised design outperforms the baseline design for three of the four cases. This relative performance increase is especially significant for cases three and four where stronger thermals are present indicating that the optimised design has favourable performance in strong flight conditions. For cases three and four the relative performance increase is 1.67% and 2.32% respectively. Similar findings are seen for case two where the relative performance increase is 0.25% with the optimised design slightly outperforming the baseline design, however here the relative increase is not as significant as for cases three and four. For case one, where weaker thermalling conditions are present, the baseline design outperforms the optimised design by 0.50%, indicating that the optimised design's performance can further be improved in this region. Overall the optimised design has an average relative performance increase of 0.94% over the four cases investigated.

These findings, alongside the findings earlier in this section and those of Section 10.3.1 therefore highlight that the optimised design indeed has enhanced cross-country performance when compared to the JS4 baseline design and it is concluded that the deep learning

framework developed in this research is capable of global cross-country performance optimisation of a SOTA standard class sailplane design – *if* the modules are configured properly.

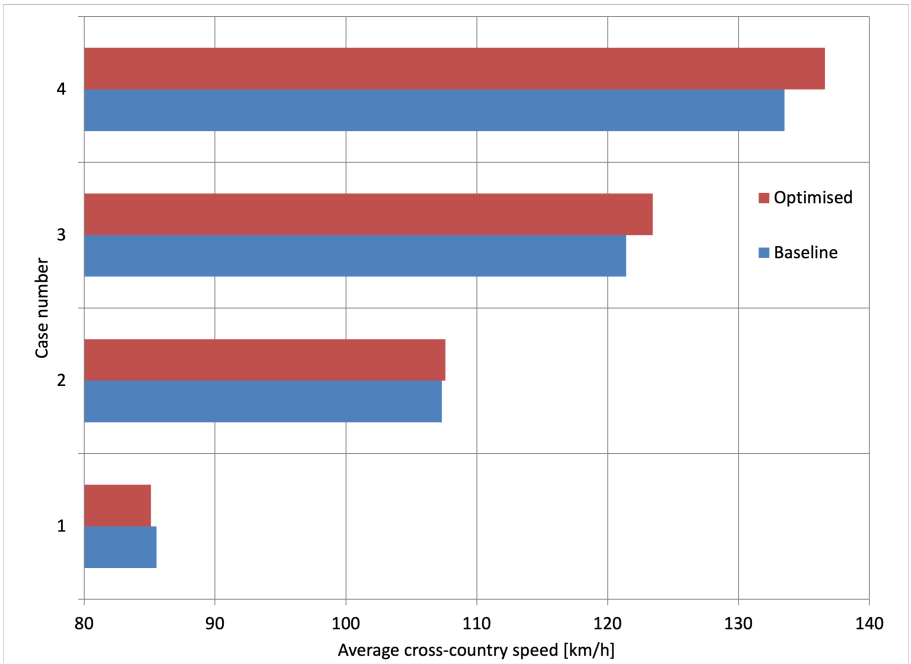


Figure 10.10: Comparison of the cross-country performance of the JS4 baseline design and the optimised design for four different flight conditions

10.4 Conclusion

In this chapter, the deep learning sailplane optimisation framework developed throughout this research is refined for use in an end-to-end sailplane optimisation pipeline with the ultimate goal of seeking cross-country performance improvements over the current JS4 sailplane design.

In this endeavour, all the modules developed throughout this research needed to be refined, including the airfoil shape mapping module and its respective sub-modules, as well as the 2D airfoil performance deep learning module. The airfoil shape mapping module is refined to ensure that: (1) only workable standard class sailplane airfoils can be generated with the module, and (2) that the parsimonious airfoil representation and the raw x and y

airfoil coordinate representation map to exactly the same airfoil profile. It is shown how this module can generate airfoils from the desired profile distribution as well as approximate the ST1 airfoil, which is a state-of-the-art standard class sailplane airfoil. In this fitting, it is however highlighted that the refined module lacks explicit control over the trailing edge in terms of the trailing edge gap, curvature, boat angle, etc. Hence, it is recommended that the parsimonious variable framework be extended in future work to allow for more explicit control here. This is important because if the optimisation module can generate SOTA airfoil designs, greater cross-country performance optimisations will be achieved with the proposed deep learning optimisation framework than those reported in this research.

Also in this chapter, the 2D airfoil performance deep learning module is refined to ensure that it can accurately model the aerodynamic performance of the sailplane airfoils generated by the airfoil shape mapping module. This is to ensure that the downstream optimised designs, resulting from this module's use, translate to actual performance enhancements when analysed and validated with the JSCode and TSM. It is shown how this module is able to achieve an average MAPE of 1.960 across all eight response variables in the testing data set – meaning on average the module is 98% accurate in its 2D aerodynamic performance predictions.

Finally, these refined modules are integrated into the optimisation module- which in turn is used in two optimisation cases – an unconstrained and a constrained optimisation case. In both cases, the objective is to find cross-country performance improvements over the JS4 baseline design. For the unconstrained case, the optimisation module is allowed to adjust the airfoils as well as the wing planform, and for the constrained case, the planform is fixed to be the same as the JS4 baseline design, and the optimisation module is only allowed to adjust the airfoils at each wing station.

In both cases, the optimisation module is able to find designs with superior performance to that of the JS4 baseline design. For the unconstrained design the average sink speed performance increase over the free stream speed range of 100 km/h to 220 km/h is 0.36%. For the constrained case this relative performance increase over the same free stream

speed range is 0.80%.

Seeing that the best-performing design is the one resulting from the constrained optimisation case, this design is selected for further analyses to understand both the individual airfoil performance per station, as well as the overall cross-country performance for different flight tasks. From the airfoil analysis it is noted that, in general, the airfoils of the optimised design have lower drag production compared to the baseline design, especially between the angles of attack of -7.5° to 2.5°, and 5° to 10°. For the cross-country performance, four cases are investigated each with the same thermal gradient, but varying thermal strengths at a thermal radius of 60 meters. It is observed that, on average, the optimised design outperformed the baseline design with a relative cross-country performance increase of 0.94%.

In conclusion, the results presented in this chapter, collectively emphasise the potential of the deep learning framework established in this research to enable global cross-country performance optimisation of a SOTA standard class sailplane design with explicit constraints set on the profile and wing planform of the final design.

Chapter 11

Conclusion and recommendations

11.1 Introduction

As introduced in Section 1.4, this research aims to develop a general framework capable of allowing for multi-objective sailplane cross-country performance optimisation for diverse flight tasks and weather conditions. In this, the TSM as displayed in Figure 1.2 is modified to ultimately form the DSM by leveraging deep learning to replace the entire airfoil performance analysis module with a surrogate model. This DSM is developed with the aim of allowing for intuitive constraint setting on the wing planform and individual airfoil profiles along the span of the wing as input to the optimisation process. In the process, a new parsimonious airfoil generation technique was developed that is capable of generating airfoils in both their parsimonious variable representations as well as their x and y coordinate representations.

This chapter provides a summary of the level of success achieved in this endeavour. In Section 11.2 a summary of the contributions made by this research is given. In Section 11.3 a consolidated view of the critical findings of this research is presented. Then, in Section 11.4, the main recommendations of this research are discussed, specifically commenting on the extension of the DSM from the perspective of the research scope and framework limitations. Finally, in Section 11.5 a brief conclusion is made regarding the research and

its outcomes.

11.2 Summary of contribution

The main contribution of this research is a framework that utilises deep learning models to simplify and improve the cross-country design and optimisation process of a sailplane over a wide range of flight and weather conditions, in a manner that seamlessly incorporates expert input into the process. The novelty of the framework is proven in which it is applied to optimise the cross-country performance of a SOTA standard class sailplane.

Additional contributions include:

1. A new software system for traditional cross-country performance design and optimisation that incorporates a parametrisation model, a 2D performance analysis model, a sailplane performance analysis model, and a cross-country model (Section 3.5).

2. The design of a framework for extracting 17 parsimonious shape and six parsimonious structural variables from an airfoil's profile definition in terms of raw x and y coordinates (Section 4.4).

3. The development of a data mining pipeline that can facilitate the extraction of airfoil profile and 2D airfoil performance data in terms of different data modalities including x and y coordinates, SPM coefficients, and 23 parsimonious shape and structural variables (Section 5.3).

4. The design of a deep learning surrogate modelling framework that, given enough high-quality data, can be used to replace CFD and panel code systems for rapid calculation of 2D performance characteristics over the entire flight envelope (Section 6.2).

5. The development of a 2D airfoil performance analysis deep learning model capable of accurately predicting the conventional and non-conventional performance characteristics in terms of the specification of 23 parsimonious variables (Section 6.4).

6. The development of a generative model capable of generating practical airfoil shapes in terms of a set of 23 parsimonious variables (Section 7.3).

7. The development of a constructive airfoil generation model that can predict the 2D profile of an airfoil in terms of x and y coordinates given as input 23 parsimonious variables (Section 7.6).

8. The development of a DSM that can facilitate fast and accurate calculation of a sailplane's cross-country performance for a given flight task and weather model (Sections 8.3 and 8.5).

9. The development and demonstration of an optimisation module that utilises a DSM to enable intuitive constraint setting on airfoil profile and planform designs as input to the sailplane cross-country performance optimisation process (Section 9.2).

10. The application of the developed optimisation module to optimise the cross-country performance of a SOTA standard class sailplane i.e. the JS4 sailplane (Section 10.3).

11.3 Key findings

In this research, numerous sub-modules are developed in light of the main objective stated in Section 1.4. These sub-modules are individually developed and their respective accuracy and efficacy are verified before integration within the larger system to form the final DSM. In this the key findings of this research can be summarised as follows:

1. In Section 3.6 the accuracy of the fully integrated TSM is verified both experimentally and by means of simulation with the proprietary model of Jonker Sailplanes. Here it is shown that the TSM's sink speed curve predictions are within 6.11% of the experimental results acquired by Idaflieg. The R^2 between these datasets is shown to be 0.88. When comparing the TSM's sink speed curve predictions with those of the JSCode, it is shown that a MAPE of 4.52% and an R^2 of 0.99 results, which

indicates that the error margin and explained variance of the TSM are within an acceptable threshold for this research.

2. In Section 4.5 the accuracy and efficiency as well as the Padulo completeness of the proposed parsimonious variable extraction framework are presented. It is shown that, for the 10 parsimonious variables that are compared with XFOIL's output, the proposed framework has an average mean absolute percentage error rate of 0.37%. The framework is also shown to adhere to most of Padulo's main criteria for an effective airfoil parametrisation method.

3. In Section 5.4 the data mining pipeline's efficacy is shown to be 30%, meaning that close to one-third of the airfoil's and 2D simulations generated from the uniform random sampling process either: (1) do not have converging parsimonious variable calculations, (2) do not pass through the multiple filtering stages, or (3) do not have converged XFOIL simulations. Also in this section, the final 2D airfoil performance dataset is discussed with 778 550 records which translates to 6 770 unique airfoils, each with 115 different operating condition simulations and the input/output schema as depicted in Table 5.4.

4. In Sections 6.6 and 6.7 the accuracy of the finalised PDLM is discussed. Here it is seen that the PDLM can accurately predict the values of the eight response variables in the hold-out dataset with an average MAPE of 5.52 and R^2 of 0.89. Given the diversity of Mach numbers, Reynolds numbers, Angles of Attack, and airfoil profile shape variance in this dataset, this level of accuracy is notable. Also, the versatility and predictive capacity of the PDLM is demonstrated where the single deep learning model can predict conventional and non-conventional response variables (Table 5.1) whereas in XFOIL, only the conventional response variables are predicted, each with its own complex mathematical formulation. The fact that the PDLM can predict all these response variables in a single model attests to the generalisation capacity and predictive power of deep neural networks.

5. In Sections 7.4 and 7.5 the airfoil generator module, developed with a BGMM, is shown to be highly effective in: (1) clustering airfoils, where airfoils with similar

shape and structural characteristics are grouped together, and (2) generating airfoil profiles with the same parsimonious variable distribution and local sequential structure as those found in the hold-out dataset.

6. In Section 7.6.4, the airfoil shape conversion module's accuracy is discussed. Here it is shown that this module is capable of constructing airfoils in terms of their x and y coordinate definition with an average MAE of 0.0038 (calculated on the hold-out dataset) given the 23 parsimonious variable descriptions of a realistic airfoil shape.

7. In Section 7.7 the airfoil generator and the airfoil shape conversion modules are combined to form the final airfoil shape mapping module. This is to allow for the final verification of the respective module's generalisation capacity by means of an airfoil fitting procedure as well as for the utilisation of this module to generate test case airfoils for both the TSM and DSM (Sections 8.2.1 and 8.4). The airfoil fitting procedure is done by fitting 100 randomly sampled airfoils from the UIUC airfoil database. It is shown that the airfoil shape mapping module can approximate these 100 airfoils with an average MSE between actual and generated profiles of 2.08×10^{-8} and a standard deviation of 5.48×10^{-10}.

8. In Section 9.3 the developed optimisation module is verified with two optimisation objectives each with a different airfoil, namely, a thick and a thin airfoil. Here it is shown that the optimisation module makes design decisions aligned with known sailplane design patterns required for cross-country performance improvement, especially for the thin airfoil case with lower lift generation where the optimisation module tends to produce planforms with larger area. The optimisation module is also shown to increase the span and area of the planform in accordance with expectations for the respective optimisation cases. Here, it is also shown that the optimisation module is able to find optimised performance while (1) adhering to the imposed constraints on the wing geometry and (2) producing realistic wing planforms.

9. In Section 9.4 the optimisation module is shown to be able to produce a new wing and airfoil design which is able to outperform the baseline design for multiple cross-

country tasks. Here, the optimised airfoil outperforms the baseline airfoil over the entire flight envelope present in the base case cross-country task. It is highlighted that the optimisation module learns to increase the area to improve the performance of the optimised design for weak flight conditions. Observations worth mentioning are that when combining the optimised wing with the baseline airfoil, slightly worse performance than the baseline design is seen. Alternatively, on combining the optimised airfoil with the baseline wing, significant performance increase is seen. Therefore it is concluded that the major performance increases of the optimised design are a combination of the planform and airfoil alterations and are not solely reliant on the increased span.

10. Finally, in Section 10.3, the verified optimisation module, after refinement for accuracy in the desired regions, is applied to optimise the performance of a SOTA glider design. The optimisation module is shown to achieve increased cross-country performance for two cases i.e. a constrained planform and an unconstrained planform case. In both optimisation cases, the performance increase is realised whilst adhering to all upfront explicitly defined constraints. Overall the best performing design has, on average, a 0.94% relative increase in cross-country performance when compared to the baseline design.

11.4 Recommendations

Although shown to be successful within the defined scope, there are additional improvements that can be made to the DSM to further the contributions of this research. These improvements are broken into four distinct themes: data mining, airfoil generation, different modelling approaches, and global optimisation.

For the data mining theme, the main recommendation is to increase the diversity of the airfoil population present in the final 2D airfoil performance and airfoil shape datasets. Throughout the data mining phase, for both the 2D performance and airfoil shape datasets, much care is taken to ensure that a diverse airfoil population is sampled. This airfoil pop-

ulation is, however, limited to the set of shapes capable of being generated by the SPM. To extend the diversity of the airfoil population, which holds the advantage of improving the shape generation as well as the IPDLM and PDLM modules, it is recommended that airfoils be generated with more than one constructive or deformative method.

For the airfoil generation theme, two recommendations are made. The first recommendation is concerned with the computational efficiency of the airfoil generator module. Although shown to be effective, the BGMM's use in the airfoil generator module added additional computational expense to the optimisation pipeline seeing that each airfoil required a total of 29 latent vectors and one class variable to generate realistic airfoil samples in terms of 23 parsimonious variables. This added computational overhead may be eliminated by employing, Explicitly Controllable GANs such as developed in (Shoshan et al., 2021) to generate airfoils by means of directly specifying the parsimonious shape and structural variables. The second recommendation here is to extend the parsimonious variable framework to allow for more explicit control over the trailing edge of the generated airfoil. This will enable full reconstruction and generation of SOTA airfoils which will facilitate further improvements of the JS4 sailplane design with the refined optimisation module.

For the theme of different modelling approaches, two recommendations are made. The first is concerned with the architecture of the IPDLM and PDLM modules. Here, it is recommended to also consider architectures that can efficiently model temporal and sequential data such as LSTMs or Transformers. This model will effectively be given an airfoil definition, in terms of parsimonious variables, together with its Mach and Reynolds numbers as input and predict the eight defined response variables for a sequence of predefined AoAs. This will hold the advantage of (1) possibly unlocking accuracy improvements seeing that these state-of-the-art methods can explicitly model sequential dependence, which vanilla MLPs can not do effectively, and (2) bringing computational improvements to the inference process seeing that a sequence of predictions can be made with a single model call. This is in opposition to the current PDLM setup where a single call is made for each unique operating condition definition.

The next recommendation with respect to the different modelling approaches theme is to possibly replace the entire TSM with a deep learning model. That is, with the TSM's accuracy validated both experimentally and by means of simulation, it is possible to use this model to mine sailplane cross-country performance data. In such a data mining pipeline, cross-country performance data can be mined for multiple sailplane configurations, airfoil designs, planform designs, and cross-country tasks. Such a dataset may then be used to develop a surrogate model, mapping sailplane designs and weather models to cross-country performance. This effectively eliminates the need for any further individual modelling by a cross-country model, sailplane performance model, 2D performance estimation model, and a LLT model. This model can then be used for direct and rapid design exploration and optimisation.

For the final theme, the global optimisation theme, two recommendation are made. The first recommendation is to make use of gradient descent for sailplane cross-country performance optimisation. This will require the replacement of the cross-country model, sailplane performance model, 2D performance estimation model, and the dynamic station LLT model with deep learning models. This holds the advantage of following the gradient across the entire solution to facilitate efficient and effective performance optimisation. The final recommendation here is to extend the TSM and DSM (and hence the underlying modules) to allow for the modelling of slats and flaps such that efficient and targeted sailplane optimisation is possible for sailplanes other than standard class designs.

11.5 Conclusion

The deep learning optimisation framework developed in this research is demonstrated to be an efficient and accurate tool for multi-objective constrained sailplane cross-country performance optimisation. This tool will aid designers and aerodynamicists in the rapid generation of feasible wing designs that adhere to constraints imposed on the airfoil profiles on the wing as well as the wing planform. It is hoped that the recommendations posed in this section will pave the way for additional research to further develop, improve, and

extend this framework to ultimately allow for more fine-grained control in the end-to-end sailplane cross-country performance optimisation process.

Appendix A

Intermediate dataset statistics

In Table A.1 we provide descriptive statistics for the dataset developed in Section 5.4.2. Here, indications of noisy data can be seen in the cases of the Slope feature (i.e. the lift-curve slope) that ranges between -50.12 and 18.56 where values in the vicinity of 2π are expected. It is also noted here that the maximum value of the camber angle lies at 116.76 degrees, further indicating impractical airfoil shapes.

Table A.1: Descriptive statistic for the response variables, operating conditions, shape features, and structural features for the intermediate dataset

Performance Target	mean	std	min	25%	50%	75%	max
Zero lift angle of attack	2.4992	10.3558	-10.0000	-6.7507	0.4004	10.1840	20.0000
Slope	3.2623	4.1081	-50.1223	1.6070	3.8044	6.0046	18.5557
Maximum lift	1.1154	0.7736	-0.8464	0.4575	1.0555	1.7482	3.0553
Lift coefficient	0.2750	0.7697	-1.9560	-0.2654	0.1133	0.7371	3.0553
Drag coefficient	0.0592	0.0464	0.0004	0.0241	0.0483	0.0806	0.4368
Moment coefficient	-0.0765	0.1359	-0.6162	-0.1581	-0.0700	0.0109	0.3195
Bottom transition location	0.4184	0.2269	0.0004	0.2387	0.4228	0.5890	1.0000
Top transition location	0.4029	0.2430	0.0000	0.2022	0.4106	0.5905	1.0000
Operating Condition Feature	mean	std	min	25%	50%	75%	max
Reynolds number	8.81×10^6	4.73×10^6	2.50×10^5	4.83×10^6	8.56×10^6	1.27×10^7	1.80×10^7
Mach number	0.15000	0.08657	0.00000	0.07600	0.15100	0.22400	0.30000
Angle of Attack	4.75363	8.65619	-10.00000	-2.75000	4.75000	12.25000	20.00000
Structural Feature	mean	std	min	25%	50%	75%	max
Area	0.2521	0.1140	0.0184	0.1579	0.2460	0.3421	0.5370
perimeter	2.3303	0.1802	2.0152	2.1818	2.3159	2.4599	2.9766
Y-centroid	0.0456	0.0514	-0.0112	0.0089	0.0250	0.0646	0.2628
X-centroid	0.4321	0.0346	0.3085	0.4121	0.4423	0.4603	0.4746
Ixx	0.0037	0.0037	0.0000	0.0007	0.0023	0.0057	0.0196
Iyy	0.0130	0.0062	0.0009	0.0079	0.0125	0.0177	0.0314
Shape Feature	mean	std	min	25%	50%	75%	max
Trailing edge angle	34.65	20.10	-12.25	18.12	36.28	51.94	73.69
Top curvature	2.37	1.46	0.05	1.34	2.00	3.06	12.24
Nose radius	0.12	0.08	0.00	0.05	0.11	0.19	0.96
Max top magnitude	0.25	0.11	0.03	0.16	0.24	0.32	0.67
Max top location	0.51	0.16	0.05	0.38	0.50	0.63	0.91
Max thickness magnitude	0.37	0.16	0.04	0.24	0.36	0.50	0.70
Max thickness location	0.39	0.11	0.13	0.30	0.42	0.49	0.55
Max camber magnitude	0.10	0.08	0.00	0.04	0.08	0.14	0.37
Max camber location	0.69	0.20	0.00	0.61	0.76	0.85	0.93
Max bottom upper magnitude	0.04	0.05	0.00	0.00	0.02	0.07	0.31
Max bottom upper location	0.86	0.21	0.00	0.84	0.92	0.97	1.00
Max bottom magnitude	-0.16	0.08	-0.36	-0.22	-0.15	-0.09	0.00
Max bottom location	0.32	0.14	0.02	0.22	0.32	0.42	0.76
Leading edge angle	5.57	15.01	-12.39	-3.46	3.17	9.36	84.58
Camber angle	40.21	21.58	0.01	22.98	41.62	56.45	116.76
Bottom upper curvature	4.62	4.38	0.00	0.00	3.73	7.64	20.16
Bottom curvature	2.24	1.60	0.03	1.32	1.90	2.64	20.78

Appendix B

Final dataset feature distribution

In Figure B.1 we provide violin plots of the feature distributions of the 3 operating condition input variables as calculated on the final 2D performance dataset as developed in Section 5.4.3. Here it is observed that, for all three features, the data is sampled from a uniform distribution. It can also be seen from this figure that, for the Reynolds number ranges between 0 to 500 000, there are fewer samples and hence in this region, XFOIL simulations converged the least.

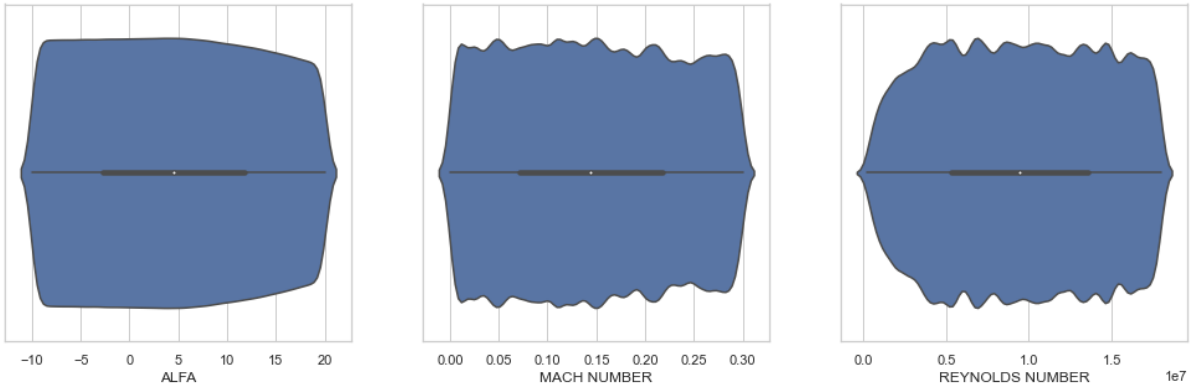


Figure B.1: Feature distribution of the operating conditions for the final 2D performance dataset

In Figure B.2 we provide violin plots of the feature distributions of the first eight parsimonious shape variables as calculated on the final 2D performance dataset as developed in Section 5.4.3. Here attention is drawn to the minimum and maximum bounds of the maximum camber magnitude and the maximum top magnitudes which are in the expected ranges for workable airfoil profiles for sailplane applications. Also here, the distribution of the maximum top magnitude feature indicates that the excessively thick airfoils of the intermediate dataset are removed from the final 2D performance dataset and that the distribution of the airfoil shape variables present here is within the acceptable range.

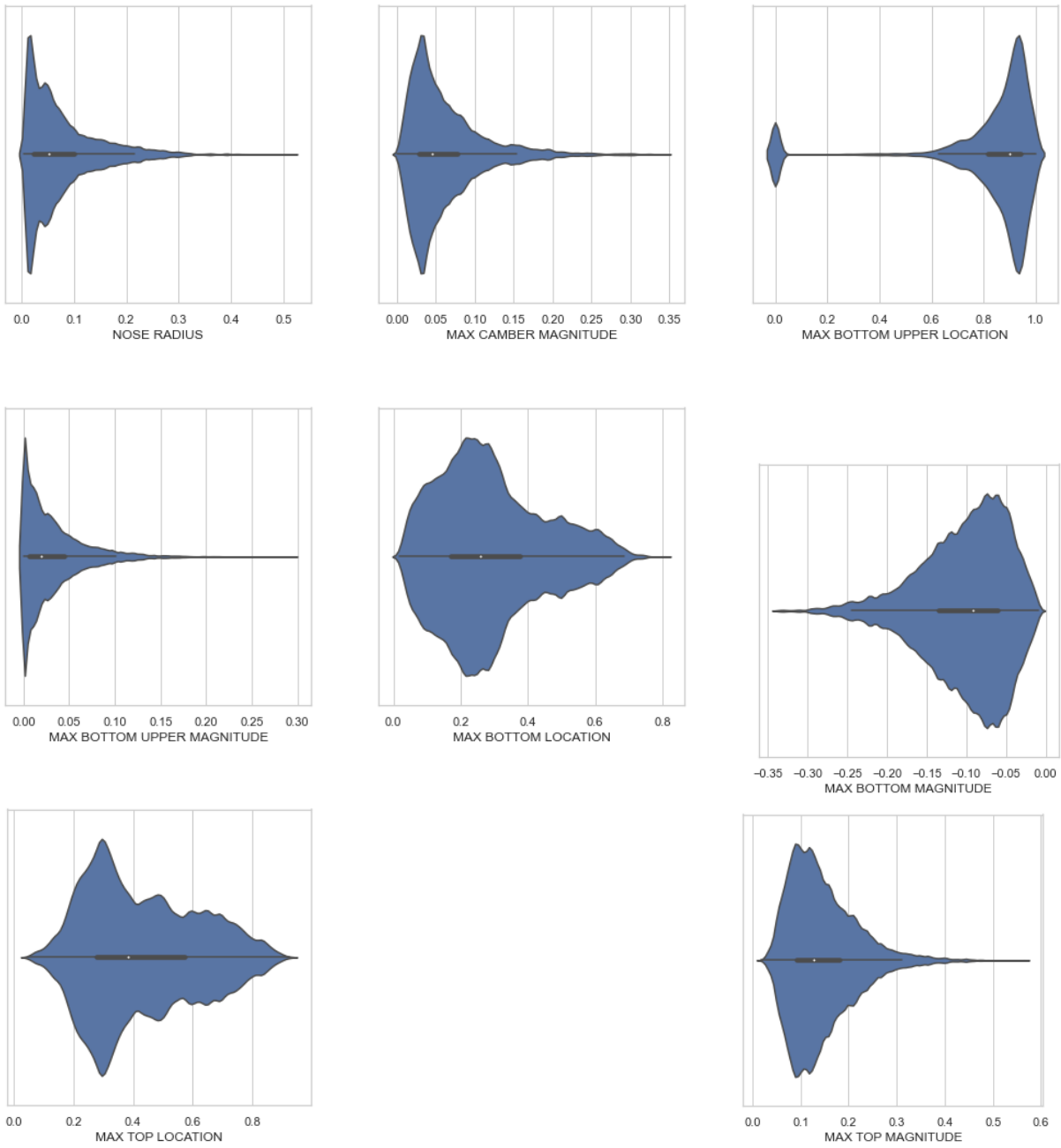


Figure B.2: Feature distributions of the first eight parsimonious shape variables for the final 2D performance dataset

In Figure B.3 we provide violin plots of the feature distributions of the final nine parsimonious shape variables as calculated on the final 2D performance dataset as developed in Section 5.4.3. Here attention is drawn to the minimum and maximum extremes of the location and magnitude of the total thickness, which are in the expected ranges for

workable airfoil profiles for sailplane applications.

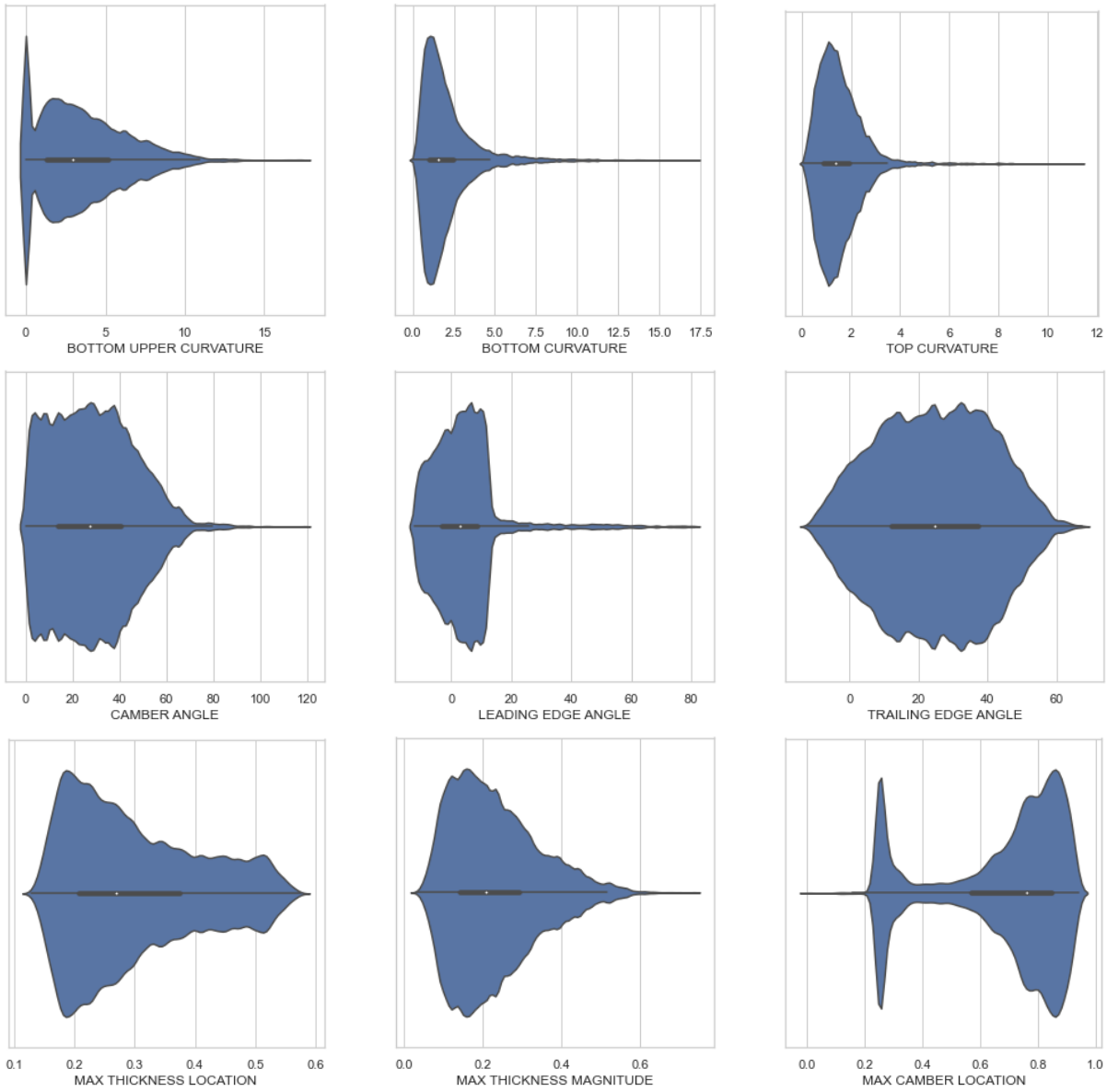


Figure B.3: Feature distributions of the final nine parsimonious shape variables for the final 2D performance dataset

In Figure B.4 we provide violin plots of the feature distributions of the parsimonious structural variables as calculated on the final 2D performance dataset as developed in Section 5.4.3. Here attention is drawn to the minimum and maximum bounds of the x-centroid and the y-centroid which are in the expected ranges for workable airfoil profiles for sailplane applications.

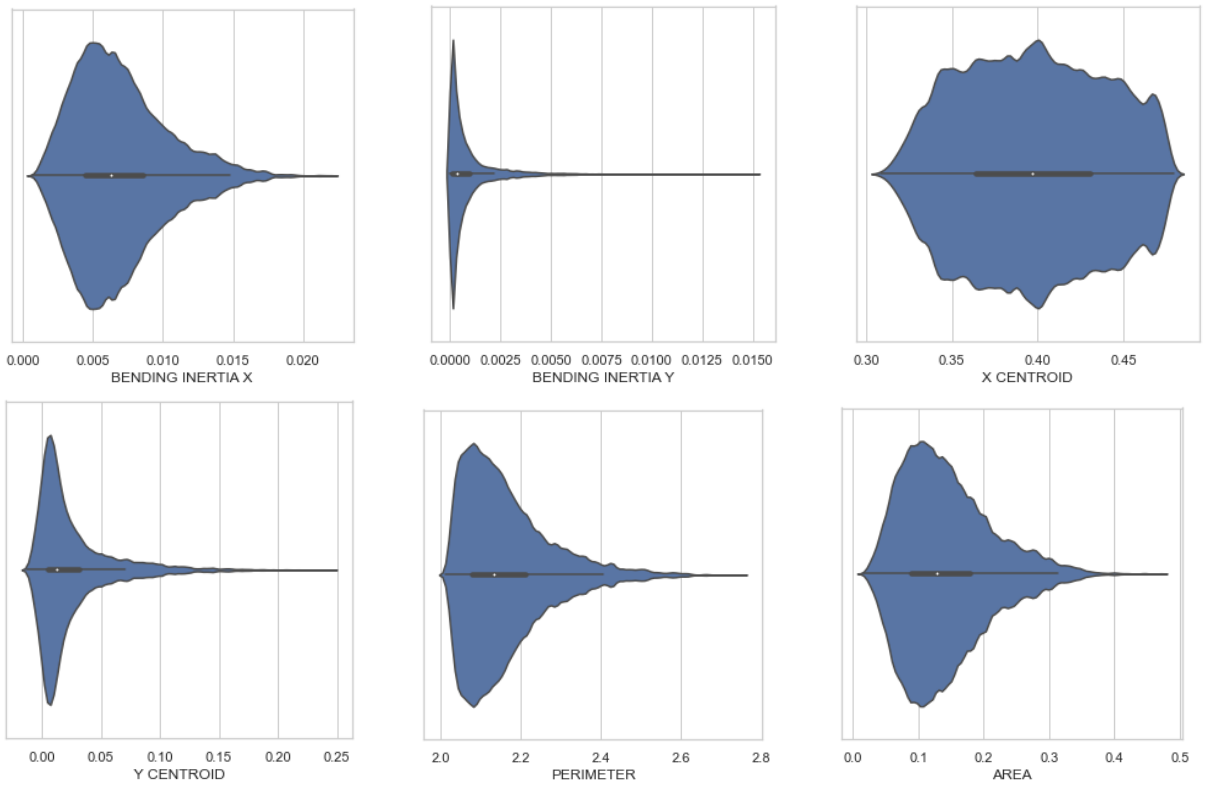


Figure B.4: Feature distribution of the structural variables for the final 2D performance dataset

In Figure B.5 we provide violin plots of the feature distributions of the eight conventional and non-conventional response variables as calculated on the final 2D performance dataset as developed in Section 5.4.3. Here attention is drawn to the minimum and maximum bounds of the top and bottom transition locations indicating that very few airfoils have complete laminar flow over these respective surfaces. Also, attention is drawn to the slope (i.e. the lift-curve slope) distribution, highlighting the effectiveness of the filtering pipeline.

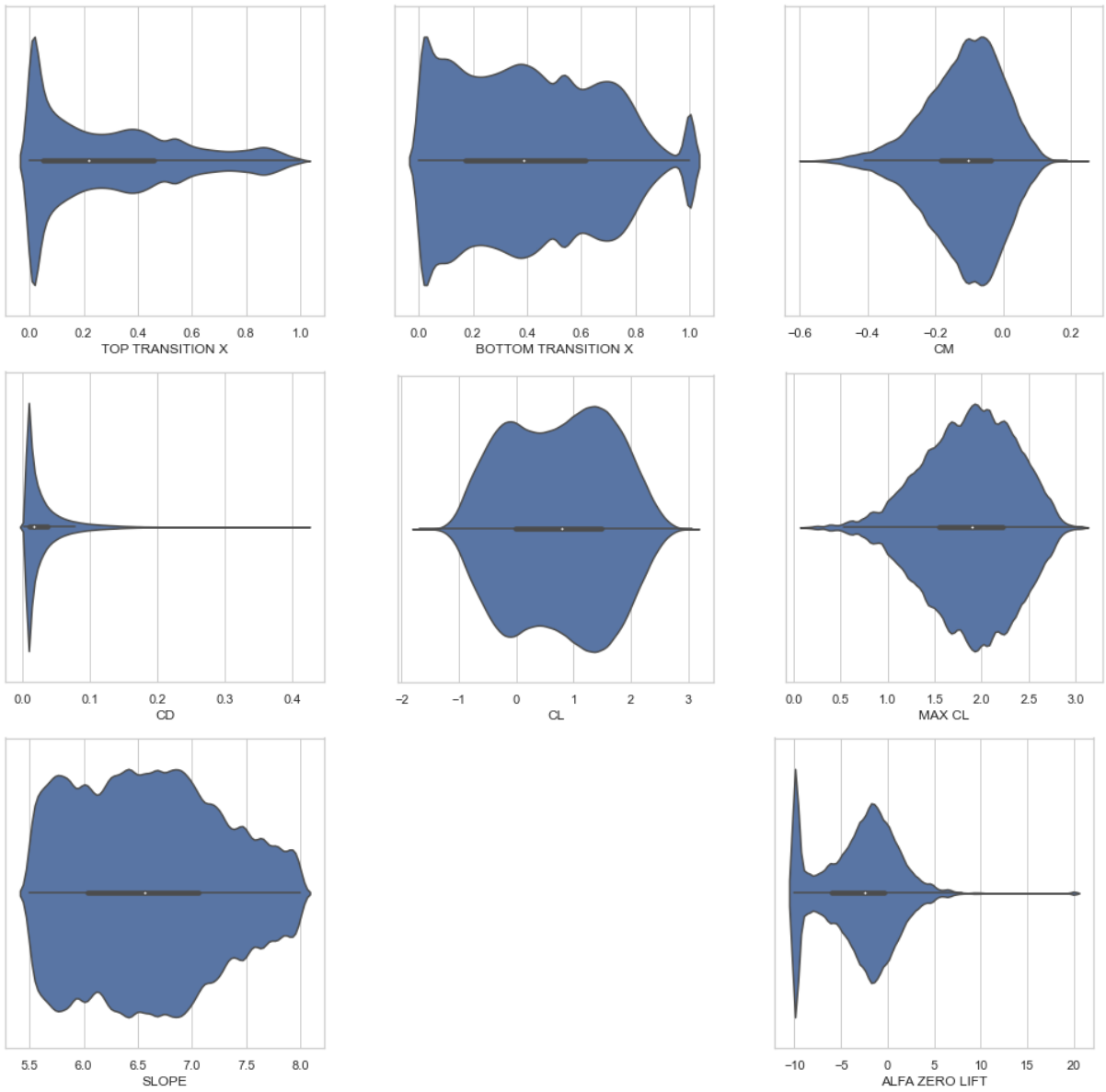


Figure B.5: Feature distribution of the response variables for the final 2D performance dataset

Appendix C

Supplementary cluster analysis

In Figure C.1, 2D cuts of the 3D t-SNE analysis as discussed in Section 7.4.1, is depicted. Here the 50 clusters are colour-coded and projected onto their 3D components as determined by the t-SNE algorithm. From this figure it can be seen that there is effective separation between most of the airfoil classes, indicating effective clustering by the BGMM-based airfoil generator module.

Figure C.1: Results of the t-SNE analysis to understand the quality of the 50 generated clusters

In Table C.1 the normalised descriptive statistics for each parsimonious feature and each airfoil cluster (as discussed in Section 7.4.3), are shown. This multivariate dataset can be viewed by zooming in and is included here to clarify the abstract concept discussed in Section 7.4.3 on the calculation of the parsimonious feature importance for cluster separation.

Table C.1: Descriptive statistics for each of the 23 parsimonious variables for each of the 50 BGMM clusters

Feature	Statistic	Cluster																																											
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	19	20	21	22	23	24	25	26	27	28	29	30	31	32	34	36	37	38	39	40	42	43	44	45	46	47	48	49
Max Top Magnitude	Min	0.100	0.215	0.057	0.310	0.227	0.143	0.076	0.035	0.007	0.102	0.096	0.035	0.219	0.147	0.000	0.183	0.129	0.002	0.349	0.065	0.101	0.028	0.188	0.036	0.266	0.136	0.091	0.387	0.050	0.220	0.172	0.002	0.164	0.026	0.111	0.069	0.023	0.365	0.008	0.023	0.208	0.113	0.080	0.014	0.007	0.315
Max Top Magnitude	Max	0.468	0.645	0.675	0.941	0.616	0.558	0.357	0.917	0.290	0.748	0.319	0.527	0.525	0.377	0.332	0.377	0.996	0.254	0.790	0.445	0.938	0.360	0.654	0.719	0.869	0.671	0.907	0.895	0.885	0.808	0.432	0.314	0.957	0.063	0.542	0.483	0.598	0.612	0.634	0.504	0.703	1.000	0.666	0.925	0.348	0.672
Max Top Magnitude	Std.dev	0.103	0.066	0.149	0.114	0.072	0.077	0.072	0.140	0.061	0.097	0.044	0.126	0.074	0.051	0.072	0.064	0.172	0.048	0.074	0.060	0.138	0.074	0.072	0.107	0.113	0.090	0.120	0.097	0.120	0.085	0.074	0.046	0.138	0.015	0.083	0.065	0.100	0.054	0.085	0.089	0.086	0.174	0.082	0.201	0.070	0.070
Max Top Magnitude	Median	0.317	0.438	0.329	0.503	0.351	0.383	0.259	0.336	0.106	0.443	0.188	0.286	0.350	0.224	0.182	0.323	0.628	0.080	0.514	0.230	0.557	0.175	0.345	0.220	0.422	0.321	0.369	0.636	0.441	0.507	0.311	0.119	0.523	0.044	0.347	0.269	0.166	0.503	0.191	0.247	0.344	0.538	0.378	0.301	0.156	0.481
Max Top Magnitude	25th percentile	0.229	0.388	0.215	0.432	0.308	0.326	0.167	0.237	0.082	0.370	0.162	0.174	0.298	0.176	0.119	0.221	0.485	0.053	0.473	0.194	0.480	0.125	0.304	0.146	0.373	0.265	0.290	0.584	0.365	0.456	0.225	0.083	0.420	0.032	0.289	0.230	0.104	0.458	0.144	0.174	0.300	0.418	0.315	0.190	0.107	0.436
Max Top Magnitude	75th percentile	0.408	0.480	0.494	0.594	0.410	0.430	0.288	0.432	0.148	0.508	0.224	0.379	0.415	0.252	0.234	0.335	0.741	0.115	0.553	0.274	0.638	0.244	0.390	0.302	0.504	0.392	0.434	0.698	0.500	0.559	0.352	0.151	0.617	0.053	0.398	0.315	0.254	0.534	0.241	0.312	0.401	0.659	0.438	0.505	0.214	0.526
Max Top Location	Min	0.394	0.315	0.079	0.568	0.484	0.187	0.527	0.215	0.259	0.633	0.533	0.281	0.304	0.310	0.258	0.338	0.185	0.122	0.549	0.544	0.290	0.234	0.517	0.048	0.409	0.137	0.189	0.023	0.312	0.345	0.368	0.450	0.218	0.245	0.218	0.350	0.133	0.572	0.037	0.071	0.355	0.023	0.357	0.000	0.033	0.525
Max Top Location	Max	0.763	0.969	0.987	0.952	0.947	0.604	0.825	0.546	0.481	0.994	0.967	0.510	0.900	0.527	0.996	0.581	0.758	0.406	0.949	0.995	0.566	0.594	0.892	0.490	0.957	0.947	0.615	0.820	0.742	0.969	0.701	1.000	0.980	0.255	0.721	0.992	0.379	0.789	0.992	0.753	0.944	0.700	0.990	0.988	0.995	0.833
Max Top Location	Std.dev	0.085	0.133	0.345	0.074	0.110	0.079	0.065	0.052	0.047	0.075	0.108	0.048	0.195	0.031	0.159	0.064	0.089	0.037	0.111	0.096	0.051	0.086	0.100	0.075	0.131	0.182	0.073	0.218	0.083	0.108	0.087	0.103	0.103	0.004	0.113	0.064	0.046	0.041	0.210	0.122	0.135	0.142	0.070	0.225	0.274	0.036
Max Top Location	Median	0.555	0.483	0.311	0.845	0.742	0.414	0.663	0.365	0.331	0.937	0.885	0.378	0.550	0.360	0.428	0.513	0.565	0.261	0.844	0.791	0.427	0.395	0.643	0.264	0.849	0.612	0.396	0.719	0.530	0.851	0.598	0.885	0.833	0.248	0.447	0.949	0.246	0.713	0.706	0.396	0.693	0.380	0.952	0.445	0.765	0.741
Max Top Location	25th percentile	0.519	0.411	0.246	0.787	0.664	0.358	0.619	0.331	0.308	0.868	0.826	0.349	0.336	0.341	0.349	0.463	0.500	0.241	0.673	0.724	0.387	0.319	0.602	0.224	0.651	0.435	0.347	0.600	0.469	0.782	0.509	0.769	0.743	0.247	0.356	0.920	0.217	0.680	0.563	0.313	0.601	0.272	0.915	0.272	0.361	0.716
Max Top Location	75th percentile	0.620	0.585	0.950	0.890	0.861	0.469	0.696	0.399	0.367	0.959	0.906	0.420	0.750	0.373	0.578	0.528	0.619	0.283	0.886	0.878	0.456	0.470	0.760	0.332	0.900	0.736	0.449	0.748	0.593	0.899	0.653	0.951	0.906	0.250	0.550	0.970	0.280	0.739	0.883	0.502	0.812	0.504	0.971	0.564	0.872	0.762
Max Bottom Magnitude	Min	0.311	0.003	0.000	0.025	0.243	0.039	0.363	0.047	0.451	0.108	0.382	0.033	0.112	0.344	0.410	0.369	0.097	0.544	0.165	0.433	0.152	0.303	0.599	0.123	0.273	0.075	0.083	0.059	0.001	0.063	0.159	0.571	0.002	0.929	0.019	0.472	0.281	0.048	0.593	0.276	0.049	0.036	0.207	0.255	0.293	0.792
Max Bottom Magnitude	Max	0.958	0.631	0.866	0.944	0.892	0.813	0.921	0.999	0.974	0.998	0.973	0.933	0.674	0.747	0.991	0.677	0.997	0.997	0.996	0.990	0.955	0.969	0.990	0.963	0.968	0.903	0.946	1.000	0.950	0.991	0.643	0.998	0.965	0.970	0.819	0.998	0.998	0.567	1.000	0.996	0.699	1.000	0.998	1.000	0.990	0.995
Max Bottom Magnitude	Std.dev	0.162	0.117	0.210	0.223	0.131	0.159	0.145	0.215	0.121	0.208	0.149	0.234	0.161	0.091	0.139	0.101	0.202	0.077	0.125	0.116	0.180	0.162	0.072	0.181	0.144	0.158	0.201	0.276	0.208	0.178	0.136	0.088	0.211	0.015	0.152	0.096	0.154	0.106	0.069	0.151	0.127	0.200	0.115	0.175	0.160	0.033
Max Bottom Magnitude	Median	0.810	0.207	0.471	0.393	0.598	0.292	0.518	0.528	0.834	0.618	0.784	0.480	0.294	0.612	0.684	0.483	0.790	0.900	0.888	0.771	0.528	0.678	0.900	0.686	0.666	0.462	0.484	0.305	0.360	0.622	0.405	0.868	0.476	0.957	0.346	0.893	0.855	0.238	0.939	0.682	0.389	0.797	0.843	0.803	0.785	0.950
Max Bottom Magnitude	25th percentile	0.674	0.133	0.232	0.242	0.497	0.196	0.445	0.356	0.749	0.446	0.627	0.304	0.212	0.550	0.585	0.417	0.595	0.859	0.835	0.681	0.386	0.535	0.852	0.530	0.555	0.349	0.337	0.199	0.207	0.485	0.328	0.799	0.293	0.947	0.256	0.811	0.693	0.174	0.888	0.564	0.293	0.634	0.755	0.640	0.657	0.924
Max Bottom Magnitude	75th percentile	0.875	0.304	0.600	0.600	0.691	0.418	0.678	0.705	0.892	0.798	0.859	0.686	0.466	0.679	0.808	0.594	0.928	0.934	0.925	0.861	0.673	0.815	0.939	0.804	0.768	0.579	0.657	0.570	0.533	0.745	0.536	0.921	0.640	0.961	0.450	0.951	0.918	0.325	0.974	0.788	0.475	0.899	0.912	0.904	0.875	0.974
Max Bottom Location	Min	0.035	0.347	0.117	0.059	0.144	0.143	0.293	0.159	0.514	0.001	0.232	0.315	0.317	0.519	0.043	0.451	0.000	0.043	0.015	0.048	0.325	0.173	0.016	0.026	0.048	0.113	0.146	0.058	0.095	0.005	0.319	0.017	0.033	0.640	0.184	0.001	0.013	0.282	0.000	0.057	0.215	0.001	0.001	0.000	0.050	0.004
Max Bottom Location	Max	0.462	0.622	0.523	0.587	0.563	0.597	0.496	0.787	0.765	0.682	0.618	0.719	0.588	0.691	0.869	0.612	0.321	0.920	0.501	0.684	0.846	0.704	0.485	0.683	0.558	0.518	0.723	0.970	0.788	0.538	0.633	0.713	0.573	0.721	0.607	0.648	0.905	0.504	0.996	0.590	0.626	0.996	0.620	1.000	0.663	0.120
Max Bottom Location	Std.dev	0.100	0.056	0.103	0.119	0.087	0.074	0.049	0.117	0.048	0.107	0.051	0.067	0.083	0.037	0.107	0.035	0.079	0.229	0.121	0.127	0.111	0.095	0.121	0.189	0.136	0.059	0.102	0.201	0.113	0.105	0.083	0.115	0.088	0.034	0.069	0.196	0.273	0.043	0.231	0.089	0.062	0.300	0.135	0.215	0.075	0.025
Max Bottom Location	Median	0.185	0.501	0.368	0.419	0.347	0.413	0.364	0.589	0.642	0.351	0.315	0.586	0.392	0.624	0.526	0.503	0.076	0.659	0.196	0.311	0.592	0.406	0.088	0.394	0.316	0.332	0.401	0.327	0.429	0.264	0.422	0.293	0.290	0.669	0.382	0.378	0.123	0.375	0.160	0.267	0.425	0.372	0.244	0.139	0.239	0.069
Max Bottom Location	25th percentile	0.123	0.450	0.243	0.330	0.279	0.363	0.333	0.509	0.605	0.299	0.287	0.538	0.352	0.601	0.441	0.485	0.023	0.379	0.085	0.229	0.511	0.355	0.058	0.164	0.152	0.291	0.349	0.281	0.361	0.183	0.367	0.262	0.232	0.656	0.344	0.176	0.063	0.340	0.061	0.205	0.388	0.163	0.168	0.060	0.204	0.049
Max Bottom Location	75th percentile	0.283	0.535	0.432	0.472	0.405	0.464	0.407	0.652	0.670	0.421	0.355	0.629	0.502	0.647	0.597	0.522	0.155	0.707	0.289	0.374	0.687	0.484	0.157	0.534	0.405	0.371	0.479	0.365	0.511	0.327	0.507	0.344	0.352	0.709	0.434	0.531	0.437	0.405	0.317	0.318	0.477	0.746	0.320	0.213	0.271	0.085
Max Bottom Upper Magnitude	Min	0.000	0.016	0.010	0.021	0.009	0.013	0.000	0.000	0.000	0.056	0.004	0.000	0.122	0.000	0.000	0.018	0.000	0.000	0.132	0.005	0.000	0.001	0.008	0.000	0.086	0.005	0.000	0.010	0.000	0.035	0.000	0.001	0.026	0.053	0.000	0.085	0.000	0.017	0.000	0.000	0.000	0.000	0.031	0.000	0.000	0.143
Max Bottom Upper Magnitude	Max	0.081	0.723	0.896	0.640	0.546	0.634	0.093	0.120	0.075	0.981	0.461	0.016	0.439	0.013	0.222	0.238	0.925	0.140	0.992	0.593	0.010	0.252	0.599	0.258	0.782	0.425	0.052	0.914	0.032	0.995	0.023	0.590	0.985	0.100	0.151	0.878	0.070	0.151	0.652	0.184	0.306	0.891	1.000	0.802	0.597	0.980
Max Bottom Upper Magnitude	Std.dev	0.015	0.135	0.268	0.089	0.131	0.132	0.021	0.014	0.014	0.205	0.107	0.002	0.091	0.002	0.046	0.073	0.204	0.024	0.185	0.112	0.002	0.057	0.139	0.062	0.121	0.090	0.003	0.234	0.005	0.168	0.004	0.095	0.177	0.019	0.033	0.122	0.018	0.031	0.131	0.038	0.084	0.189	0.156	0.150	0.130	0.119
Max Bottom Upper Magnitude	Median	0.009	0.205	0.236	0.372	0.188	0.219	0.005	0.000	0.012	0.569	0.224	0.000	0.304	0.000	0.044	0.085	0.174	0.012	0.706	0.192	0.000	0.090	0.254	0.029	0.452	0.193	0.001	0.119	0.002	0.655	0.002	0.144	0.537	0.075	0.027	0.450	0.000	0.054	0.203	0.017	0.112	0.106	0.662	0.214	0.162	0.715
Max Bottom Upper Magnitude	25th percentile	0.003	0.114	0.136	0.314	0.058	0.120	0.002	0.000	0.004	0.409	0.149	0.000	0.213	0.000	0.011	0.046	0.077	0.000	0.534	0.116	0.000	0.051	0.162	0.009	0.370	0.124	0.000	0.050	0.000	0.524	0.001	0.080	0.413	0.058	0.009	0.379	0.000	0.037	0.078	0.005	0.036	0.019	0.541	0.110	0.072	0.618
Max Bottom Upper Magnitude	75th percentile	0.016	0.311	0.648	0.429	0.298	0.324	0.015	0.000	0.022	0.712	0.284	0.001	0.354	0.001	0.085	0.174	0.359	0.034	0.823	0.275	0.001	0.139	0.355	0.093	0.526	0.254	0.003	0.207	0.006	0.753	0.005	0.224	0.651	0.088	0.057	0.542	0.011	0.082	0.302	0.052	0.188	0.316	0.777	0.321	0.281	0.790
Max Bottom Upper Location	Min	0.861	0.893	0.867	0.705	0.732	0.868	0.885	0.000	0.899	0.706	0.816	0.000	0.890	0.000	0.000	0.915	0.000	0.000	0.599	0.666	0.977	0.855	0.597	0.846	0.661	0.774	0.905	0.017	0.975	0.596	0.975	0.661	0.674	0.908	0.908	0.622	0.000	0.912	0.014	0.670	0.880	0.013	0.614	0.016	0.688	0.516
Max Bottom Upper Location	Max	1.000	0.975	0.968	0.956	0.974	0.975	1.000	0.167	1.000	0.941	0.981	1.000	0.944	1.000	1.000	0.975	0.997	1.000	0.932	0.973	1.000	0.989	0.964	1.000	0.915	0.975	1.000	0.980	1.000	0.915	1.000	0.985	0.965	0.935	1.000	0.932	0.987	0.975	0.999	1.000	0.999	1.000	0.950	1.000	0.997	0.790
Max Bottom Upper Location	Std.dev	0.022	0.014	0.014	0.050	0.050	0.018	0.023	0.023	0.022	0.038	0.035	0.492	0.012	0.498	0.209	0.015	0.173	0.424	0.083	0.066	0.004	0.019	0.081	0.030	0.068	0.030	0.009	0.302	0.007	0.062	0.007	0.051	0.052	0.012	0.020	0.048	0.321	0.015	0.227	0.067	0.020	0.411	0.055	0.241	0.061	0.031
Max Bottom Upper Location	Median	0.973	0.938	0.922	0.906	0.909	0.927	0.983	0.000	0.964	0.906	0.888	0.985	0.925	0.000	0.945	0.940	0.675	0.918	0.822	0.881	0.998	0.927	0.741	0.953	0.869	0.900	0.992	0.922	0.994	0.857	0.992	0.894	0.868	0.928	0.959	0.897	0.000	0.948	0.817	0.937	0.942	0.604	0.894	0.798	0.879	0.718
Max Bottom Upper Location	25th percentile	0.965	0.930	0.914	0.866	0.865	0.918	0.970	0.000	0.951	0.881	0.867	0.000	0.919	0.000	0.928	0.929	0.578	0.000	0.699	0.810	0.995	0.915	0.698	0.924	0.770	0.879	0.985	0.887	0.987	0.823	0.987	0.856	0.820	0.913	0.946	0.874	0.000	0.936	0.701	0.874	0.931	0.073	0.863	0.623	0.823	0.699
Max Bottom Upper Location	75th percentile	0.988	0.949	0.929	0.920	0.928	0.940	0.991	0.000	0.975	0.918	0.917	0.998	0.929	0.992	0.966	0.951	0.847	0.947	0.854	0.912	1.000	0.935	0.808	0.971	0.889	0.918	0.998	0.943	0.998	0.878	0.997	0.914	0.900	0.933	0.977	0.911	0.239	0.960	0.876	0.971	0.963	0.960	0.910	0.873	0.912	0.739
Max Camber Magnitude	Min	0.043	0.085	0.036	0.262	0.139	0.059	0.025	0.001	0.006	0.177	0.085	0.002	0.216	0.000	0.001	0.134	0.138	0.003	0.473	0.079	0.023	0.010	0.131	0.006	0.259	0.048	0.006	0.288	0.004	0.212	0.010	0.013	0.181	0.052	0.001	0.134	0.018	0.200	0.008	0.002	0.020	0.019	0.153	0.006	0.004	0.360
Max Camber Magnitude	Max	0.420	0.887	0.938	0.975	0.689	0.716	0.254	0.801	0.102	0.982	0.490	0.091	0.626	0.090	0.283	0.301	0.961	0.139	0.994	0.605	0.811	0.291	0.708	0.532	0.950	0.602	0.813	0.948	0.734	0.988	0.118	0.577	1.000	0.099	0.379	0.866	0.489	0.485	0.648	0.414	0.571	0.988	0.979	0.917	0.585	0.992
Max Camber Magnitude	Std.dev	0.087	0.138	0.291	0.114	0.105	0.122	0.053	0.120	0.021	0.163	0.087	0.022	0.113	0.014	0.055	0.047	0.159	0.024	0.107	0.096	0.166	0.061	0.104	0.063	0.117	0.098	0.117	0.159	0.118	0.127	0.029	0.084	0.163	0.019	0.066	0.115	0.055	0.059	0.110	0.080	0.111	0.220	0.149	0.220	0.124	0.105
Max Camber Magnitude	Median	0.241	0.352	0.278	0.589	0.369	0.314	0.082	0.087	0.067	0.655	0.260	0.046	0.402	0.078	0.104	0.206	0.638	0.057	0.813	0.285	0.298	0.147	0.449	0.079	0.596	0.313	0.087	0.543	0.129	0.774	0.072	0.171	0.715	0.074	0.106	0.467	0.083	0.330	0.283	0.118	0.267	0.471	0.668	0.345	0.185	0.787
Max Camber Magnitude	25th percentile	0.162	0.262	0.174	0.520	0.280	0.229	0.063	0.063	0.048	0.522	0.196	0.029	0.301	0.071	0.071	0.181	0.521	0.038	0.733	0.227	0.182	0.100	0.384	0.061	0.522	0.250	0.056	0.459	0.083	0.679	0.038	0.116	0.585	0.058	0.067	0.396	0.070	0.286	0.193	0.074	0.173	0.326	0.554	0.216	0.105	0.709
Max Camber Magnitude	75th percentile	0.298	0.456	0.754	0.678	0.439	0.414	0.104	0.136	0.078	0.785	0.321	0.064	0.468	0.086	0.152	0.256	0.759	0.075	0.872	0.353	0.428	0.193	0.515	0.108	0.680	0.379	0.149	0.689	0.212	0.841	0.086	0.237	0.825	0.086	0.159	0.552	0.115	0.374	0.344	0.183	0.355	0.680	0.778	0.552	0.300	0.858
Max Camber Location	Min	0.395	0.864	0.867	0.632	0.665	0.835	0.698	0.064	0.265	0.724	0.789	0.265	0.904	0.265	0.265	0.887	0.252	0.264	0.600	0.671	0.055	0.836	0.593	0.265	0.598	0.625	0.002	0.029	0.044	0.472	0.803	0.705	0.695	0.972	0.692	0.617	0.102	0.782	0.045	0.063	0.663	0.029	0.610	0.000	0.713	0.535
Max Camber Location	Max	0.870	1.000	1.000	0.984	0.979	0.990	0.973	0.970	0.394	0.990	0.975	0.367	0.989	0.888	1.000	0.981	0.812	0.352	0.949	0.990	0.645	0.998	0.915	0.997	0.968	0.983	0.951	0.904	0.997	0.972	0.997	0.998	0.985	0.998	0.999	0.989	0.374	0.933	0.990	0.950	0.994	0.870	0.989	0.990	0.992	0.831
Max Camber Location	Std.dev	0.124	0.021	0.023	0.070	0.075	0.026	0.057	0.064	0.026	0.052	0.029	0.010	0.022	0.071	0.205	0.027	0.097	0.014	0.102	0.077	0.095	0.033	0.092	0.349	0.105	0.063	0.219	0.262	0.208	0.083	0.073	0.067	0.072	0.012	0.059	0.056	0.038	0.031	0.209	0.177	0.048	0.218	0.061	0.229	0.063	0.034
Max Camber Location	Median	0.610	0.966	0.971	0.920	0.848	0.958	0.850	0.282	0.270	0.954	0.919	0.269	0.964	0.268	0.946	0.931	0.620	0.270	0.866	0.860	0.362	0.949	0.697	0.274	0.896	0.896	0.412	0.821	0.749	0.886	0.850	0.923	0.880	0.992	0.937	0.953	0.277	0.874	0.775	0.742	0.946	0.401	0.953	0.672	0.892	0.754
Max Camber Location	25th percentile	0.533	0.951	0.957	0.863	0.794	0.938	0.820	0.270	0.268	0.903	0.901	0.267	0.951	0.266	0.888	0.916	0.535	0.268	0.707	0.800	0.314	0.923	0.656	0.268	0.721	0.852	0.295	0.675	0.531	0.841	0.819	0.854	0.812	0.977	0.875	0.927	0.268	0.848	0.657	0.615	0.921	0.193	0.919	0.535	0.834	0.732
Max Camber Location	75th percentile	0.759	0.976	0.978	0.947	0.924	0.972	0.908	0.309	0.279	0.970	0.937	0.271	0.971	0.269	0.972	0.961	0.676	0.274	0.898	0.931	0.409	0.969	0.779	0.976	0.928	0.928	0.744	0.848	0.841	0.917	0.961	0.961	0.937	0.996	0.970	0.971	0.312	0.892	0.898	0.812	0.963	0.579	0.970	0.866	0.942	0.775
Max Thickness Magnitude	Min	0.139	0.377	0.141	0.163	0.107	0.247	0.136	0.053	0.013	0.011	0.013	0.063	0.359	0.226	0.011	0.334	0.011	0.001	0.012	0.010	0.115	0.017	0.068	0.054	0.097	0.209	0.148	0.068	0.089	0.018	0.316	0.000	0.037	0.017	0.222	0.001	0.014	0.585	0.006	0.059	0.395	0.001	0.001	0.006	0.008	0.011
Max Thickness Magnitude	Max	0.688	1.000	0.988	0.999	0.812	0.983	0.643	0.997	0.566	0.945	0.600	0.993	0.919	0.688	0.608	0.698	0.998	0.518	0.986	0.551	1.000	0.670	0.542	0.919	0.894	0.956	0.934	0.994	1.000	0.976	0.856	0.396	0.996	0.067	0.958	0.611	0.801	0.990	0.640	0.773	0.987	0.991	0.868	0.912	0.704	0.342
Max Thickness Magnitude	Std.dev	0.149	0.118	0.204	0.199	0.129	0.151	0.145	0.227	0.124	0.228	0.154	0.248	0.147	0.103	0.149	0.118	0.269	0.102	0.167	0.111	0.174	0.157	0.105	0.190	0.146	0.153	0.178	0.236	0.181	0.177	0.157	0.082	0.228	0.020	0.152	0.106	0.183	0.092	0.111	0.147	0.135	0.222	0.132	0.212	0.159	0.076
Max Thickness Magnitude	Median	0.388	0.841	0.589	0.703	0.478	0.750	0.471	0.574	0.160	0.394	0.182	0.540	0.692	0.396	0.334	0.595	0.593	0.112	0.220	0.252	0.813	0.338	0.248	0.389	0.470	0.578	0.649	0.869	0.774	0.449	0.617	0.109	0.590	0.038	0.680	0.137	0.224	0.833	0.119	0.401	0.638	0.590	0.164	0.406	0.197	0.124
Max Thickness Magnitude	25th percentile	0.285	0.743	0.442	0.528	0.390	0.632	0.298	0.381	0.113	0.197	0.100	0.327	0.567	0.314	0.205	0.426	0.320	0.058	0.136	0.171	0.668	0.227	0.195	0.265	0.370	0.461	0.496	0.741	0.621	0.337	0.425	0.054	0.393	0.022	0.570	0.070	0.137	0.750	0.065	0.297	0.548	0.421	0.089	0.231	0.107	0.061
Max Thickness Magnitude	75th percentile	0.510	0.914	0.804	0.858	0.565	0.839	0.553	0.748	0.251	0.595	0.338	0.727	0.794	0.462	0.440	0.638	0.796	0.185	0.320	0.333	0.900	0.485	0.338	0.557	0.578	0.694	0.768	0.947	0.885	0.582	0.698	0.178	0.783	0.053	0.770	0.225	0.416	0.889	0.201	0.531	0.745	0.766	0.265	0.584	0.313	0.178
Max Thickness Location	Min	0.634	0.512	0.272	0.597	0.528	0.313	0.639	0.330	0.569	0.352	0.419	0.442	0.516	0.692	0.506	0.842	0.238	0.233	0.439	0.406	0.497	0.342	0.554	0.247	0.318	0.309	0.316	0.680	0.523	0.243	0.683	0.235	0.239	0.287	0.347	0.231	0.243	0.619	0.234	0.243	0.663	0.248	0.233	0.000	0.236	0.536
Max Thickness Location	Max	0.994	0.995	0.978	0.998	0.989	0.959	0.993	0.996	0.977	1.000	0.980	0.995	0.915	0.930	0.993	0.927	1.000	0.899	0.992	0.999	0.995	0.895	0.990	0.869	0.984	0.926	0.842	0.998	1.000	0.996	0.982	0.995	0.998	0.407	0.972	0.957	0.777	0.990	0.998	0.912	0.995	0.990	0.808	0.930	0.836	0.996
Max Thickness Location	Std.dev	0.060	0.078	0.192	0.049	0.077	0.133	0.060	0.164	0.081	0.117	0.110	0.107	0.106	0.060	0.093	0.025	0.186	0.154	0.111	0.108	0.104	0.120	0.081	0.171	0.157	0.138	0.108	0.051	0.071	0.195	0.042	0.203	0.178	0.044	0.136	0.213	0.119	0.064	0.223	0.137	0.034	0.176	0.132	0.146	0.138	0.094
Max Thickness Location	Median	0.937	0.891	0.504	0.915	0.891	0.698	0.919	0.816	0.883	0.893	0.840	0.898	0.661	0.862	0.879	0.885	0.850	0.415	0.868	0.876	0.880	0.627	0.898	0.430	0.723	0.646	0.654	0.934	0.905	0.551	0.915	0.668	0.722	0.342	0.675	0.677	0.350	0.912	0.761	0.504	0.919	0.814	0.441	0.387	0.405	0.883
Max Thickness Location	25th percentile	0.914	0.849	0.394	0.887	0.849	0.577	0.890	0.650	0.826	0.809	0.766	0.829	0.607	0.805	0.813	0.869	0.656	0.339	0.764	0.795	0.804	0.534	0.851	0.348	0.568	0.528	0.566	0.914	0.859	0.414	0.891	0.487	0.546	0.322	0.567	0.378	0.301	0.876	0.478	0.429	0.899	0.643	0.346	0.323	0.338	0.794
Max Thickness Location	75th percentile	0.960	0.919	0.692	0.932	0.917	0.786	0.946	0.907	0.915	0.924	0.891	0.935	0.723	0.899	0.918	0.910	0.918	0.546	0.915	0.919	0.921	0.730	0.922	0.629	0.839	0.759	0.738	0.958	0.931	0.759	0.932	0.860	0.851	0.378	0.791	0.801	0.439	0.942	0.894	0.612	0.937	0.892	0.555	0.540	0.498	0.919
Trailing Edge Angle	Min	0.163	0.468	0.231	0.433	0.312	0.362	0.188	0.000	0.129	0.596	0.275	0.003	0.713	0.011	0.001	0.440	0.042	0.015	0.526	0.242	0.131	0.161	0.268	0.126	0.446	0.216	0.126	0.110	0.128	0.278	0.176	0.157	0.414	0.423	0.133	0.417	0.008	0.520	0.012	0.126	0.166	0.041	0.450	0.010	0.138	0.373
Trailing Edge Angle	Max	0.668	0.995	1.000	0.977	0.916	0.974	0.693	0.535	0.488	0.999	0.875	0.547	0.935	0.390	0.836	0.842	0.818	0.536	0.965	0.908	0.540	0.853	0.849	0.842	0.971	0.920	0.495	0.902	0.725	0.986	0.590	0.928	0.998	0.591	0.806	0.994	0.425	0.808	0.939	0.582	0.926	0.689	1.000	0.898	0.913	0.860
Trailing Edge Angle	Std.dev	0.132	0.093	0.178	0.067	0.142	0.116	0.133	0.058	0.094	0.082	0.123	0.100	0.055	0.087	0.194	0.115	0.119	0.116	0.115	0.131	0.084	0.155	0.133	0.187	0.114	0.141	0.065	0.214	0.118	0.114	0.084	0.192	0.103	0.064	0.159	0.084	0.076	0.068	0.223	0.093	0.146	0.135	0.071	0.185	0.176	0.049
Trailing Edge Angle	Median	0.353	0.851	0.811	0.878	0.678	0.819	0.346	0.091	0.296	0.933	0.710	0.153	0.878	0.121	0.491	0.649	0.589	0.180	0.884	0.630	0.223	0.571	0.570	0.402	0.835	0.715	0.222	0.733	0.329	0.895	0.287	0.571	0.872	0.473	0.479	0.881	0.105	0.680	0.476	0.293	0.751	0.339	0.932	0.514	0.568	0.771
Trailing Edge Angle	25th percentile	0.276	0.777	0.666	0.830	0.567	0.721	0.249	0.058	0.224	0.864	0.637	0.111	0.821	0.067	0.308	0.569	0.509	0.111	0.715	0.533	0.179	0.447	0.486	0.275	0.709	0.606	0.175	0.557	0.245	0.830	0.251	0.399	0.792	0.438	0.343	0.830	0.070	0.634	0.310	0.228	0.603	0.242	0.887	0.363	0.408	0.742
Trailing Edge Angle	75th percentile	0.489	0.907	0.977	0.911	0.799	0.878	0.435	0.117	0.363	0.966	0.773	0.229	0.897	0.211	0.616	0.789	0.657	0.283	0.918	0.733	0.302	0.689	0.716	0.574	0.895	0.781	0.273	0.795	0.423	0.933	0.343	0.734	0.933	0.515	0.597	0.922	0.125	0.727	0.706	0.364	0.821	0.441	0.961	0.619	0.691	0.799
Leading Edge Angle	Min	0.016	0.000	0.000	0.002	0.000	0.000	0.002	0.092	0.131	0.000	0.000	0.127	0.001	0.122	0.027	0.092	0.001	0.128	0.000	0.000	0.136	0.029	0.031	0.032	0.024	0.000	0.001	0.002	0.000	0.000	0.003	0.000	0.000	0.159	0.000	0.002	0.165	0.005	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.015
Leading Edge Angle	Max	0.725	0.245	0.245	0.262	0.245	0.245	0.243	0.979	0.244	0.246	0.202	0.245	0.240	0.244	0.333	0.233	0.944	0.315	0.301	0.245	0.962	0.244	0.259	0.906	0.271	0.249	0.953	1.000	0.947	0.857	0.236	0.244	0.246	0.239	0.245	0.246	0.953	0.243	0.987	0.916	0.245	1.000	0.288	0.996	0.239	0.274
Leading Edge Angle	Std.dev	0.099	0.071	0.070	0.053	0.063	0.060	0.055	0.243	0.026	0.066	0.039	0.030	0.090	0.019	0.057	0.036	0.213	0.031	0.071	0.063	0.236	0.048	0.052	0.089	0.047	0.068	0.224	0.305	0.171	0.069	0.060	0.056	0.058	0.031	0.068	0.052	0.194	0.063	0.221	0.188	0.062	0.341	0.068	0.272	0.047	0.055
Leading Edge Angle	Median	0.208	0.172	0.135	0.170	0.106	0.147	0.051	0.239	0.211	0.084	0.040	0.186	0.066	0.229	0.172	0.162	0.253	0.199	0.159	0.087	0.766	0.138	0.211	0.210	0.189	0.100	0.199	0.157	0.163	0.152	0.056	0.073	0.065	0.199	0.110	0.179	0.238	0.101	0.184	0.170	0.089	0.774	0.131	0.228	0.060	0.117
Leading Edge Angle	25th percentile	0.155	0.097	0.078	0.128	0.054	0.097	0.026	0.209	0.186	0.041	0.018	0.162	0.026	0.218	0.118	0.140	0.168	0.174	0.090	0.043	0.552	0.107	0.164	0.183	0.152	0.049	0.146	0.079	0.092	0.101	0.026	0.036	0.030	0.185	0.051	0.139	0.215	0.054	0.139	0.114	0.037	0.258	0.075	0.161	0.029	0.082
Leading Edge Angle	75th percentile	0.234	0.216	0.197	0.208	0.157	0.189	0.080	0.508	0.223	0.144	0.070	0.209	0.210	0.238	0.216	0.195	0.442	0.223	0.211	0.144	0.868	0.170	0.232	0.229	0.218	0.163	0.270	0.232	0.227	0.198	0.121	0.121	0.114	0.226	0.167	0.213	0.373	0.156	0.228	0.227	0.140	0.971	0.186	0.432	0.101	0.165
Camber Angle	Min	0.036	0.269	0.063	0.274	0.075	0.181	0.000	0.000	0.014	0.275	0.031	0.000	0.385	0.001	0.000	0.286	0.109	0.000	0.303	0.004	0.022	0.021	0.101	0.017	0.224	0.008	0.001	0.197	0.000	0.164	0.002	0.000	0.137	0.260	0.001	0.156	0.002	0.223	0.000	0.001	0.016	0.008	0.196	0.003	0.001	0.291
Camber Angle	Max	0.648	0.769	0.769	0.757	0.656	0.756	0.472	0.997	0.372	0.773	0.574	0.422	0.675	0.291	0.639	0.614	0.985	0.412	0.739	0.675	0.967	0.648	0.656	0.831	0.749	0.661	0.918	0.839	0.930	0.860	0.453	0.670	0.768	0.457	0.599	0.773	0.916	0.579	0.940	0.827	0.646	1.000	0.774	0.991	0.618	0.650
Camber Angle	Std.dev	0.115	0.091	0.151	0.064	0.126	0.103	0.124	0.222	0.087	0.084	0.100	0.088	0.057	0.070	0.164	0.090	0.148	0.099	0.102	0.122	0.229	0.126	0.130	0.165	0.108	0.123	0.194	0.131	0.159	0.095	0.097	0.166	0.096	0.072	0.146	0.088	0.169	0.076	0.190	0.169	0.117	0.245	0.080	0.201	0.139	0.061
Camber Angle	Median	0.258	0.580	0.560	0.612	0.417	0.547	0.106	0.066	0.200	0.585	0.398	0.068	0.540	0.076	0.321	0.432	0.482	0.097	0.575	0.365	0.614	0.359	0.406	0.290	0.596	0.442	0.137	0.496	0.196	0.603	0.075	0.310	0.531	0.332	0.260	0.627	0.090	0.416	0.395	0.175	0.446	0.676	0.620	0.459	0.297	0.498
Camber Angle	25th percentile	0.167	0.518	0.423	0.571	0.321	0.467	0.033	0.029	0.130	0.532	0.334	0.027	0.514	0.040	0.150	0.385	0.408	0.034	0.489	0.279	0.415	0.259	0.321	0.173	0.496	0.355	0.081	0.435	0.113	0.546	0.036	0.165	0.470	0.299	0.139	0.566	0.048	0.359	0.238	0.095	0.350	0.359	0.564	0.324	0.174	0.455
Camber Angle	75th percentile	0.335	0.646	0.647	0.648	0.514	0.606	0.194	0.278	0.262	0.639	0.442	0.140	0.585	0.157	0.424	0.531	0.571	0.182	0.655	0.463	0.720	0.449	0.537	0.424	0.656	0.502	0.239	0.583	0.302	0.653	0.118	0.446	0.584	0.382	0.370	0.677	0.216	0.475	0.532	0.275	0.514	0.804	0.675	0.593	0.393	0.541
Top Curvature	Min	0.016	0.008	0.007	0.009	0.016	0.011	0.016	0.007	0.003	0.055	0.007	0.012	0.010	0.039	0.001	0.019	0.012	0.006	0.082	0.020	0.030	0.003	0.050	0.020	0.046	0.002	0.009	0.065	0.006	0.013	0.035	0.003	0.008	0.020	0.003	0.057	0.010	0.014	0.000	0.006	0.004	0.011	0.081	0.003	0.003	0.072
Top Curvature	Max	0.147	0.509	0.635	0.505	0.384	0.134	0.095	0.168	0.067	0.669	0.389	0.117	0.205	0.113	0.504	0.054	0.248	0.094	0.527	0.548	0.168	0.082	0.322	0.143	0.545	0.292	0.184	0.637	0.198	0.599	0.110	0.583	0.643	0.049	0.133	0.676	0.130	0.181	0.647	0.118	0.322	0.752	0.683	1.000	0.558	0.293
Top Curvature	Std.dev	0.032	0.058	0.209	0.077	0.075	0.021	0.017	0.026	0.014	0.145	0.084	0.020	0.041	0.012	0.044	0.010	0.037	0.016	0.119	0.088	0.029	0.014	0.054	0.023	0.120	0.041	0.029	0.071	0.029	0.133	0.016	0.135	0.122	0.011	0.023	0.139	0.027	0.034	0.130	0.021	0.050	0.041	0.131	0.074	0.097	0.032
Top Curvature	Median	0.050	0.064	0.101	0.251	0.141	0.064	0.055	0.074	0.042	0.452	0.151	0.053	0.053	0.063	0.037	0.041	0.137	0.044	0.323	0.121	0.096	0.034	0.111	0.078	0.223	0.065	0.079	0.177	0.074	0.308	0.064	0.097	0.268	0.035	0.061	0.447	0.063	0.104	0.072	0.047	0.069	0.107	0.508	0.093	0.079	0.197
Top Curvature	25th percentile	0.038	0.047	0.068	0.200	0.098	0.050	0.042	0.058	0.030	0.292	0.073	0.038	0.032	0.058	0.025	0.030	0.111	0.033	0.170	0.085	0.073	0.025	0.088	0.062	0.148	0.042	0.060	0.146	0.055	0.199	0.057	0.049	0.193	0.028	0.046	0.327	0.046	0.079	0.046	0.034	0.048	0.084	0.398	0.059	0.038	0.178
Top Curvature	75th percentile	0.078	0.080	0.481	0.305	0.197	0.078	0.069	0.094	0.049	0.529	0.210	0.069	0.077	0.070	0.051	0.046	0.161	0.054	0.392	0.174	0.119	0.046	0.153	0.094	0.345	0.093	0.099	0.217	0.094	0.411	0.071	0.242	0.382	0.045	0.079	0.547	0.089	0.125	0.185	0.064	0.110	0.134	0.568	0.131	0.150	0.219
Bottom Curvature	Min	0.003	0.023	0.004	0.009	0.014	0.002	0.008	0.002	0.003	0.001	0.006	0.008	0.025	0.029	0.002	0.027	0.028	0.001	0.004	0.004	0.009	0.002	0.006	0.004	0.005	0.006	0.003	0.014	0.001	0.008	0.023	0.000	0.020	0.010	0.005	0.002	0.002	0.022	0.000	0.002	0.015	0.003	0.000	0.004	0.001	0.023
Bottom Curvature	Max	0.075	0.101	0.089	0.117	0.078	0.089	0.045	0.091	0.053	0.159	0.061	0.075	0.082	0.064	0.063	0.060	1.000	0.094	0.217	0.071	0.078	0.062	0.129	0.154	0.204	0.101	0.084	0.109	0.095	0.457	0.065	0.063	0.164	0.042	0.092	0.170	0.188	0.082	0.264	0.092	0.079	0.745	0.377	0.606	0.081	0.155
Bottom Curvature	Std.dev	0.019	0.013	0.017	0.016	0.012	0.014	0.009	0.015	0.011	0.016	0.011	0.013	0.016	0.006	0.011	0.008	0.153	0.011	0.047	0.012	0.013	0.012	0.029	0.019	0.030	0.016	0.017	0.018	0.015	0.044	0.009	0.011	0.020	0.011	0.015	0.022	0.041	0.012	0.032	0.017	0.011	0.057	0.036	0.095	0.016	0.024
Bottom Curvature	Median	0.030	0.057	0.045	0.053	0.042	0.050	0.030	0.034	0.030	0.042	0.033	0.032	0.057	0.041	0.028	0.044	0.125	0.021	0.047	0.034	0.039	0.027	0.064	0.031	0.050	0.050	0.034	0.060	0.039	0.056	0.038	0.023	0.068	0.026	0.043	0.030	0.050	0.050	0.030	0.036	0.041	0.048	0.030	0.069	0.038	0.071
Bottom Curvature	25th percentile	0.017	0.049	0.031	0.044	0.034	0.041	0.024	0.024	0.020	0.032	0.024	0.023	0.041	0.036	0.020	0.040	0.087	0.014	0.028	0.025	0.029	0.019	0.038	0.023	0.038	0.039	0.023	0.046	0.027	0.042	0.033	0.014	0.056	0.020	0.034	0.019	0.022	0.042	0.018	0.024	0.035	0.033	0.021	0.042	0.026	0.058
Bottom Curvature	75th percentile	0.047	0.066	0.057	0.064	0.051	0.059	0.038	0.045	0.037	0.053	0.040	0.043	0.066	0.045	0.036	0.050	0.220	0.028	0.092	0.043	0.049	0.036	0.080	0.044	0.074	0.062	0.048	0.071	0.050	0.077	0.042	0.032	0.081	0.032	0.054	0.043	0.078	0.060	0.049	0.049	0.049	0.065	0.042	0.134	0.048	0.088
Bottom Upper Curvature	Min	0.000	0.247	0.057	0.030	0.020	0.171	0.000	0.000	0.000	0.051	0.000	0.000	0.292	0.000	0.000	0.198	0.000	0.000	0.035	0.024	0.000	0.000	0.005	0.000	0.036	0.038	0.000	0.000	0.000	0.028	0.000	0.000	0.042	0.120	0.000	0.037	0.000	0.120	0.000	0.000	0.000	0.000	0.059	0.000	0.000	0.023
Bottom Upper Curvature	Max	0.229	0.611	0.523	0.545	0.485	0.557	0.403	0.225	0.328	0.537	0.425	0.000	0.506	0.000	0.509	0.413	0.277	0.239	0.432	0.502	0.000	0.473	0.264	0.427	0.459	0.533	0.119	0.755	0.000	0.463	0.000	0.475	0.514	0.246	0.528	0.506	0.157	0.452	0.596	0.213	0.545	1.000	0.500	0.709	0.398	0.206
Bottom Upper Curvature	Std.dev	0.076	0.055	0.110	0.117	0.113	0.071	0.105	0.023	0.087	0.098	0.061	0.000	0.049	0.000	0.131	0.061	0.040	0.056	0.108	0.104	0.000	0.093	0.068	0.116	0.121	0.094	0.007	0.132	0.000	0.098	0.000	0.106	0.108	0.046	0.143	0.096	0.026	0.065	0.103	0.043	0.134	0.227	0.092	0.088	0.075	0.025
Bottom Upper Curvature	Median	0.047	0.439	0.369	0.365	0.171	0.385	0.000	0.000	0.109	0.396	0.201	0.000	0.421	0.000	0.184	0.290	0.057	0.027	0.260	0.129	0.000	0.204	0.054	0.180	0.280	0.226	0.000	0.221	0.000	0.303	0.000	0.126	0.283	0.149	0.202	0.350	0.000	0.263	0.070	0.047	0.349	0.102	0.380	0.072	0.119	0.134
Bottom Upper Curvature	25th percentile	0.000	0.401	0.244	0.249	0.120	0.331	0.000	0.000	0.020	0.292	0.160	0.000	0.372	0.000	0.027	0.253	0.037	0.000	0.099	0.087	0.000	0.142	0.036	0.082	0.112	0.156	0.000	0.173	0.000	0.223	0.000	0.063	0.186	0.127	0.000	0.275	0.000	0.215	0.032	0.023	0.289	0.044	0.305	0.048	0.081	0.121
Bottom Upper Curvature	75th percentile	0.112	0.470	0.422	0.422	0.293	0.432	0.116	0.000	0.172	0.432	0.241	0.000	0.441	0.000	0.279	0.351	0.083	0.088	0.310	0.233	0.000	0.282	0.111	0.263	0.352	0.299	0.000	0.277	0.000	0.357	0.000	0.236	0.384	0.162	0.296	0.409	0.017	0.310	0.176	0.079	0.405	0.241	0.417	0.139	0.176	0.151
Nose Radius	Min	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Nose Radius	Max	0.001	0.001	0.001	0.001	0.001	0.001	0.000	0.040	0.000	0.007	0.000	0.001	0.001	0.000	0.000	0.000	0.001	0.001	0.001	0.000	0.001	0.001	0.000	0.001	0.001	0.001	0.001	0.026	0.001	0.001	0.001	0.001	0.001	0.000	0.001	0.001	0.001	0.001	0.311	0.002	0.001	0.280	0.001	1.000	0.001	0.000
Nose Radius	Std.dev	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.008	0.000	0.000	0.014	0.000	0.037	0.000	0.000
Nose Radius	Median	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
Nose Radius	25th percentile	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Nose Radius	75th percentile	0.000	0.001	0.001	0.001	0.000	0.001	0.000	0.001	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.002	0.000	0.001	0.000	0.000
Area	Min	0.134	0.338	0.095	0.158	0.101	0.194	0.137	0.046	0.012	0.010	0.013	0.055	0.330	0.210	0.010	0.313	0.011	0.001	0.011	0.009	0.101	0.015	0.069	0.039	0.090	0.182	0.126	0.058	0.082	0.015	0.297	0.000	0.033	0.012	0.182	0.001	0.012	0.576	0.005	0.041	0.363	0.001	0.000	0.004	0.006	0.011
Area	Max	0.679	0.995	0.975	0.992	0.720	0.968	0.588	0.986	0.496	0.837	0.538	0.989	0.796	0.582	0.588	0.646	1.000	0.391	0.986	0.503	0.990	0.590	0.516	0.783	0.792	0.913	0.810	0.993	0.990	0.956	0.782	0.362	0.993	0.048	0.921	0.515	0.651	0.971	0.566	0.623	0.986	0.978	0.717	0.820	0.561	0.336
Area	Std.dev	0.147	0.113	0.212	0.187	0.115	0.148	0.138	0.210	0.110	0.209	0.136	0.235	0.121	0.085	0.138	0.109	0.253	0.076	0.161	0.101	0.165	0.142	0.094	0.168	0.122	0.141	0.151	0.232	0.170	0.172	0.142	0.073	0.215	0.015	0.146	0.089	0.146	0.086	0.101	0.126	0.129	0.206	0.105	0.179	0.127	0.070
Area	Median	0.366	0.759	0.461	0.651	0.437	0.644	0.445	0.508	0.145	0.358	0.165	0.504	0.639	0.355	0.309	0.527	0.541	0.084	0.198	0.230	0.739	0.294	0.228	0.302	0.407	0.489	0.557	0.832	0.716	0.370	0.561	0.094	0.508	0.027	0.601	0.114	0.164	0.797	0.102	0.332	0.586	0.520	0.124	0.300	0.146	0.112
Area	25th percentile	0.266	0.669	0.332	0.485	0.358	0.537	0.283	0.338	0.103	0.181	0.093	0.303	0.519	0.277	0.187	0.373	0.278	0.046	0.120	0.159	0.618	0.187	0.179	0.194	0.325	0.384	0.430	0.688	0.581	0.269	0.397	0.046	0.333	0.014	0.500	0.056	0.093	0.718	0.054	0.239	0.506	0.369	0.067	0.165	0.076	0.056
Area	75th percentile	0.501	0.828	0.685	0.790	0.516	0.736	0.532	0.673	0.230	0.544	0.295	0.679	0.692	0.404	0.405	0.576	0.723	0.141	0.292	0.304	0.824	0.422	0.300	0.442	0.497	0.595	0.649	0.902	0.822	0.502	0.654	0.155	0.694	0.037	0.687	0.188	0.299	0.852	0.174	0.439	0.695	0.680	0.206	0.459	0.239	0.163
Perimeter	Min	0.021	0.214	0.051	0.150	0.102	0.118	0.026	0.004	0.000	0.088	0.040	0.004	0.198	0.051	0.000	0.163	0.024	0.000	0.174	0.030	0.015	0.006	0.051	0.019	0.137	0.079	0.026	0.158	0.008	0.091	0.084	0.000	0.120	0.012	0.048	0.050	0.003	0.262	0.000	0.004	0.131	0.030	0.082	0.010	0.003	0.101
Perimeter	Max	0.352	0.920	1.000	0.856	0.451	0.816	0.241	0.639	0.168	0.940	0.327	0.513	0.615	0.238	0.289	0.280	0.757	0.182	0.852	0.324	0.640	0.302	0.351	0.388	0.807	0.593	0.559	0.805	0.577	0.924	0.356	0.257	0.990	0.041	0.521	0.630	0.333	0.598	0.342	0.321	0.691	0.859	0.795	0.741	0.291	0.502
Perimeter	Std.dev	0.088	0.121	0.278	0.136	0.057	0.109	0.067	0.122	0.039	0.154	0.047	0.129	0.116	0.041	0.061	0.034	0.145	0.031	0.117	0.047	0.112	0.071	0.053	0.078	0.119	0.102	0.098	0.122	0.112	0.142	0.079	0.047	0.178	0.012	0.093	0.099	0.070	0.069	0.058	0.068	0.097	0.153	0.131	0.127	0.059	0.075
Perimeter	Median	0.133	0.457	0.310	0.497	0.245	0.396	0.159	0.198	0.031	0.464	0.129	0.180	0.423	0.102	0.101	0.213	0.382	0.025	0.455	0.139	0.358	0.123	0.178	0.129	0.382	0.275	0.236	0.534	0.323	0.533	0.203	0.059	0.527	0.022	0.270	0.250	0.075	0.428	0.094	0.129	0.291	0.341	0.413	0.227	0.112	0.345
Perimeter	25th percentile	0.071	0.378	0.197	0.393	0.209	0.322	0.077	0.108	0.018	0.348	0.099	0.078	0.322	0.071	0.051	0.199	0.262	0.014	0.350	0.103	0.278	0.067	0.132	0.091	0.295	0.207	0.154	0.454	0.231	0.423	0.127	0.034	0.388	0.015	0.205	0.200	0.041	0.375	0.050	0.079	0.238	0.229	0.319	0.130	0.071	0.288
Perimeter	75th percentile	0.225	0.542	0.737	0.606	0.287	0.465	0.201	0.303	0.057	0.586	0.163	0.289	0.503	0.129	0.149	0.251	0.484	0.044	0.512	0.171	0.416	0.184	0.204	0.201	0.472	0.354	0.308	0.591	0.399	0.619	0.268	0.100	0.662	0.032	0.332	0.322	0.140	0.481	0.136	0.191	0.362	0.453	0.515	0.325	0.155	0.399
Y Centroid	Min	0.084	0.056	0.017	0.115	0.070	0.038	0.036	0.043	0.049	0.059	0.038	0.046	0.058	0.045	0.036	0.081	0.164	0.047	0.234	0.036	0.068	0.041	0.145	0.013	0.139	0.016	0.030	0.189	0.032	0.075	0.036	0.002	0.033	0.061	0.019	0.038	0.059	0.109	0.008	0.017	0.041	0.034	0.017	0.007	0.000	0.402
Y Centroid	Max	0.438	0.287	0.203	0.855	0.543	0.280	0.297	0.919	0.149	0.727	0.192	0.099	0.279	0.079	0.300	0.180	1.000	0.164	0.881	0.499	0.846	0.189	0.707	0.616	0.864	0.517	0.860	0.866	0.790	0.852	0.077	0.311	0.805	0.102	0.323	0.567	0.599	0.317	0.698	0.486	0.518	0.981	0.796	0.976	0.315	0.811
Y Centroid	Std.dev	0.078	0.034	0.045	0.154	0.101	0.039	0.049	0.119	0.022	0.102	0.034	0.008	0.041	0.007	0.031	0.023	0.161	0.018	0.151	0.091	0.165	0.025	0.103	0.054	0.176	0.075	0.126	0.137	0.127	0.117	0.009	0.033	0.149	0.016	0.036	0.069	0.062	0.043	0.104	0.085	0.059	0.206	0.105	0.255	0.065	0.084
Y Centroid	Median	0.239	0.122	0.101	0.251	0.192	0.105	0.046	0.087	0.070	0.186	0.094	0.057	0.114	0.062	0.078	0.115	0.620	0.075	0.451	0.146	0.329	0.078	0.416	0.079	0.279	0.136	0.102	0.390	0.120	0.310	0.048	0.072	0.265	0.083	0.065	0.143	0.102	0.190	0.194	0.126	0.101	0.477	0.144	0.241	0.053	0.608
Y Centroid	25th percentile	0.193	0.103	0.067	0.190	0.137	0.084	0.044	0.067	0.063	0.147	0.069	0.052	0.088	0.058	0.062	0.105	0.502	0.063	0.362	0.105	0.232	0.063	0.320	0.068	0.221	0.096	0.073	0.304	0.077	0.252	0.042	0.054	0.186	0.073	0.048	0.118	0.083	0.161	0.134	0.082	0.077	0.336	0.108	0.126	0.036	0.559
Y Centroid	75th percentile	0.302	0.145	0.144	0.391	0.287	0.131	0.055	0.140	0.087	0.252	0.121	0.063	0.127	0.065	0.097	0.130	0.744	0.086	0.585	0.232	0.458	0.095	0.479	0.090	0.472	0.195	0.174	0.502	0.217	0.385	0.053	0.092	0.403	0.095	0.085	0.182	0.136	0.221	0.276	0.191	0.133	0.661	0.191	0.555	0.108	0.665
X Centroid	Min	0.766	0.637	0.111	0.735	0.682	0.249	0.754	0.297	0.687	0.297	0.509	0.542	0.656	0.717	0.545	0.886	0.018	0.002	0.385	0.474	0.596	0.322	0.680	0.040	0.239	0.222	0.258	0.708	0.684	0.033	0.789	0.007	0.013	0.111	0.339	0.000	0.029	0.763	0.007	0.031	0.783	0.043	0.002	0.003	0.008	0.675
X Centroid	Max	0.991	0.999	0.963	0.999	0.999	0.919	0.996	0.997	0.998	1.000	0.998	0.997	0.978	0.998	0.999	0.988	0.997	0.895	0.995	0.999	0.990	0.890	0.998	0.914	0.983	0.930	0.887	0.996	1.000	0.959	0.994	0.998	0.997	0.366	0.933	0.956	0.818	0.968	1.000	0.921	0.998	0.998	0.863	0.977	0.904	0.994
X Centroid	Std.dev	0.040	0.066	0.226	0.044	0.063	0.128	0.042	0.149	0.070	0.101	0.092	0.084	0.076	0.064	0.074	0.027	0.186	0.224	0.097	0.086	0.078	0.112	0.066	0.231	0.154	0.137	0.097	0.042	0.055	0.231	0.036	0.225	0.189	0.091	0.116	0.277	0.198	0.041	0.253	0.167	0.030	0.188	0.208	0.217	0.202	0.074
X Centroid	Median	0.938	0.927	0.534	0.945	0.926	0.754	0.922	0.849	0.918	0.918	0.871	0.914	0.758	0.897	0.903	0.924	0.871	0.404	0.899	0.901	0.904	0.692	0.930	0.417	0.777	0.704	0.724	0.950	0.926	0.634	0.951	0.734	0.776	0.219	0.749	0.747	0.296	0.910	0.807	0.554	0.948	0.849	0.452	0.361	0.377	0.908
X Centroid	25th percentile	0.916	0.884	0.368	0.913	0.879	0.643	0.901	0.724	0.859	0.845	0.809	0.854	0.728	0.845	0.847	0.908	0.748	0.236	0.805	0.832	0.842	0.613	0.881	0.268	0.638	0.587	0.641	0.933	0.882	0.410	0.928	0.545	0.611	0.180	0.656	0.335	0.161	0.885	0.526	0.451	0.926	0.723	0.262	0.219	0.241	0.831
X Centroid	75th percentile	0.956	0.955	0.750	0.966	0.956	0.825	0.946	0.929	0.954	0.955	0.922	0.952	0.780	0.946	0.945	0.939	0.934	0.612	0.952	0.946	0.942	0.774	0.960	0.659	0.872	0.802	0.784	0.966	0.954	0.809	0.965	0.888	0.880	0.283	0.829	0.843	0.429	0.934	0.925	0.684	0.967	0.923	0.618	0.582	0.503	0.944
Bending Inertia Y	Min	0.002	0.035	0.001	0.019	0.017	0.009	0.002	0.000	0.000	0.000	0.001	0.000	0.031	0.009	0.000	0.026	0.001	0.000	0.002	0.000	0.001	0.000	0.003	0.000	0.008	0.008	0.002	0.014	0.001	0.005	0.022	0.000	0.004	0.000	0.006	0.000	0.000	0.140	0.000	0.000	0.043	0.000	0.000	0.000	0.000	0.002
Bending Inertia Y	Max	0.246	0.703	0.784	0.820	0.310	0.623	0.143	0.755	0.095	0.523	0.129	0.644	0.441	0.166	0.137	0.188	0.853	0.054	0.795	0.105	0.742	0.167	0.140	0.395	0.502	0.593	0.491	0.903	0.699	0.826	0.334	0.041	1.000	0.000	0.560	0.115	0.243	0.688	0.163	0.210	0.704	0.773	0.317	0.473	0.159	0.133
Bending Inertia Y	Std.dev	0.064	0.138	0.195	0.186	0.055	0.129	0.046	0.145	0.019	0.105	0.028	0.154	0.119	0.031	0.030	0.055	0.193	0.007	0.102	0.018	0.160	0.039	0.023	0.071	0.079	0.102	0.104	0.226	0.159	0.129	0.092	0.005	0.188	0.000	0.110	0.012	0.042	0.118	0.016	0.043	0.125	0.157	0.022	0.081	0.027	0.020
Bending Inertia Y	Median	0.040	0.350	0.103	0.254	0.081	0.230	0.068	0.105	0.003	0.075	0.010	0.093	0.221	0.037	0.023	0.113	0.169	0.001	0.041	0.017	0.321	0.022	0.019	0.026	0.089	0.110	0.143	0.480	0.274	0.097	0.131	0.002	0.190	0.000	0.173	0.006	0.006	0.386	0.003	0.034	0.159	0.143	0.012	0.038	0.007	0.024
Bending Inertia Y	25th percentile	0.015	0.239	0.038	0.132	0.051	0.137	0.019	0.033	0.001	0.027	0.004	0.021	0.118	0.018	0.006	0.045	0.041	0.000	0.025	0.009	0.189	0.006	0.013	0.008	0.052	0.056	0.067	0.270	0.146	0.053	0.049	0.001	0.076	0.000	0.101	0.003	0.001	0.289	0.001	0.013	0.106	0.058	0.006	0.009	0.002	0.012
Bending Inertia Y	75th percentile	0.099	0.453	0.318	0.446	0.121	0.329	0.110	0.238	0.010	0.175	0.028	0.227	0.313	0.055	0.050	0.147	0.360	0.003	0.067	0.030	0.421	0.063	0.034	0.078	0.149	0.191	0.228	0.614	0.409	0.182	0.199	0.005	0.371	0.000	0.253	0.013	0.029	0.469	0.008	0.075	0.255	0.288	0.022	0.112	0.016	0.039
Bending Inertia X	Min	0.126	0.297	0.072	0.152	0.092	0.148	0.137	0.039	0.012	0.008	0.012	0.047	0.295	0.178	0.009	0.268	0.010	0.000	0.009	0.008	0.087	0.012	0.063	0.030	0.081	0.147	0.102	0.050	0.076	0.013	0.275	0.000	0.026	0.009	0.157	0.001	0.009	0.555	0.004	0.032	0.308	0.001	0.000	0.004	0.005	0.010
Bending Inertia X	Max	0.669	0.992	0.963	0.985	0.630	0.942	0.581	0.975	0.429	0.822	0.500	0.984	0.714	0.477	0.588	0.617	1.000	0.327	0.983	0.491	0.979	0.531	0.504	0.648	0.753	0.852	0.718	0.995	0.987	0.925	0.750	0.336	0.992	0.036	0.892	0.419	0.577	0.953	0.556	0.551	0.985	0.965	0.574	0.729	0.438	0.328
Bending Inertia X	Std.dev	0.145	0.114	0.206	0.178	0.102	0.144	0.132	0.196	0.097	0.191	0.120	0.223	0.100	0.070	0.128	0.102	0.235	0.061	0.155	0.093	0.161	0.125	0.086	0.141	0.106	0.130	0.129	0.229	0.162	0.164	0.131	0.065	0.201	0.011	0.141	0.077	0.120	0.085	0.093	0.110	0.127	0.190	0.087	0.149	0.102	0.065
Bending Inertia X	Median	0.344	0.672	0.372	0.595	0.392	0.543	0.424	0.442	0.129	0.321	0.144	0.457	0.556	0.307	0.279	0.460	0.483	0.072	0.172	0.206	0.656	0.248	0.208	0.243	0.345	0.408	0.467	0.780	0.656	0.308	0.510	0.080	0.432	0.021	0.523	0.096	0.130	0.758	0.087	0.275	0.541	0.458	0.099	0.239	0.113	0.095
Bending Inertia X	25th percentile	0.246	0.591	0.266	0.439	0.322	0.445	0.260	0.294	0.092	0.163	0.084	0.277	0.461	0.244	0.167	0.330	0.248	0.038	0.106	0.144	0.541	0.155	0.161	0.149	0.275	0.315	0.365	0.626	0.530	0.222	0.370	0.039	0.280	0.011	0.426	0.047	0.074	0.685	0.047	0.195	0.462	0.317	0.054	0.128	0.059	0.051
Bending Inertia X	75th percentile	0.496	0.745	0.564	0.723	0.466	0.639	0.495	0.595	0.203	0.490	0.255	0.627	0.593	0.350	0.367	0.517	0.651	0.113	0.264	0.274	0.751	0.362	0.267	0.367	0.420	0.505	0.549	0.857	0.753	0.435	0.594	0.134	0.604	0.029	0.604	0.158	0.239	0.810	0.151	0.365	0.643	0.603	0.168	0.373	0.189	0.148

In Figure C.2 selected parsimonious features are projected on a 3D axis and colour-coded by airfoil cluster. This is to visually emphasise the cluster separation by the selected parsimonious features. From this figure, it can be observed that the clearest separation comes between the following parsimonious feature values for the 50 clusters:

1. max top magnitude, trailing edge angle, and max camber location,
2. max top magnitude, max bottom upper magnitude, and max camber location, and
3. max top magnitude, top curvature, and trailing edge angle.

It is, however, important to note here that this is only a 3D visualisation of the selected parsimonious characteristics, colour-coded by cluster, and clearer separation is present in the higher dimensional space, i.e. for all 23 parsimonious variables.

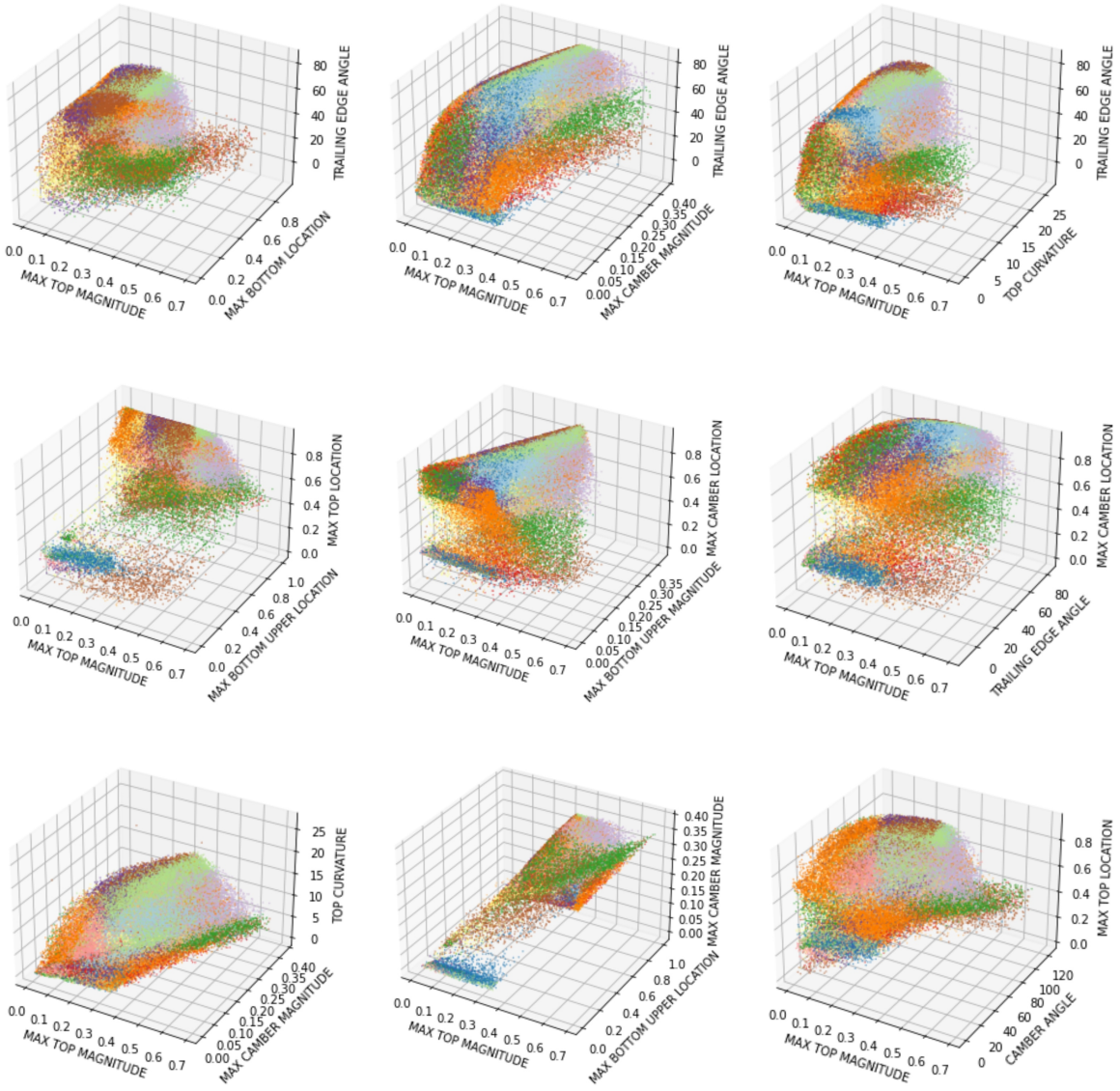


Figure C.2: Illustration of the distance of airfoil class separation for different selections of the parsimonious shape and structural characteristics

Appendix D

Defining a baseline design

D.1 Introduction

For the optimisation verification process, a baseline performance is required, and hence a baseline sailplane design. This baseline sailplane design includes the full definition of the cross-country task, the airfoil fuselage/tail configuration, the wing geometry, and the airfoil design of each station along the wing span.

This baseline performance acts as the reference point against which all optimisation results are measured. The baseline design is also used to establish the upper and lower bounds of the wing geometry and provide the baseline airfoil profiles, which are subjected to geometric alterations.

At the 2022 world gliding championships, the JS3 sailplane took the first six podium positions and therefore it can be argued that the current JS3 sailplane design, at the time of writing, has the best approximation to an optimal design for an 18m class sailplane. The baseline design is thus deemed to be similar to the JS3 18m class sailplane. This however does provide difficulty in optimising the designs in the verification process but ensures a strong baseline because well-rounded and proven cross-country performance characteristics are used. Due to intellectual property, the exact JS3 design cannot be used. This means the airfoils at each station and the wing geometry are chosen to be a near approximation of the current design.

D.2 Baseline design

To define a baseline design, at a minimum, a flight task, weather model, the airfoil profiles at each station, the wing geometry, and the fuselage/tail performance at various flight speeds are required.

For the airfoil design, the Althaus AH 80-129 (as seen in Figure D.1) is used at each station. This airfoil is chosen due to its similarity with some of the actual airfoils used along the JS3's span. The Althaus AH 80-129 airfoil has the following specification and

its profile can be seen in Figure D.1:

1. maximum thickness of 12.9% at 40.2% chord, and
2. maximum camber of 4.1% at 43.5% chord.

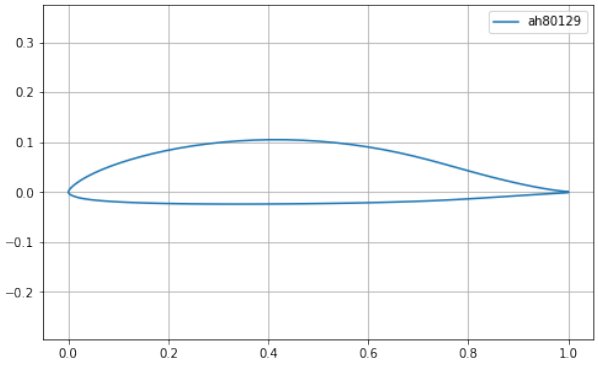


Figure D.1: Althaus AH 80-129 airfoil used at all five stations in the baseline sailplane design

For the wing geometry, the specification of Table D.1 is used and the resulting wing has a total full span of 17.98 metres, an area of 9.94 m^2, and an aspect ratio of 32.91.

Table D.1: JS3 wing geometry

Section (m)	Root (m)	Tip (m)	Span (m)	True span (m)
1	0.750	0.718	1.600	1.600
2	0.718	0.615	2.602	4.202
3	0.615	0.479	1.800	6.002
4	0.479	0.320	2.168	8.170
5	0.320	0.200	0.892	9.062

In Figure D.2, the drag performance (in Newtons) of the fuselage and the empennage design at various flight speeds is depicted. This performance curve was calculated with CFD simulations of the actual JS3 fuselage and tail design and provided by Jonker Sailplanes. It is important to note that the calculation of these additional drag terms is beyond the scope of this research and therefore the details of the calculations are excluded.

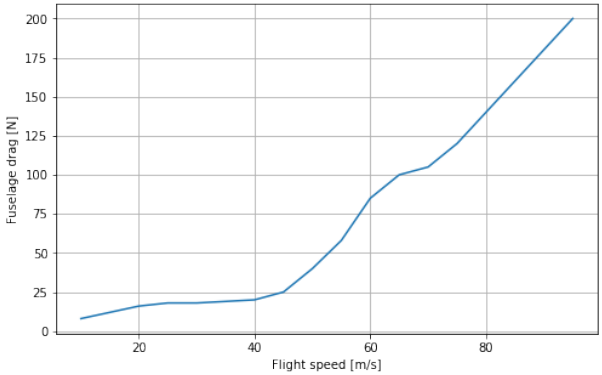


Figure D.2: Drag performance of the sailplane's fuselage and empennage at different flight speeds

D.3 Translating the baseline design

The baseline design specified up to this point is done in the format of the TSM and hence a translation needs to be done to go from the TSM format to the DSM format. In order to translate the baseline design into an applicable format for the DSM the only conversion that is required is the mapping of the Althaus AH 80-129 airfoil from the raw x and y coordinate domain to the parsimonious feature vector space. This is done by utilising the airfoil shape mapping module in an optimisation routine. More specifically, the airfoil fitting is done by setting up an optimisation problem where the objective is to minimise the sum of the MAE between the generated airfoil x and y coordinates, and the target airfoil coordinates. In each iteration, the 23 parsimonious variables are updated by means of a GA, in the direction that would most probably minimise the objective function. It was found that the GA is able to reach acceptable fitting accuracy when a population size of 100 with 20 iterations and a cross-over probability of 0.5 is used. After 20 iterations, an objective function value i.e. total MAE, of 6.69×10^{-6} is achieved. In Figure D.3 the fitted versus actual profiles are shown.

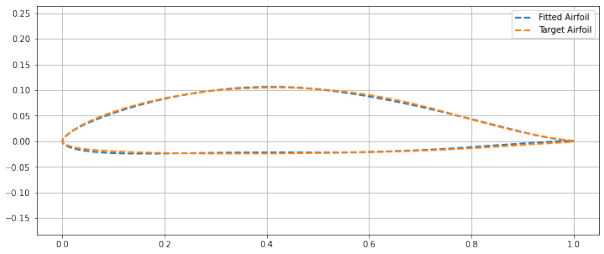


Figure D.3: Actual and fitted airfoil profiles resulting from the optimisation routine

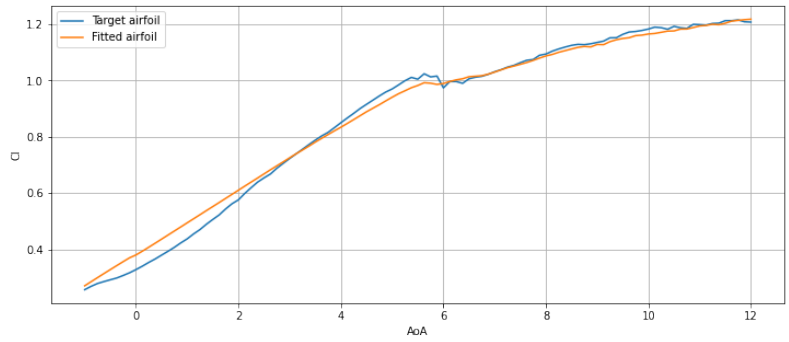
Although the total MAE is sufficiently low, from visual inspection of the actual vs fitted airfoils as shown in Figure D.3 slight deviations are still noticeable at the bottom surface leading edge and the bottom surface trailing edge. Ensuring that the fit is adequate comes in the form of analysing the aerodynamic performance of both the fitted and the actual airfoils. Here, both fitted and actual profiles are evaluated with XFOIL with the following simulation inputs:

1. N_{crit} set to 9.
2. Reynolds number of 2e6.
3. Mach number of 0.2.
4. AoA ranging from 0 to 12 in 0.125 increments.

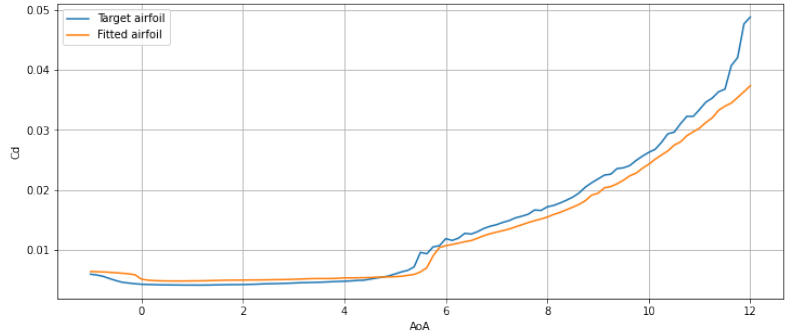
The results of the aerodynamic performance analysis can be seen in Figure D.4. For the analysis conducted the coefficients of determination are calculated by comparing the actual and predicted values for the C_l, C_d, and C_m coefficients. The respective R^2 values are as follows:

1. lift coefficient: 0.9925
2. drag coefficient: 0.9574
3. lift/drag coefficient: 0.9401

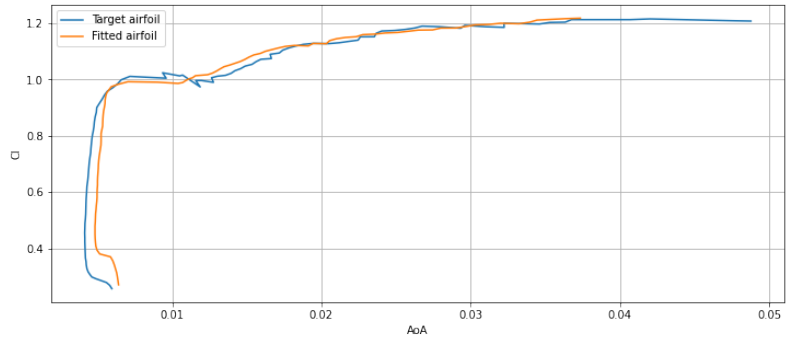
Both the high R^2 values between the actual and fitted airfoil performance and the strong correlation observed when analysing Figure D.4 suggest that the low-order parsimonious feature approximation of the Althaus AH 80-129 airfoil is acceptable for further use in the optimisation verification study.



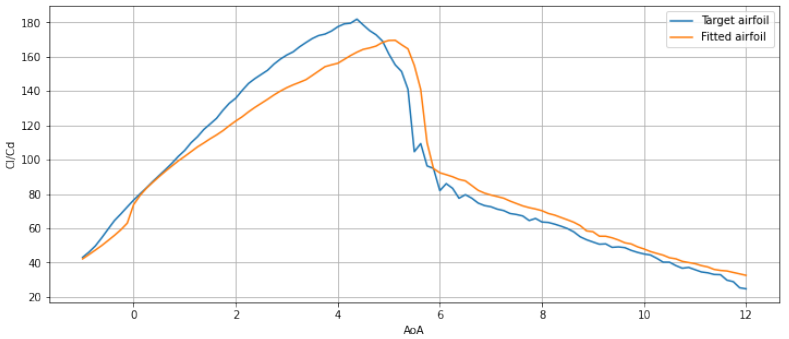
(a) C_l vs AoA



(b) C_d vs AoA



(c) C_l vs C_d



(d) C_l/C_d vs AoA

Figure D.4: Aerodynamic performance of the AH 80-129 airfoil as evaluated with XFOIL

D.4 Cross-country flight task definition

The final element that is required before the baseline design's cross-country performance can be calculated with the various models is a cross-country flight task and a weather model, i.e. the simulation constants. For the sailplane constants and dynamic station LLT constants respectively: (1) the sailplane is set to have a weight of 539 kg and fuselage/tail performance of Figure D.2, and (2) the wing is set to have a 0° twist angle and is divided into 200 subdivisions. For the cross-country simulation, the Quasts weather model is used with: A1 (0.00%), A2 (56.67%), B1 (3.33%), and B2 (40.00%), a 300km flight task is specified, and the thermal radii are defined to vary between 30 metres and 400 metres in increments of 10 metres.

D.5 Baseline simulation comparison

The final step in this section is to establish the performance of the baseline design with the DSM, the TSM, and the JSCode. Using the simulation constants and the cross-country flight task defined in section D.4 as well as the baseline sailplane design discussed in Sections D.3 and D.2, the baseline design's cross-country performance is estimated using the three models.

The objective here is twofold, firstly it is to establish the baseline performance of the defined sailplane for the specified flight task. Secondly, it is to establish if the translation of the sailplane design to an appropriate format for the DSM model is sufficient and hence yielded accurate performance estimations when compared to the ground truth i.e. the TSM model results.

The results from the TSM model simulation is an average cross-country speed of 97.32 km/h and for the DSM model simulation this value was 95.14 km/h - meaning there is a 2.28 MAPE between the TSM, and DSM. It is also noted that there is a strong correlation (Pearson correlation coefficient of 0.81), between the total drag predictions from the two models, for the same glider speed. Figure D.5 shows the total drag predictions at the same

flight speeds for the three respective models and from this plot, and preceding analysis, it is concluded that the translation of the sailplane design to the DSM framework is adequate and that the model predictions are indeed accurate.

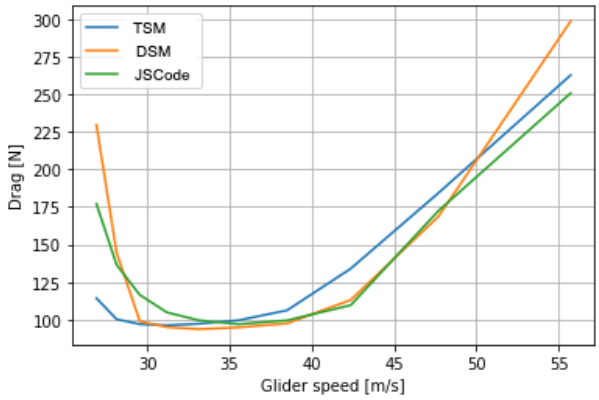


Figure D.5: Comparison of the total drag predictions made by the DSM, TSM, and JSCode models for the baseline sailplane design and flight task

Appendix E

Detailed results from PDLM model optimisation

In Table E.1 we provide the top 15 performing model architectures, sorted by validation MSE, from the refined model search pipeline as discussed in Section 6.4.2. From this table the following observations are made:

1. The top five performing models all have L1 and L2 regularisation situated at the lower constraint defined in Table 6.3.
2. The top five performing models all have three or more hidden layers with the models with three hidden layers consistently having higher learning rates and larger batch sizes than the rest of the models in the top five.
3. The best-performing model in this table outperforms the bottom 10 models by more than 10% on average – in terms of validation MSE.

Table E.1: Refined random search hyperparameters and model performance on training and validation datasets: 15 best performers

Batchsize	L1Reg.	L2Reg.	LR	No.layers	No.neurons	TrainingMSE	Valid.MSE
64	1.00×10^{-10}	1.00×10^{-10}	0.00001	7	2048	0.000728	0.003872
256	1.00×10^{-10}	1.00×10^{-10}	0.001	3	1000	0.000558	0.004059
128	1.00×10^{-10}	1.00×10^{-10}	0.001	8	128	0.001134	0.004097
256	1.00×10^{-9}	1.00×10^{-9}	0.00001	6	3000	0.000724	0.004130
128	1.00×10^{-9}	1.00×10^{-9}	0.001	3	900	0.000883	0.004181
128	1.00×10^{-8}	1.00×10^{-8}	0.001	3	1024	0.001105	0.004207
1024	1.00×10^{-10}	1.00×10^{-10}	0.01	3	1000	0.000612	0.004250
64	1.00×10^{-8}	1.00×10^{-8}	0.0001	6	200	0.001663	0.004340
256	1.00×10^{-9}	1.00×10^{-9}	0.01	3	3000	0.001380	0.004350
1024	1.00×10^{-10}	1.00×10^{-10}	0.01	8	600	0.000796	0.004353
512	1.00×10^{-10}	1.00×10^{-10}	0.01	5	700	0.000890	0.004394
128	1.00×10^{-9}	1.00×10^{-9}	0.001	10	128	0.001166	0.004467
64	1.00×10^{-10}	1.00×10^{-10}	0.0001	2	512	0.001429	0.004467
1024	1.00×10^{-9}	1.00×10^{-9}	0.01	5	100	0.000901	0.004537
512	1.00×10^{-10}	1.00×10^{-10}	0.01	2	512	0.001029	0.004591

In Table E.2 we provide the 15 median performing model architectures, sorted by validation MSE, from the refined model search pipeline as discussed in Section 6.4.2. From this table, it is observed that in general, these model architectures have larger batch sizes,

fewer neurons per hidden layer, and more hidden layers than the architectures shown in Table E.1.

Table E.2: Refined random search hyperparameters and model performance on training and validation datasets: 15 median performers

Batchsize	L1Reg.	L2Reg.	LR	No.layers	No.neurons	TrainingMSE	Valid.MSE
1024	1.00×10^{-4}	1.00×10^{-4}	0.0001	10	200	0.004648	0.008146
1024	1.00×10^{-6}	1.00×10^{-6}	0.01	10	64	0.004611	0.008209
256	1.00E-10	1.00E-10	0.1	2	64	0.008404	0.008366
2048	1.00×10^{-4}	1.00×10^{-4}	0.0001	2	400	0.006002	0.008405
1024	1.00×10^{-5}	1.00×10^{-5}	0.0001	9	200	0.003936	0.008417
2048	1.00×10^{-9}	1.00×10^{-9}	0.01	1	64	0.007850	0.008435
2048	1.00×10^{-6}	1.00×10^{-6}	0.01	10	64	0.003687	0.008521
512	1.00×10^{-4}	1.00×10^{-4}	0.00001	4	400	0.005178	0.008524
2048	1.00E-10	1.00E-10	0.00001	9	3500	0.000663	0.008528
64	1.00×10^{-4}	1.00×10^{-4}	0.0001	5	512	0.007649	0.008649
2048	1.00×10^{-4}	1.00×10^{-4}	0.0001	2	1024	0.006296	0.008798
256	1.00E-10	1.00E-10	0.1	10	256	0.005027	0.008903
128	1.00×10^{-8}	1.00×10^{-8}	0.01	8	2048	0.004261	0.008973
1024	1.00×10^{-4}	1.00×10^{-4}	0.0001	5	700	0.005896	0.009099
256	1.00×10^{-3}	1.00×10^{-3}	0.00001	1	200	0.009127	0.009101

In Table E.3 we provide the 15 worst performing model architectures, sorted by validation MSE, from the refined model search pipeline as discussed in Section 6.4.2. From this table, it is observed that in general, these model architectures have higher learning rates, higher levels of L1 and L2 regularisation, more neurons per hidden layer, and more or less the same number of hidden layers as the architectures shown in Table E.1.

Table E.3: Refined random search hyperparameters and model performance on training and validation datasets: 15 worst performers

Batchsize	L1Reg.	L2Reg.	LR	No.layers	No.neurons	TrainingMSE	Valid.MSE
256	1.00×10^{-4}	1.00×10^{-4}	0.1	5	2048	14.511664	17.945860
512	1.00×10^{-3}	1.00×10^{-3}	0.1	8	400	26.815718	12.989746
1024	1.00×10^{-3}	1.00×10^{-3}	0.1	6	512	9.951712	12.930624
128	1.00×10^{-3}	1.00×10^{-3}	0.001	7	4096	11.869102	11.844732
128	1.00×10^{-3}	1.00×10^{-3}	0.1	3	512	7.167179	6.530677
1024	1.00×10^{-3}	1.00×10^{-3}	0.01	7	1000	6.222282	5.969728
512	1.00×10^{-3}	1.00×10^{-3}	0.01	9	800	5.661497	5.032884
64	1.00×10^{-3}	1.00×10^{-3}	0.01	2	2048	4.618849	4.587899
64	1.00×10^{-3}	1.00×10^{-3}	0.01	9	512	2.370751	2.354449
512	1.00E-10	1.00E-10	0.1	9	3000	0.383596	2.196884
64	1.00×10^{-7}	1.00×10^{-7}	0.1	3	2048	1.343324	1.797792
128	1.00×10^{-5}	1.00×10^{-5}	0.01	10	4096	1.527870	1.644997
1024	1.00×10^{-3}	1.00×10^{-3}	0.1	3	256	1.482997	1.433730
512	1.00×10^{-6}	1.00×10^{-6}	0.1	2	3000	0.797668	1.139087
1024	1.00×10^{-4}	1.00×10^{-4}	0.01	9	1024	0.518459	0.639048

Appendix F

Supplementary airfoil shape mapping module analysis

In Figure F.1 we provide additional examples for the fitted versus the target airfoil profiles as discussed in Section 7.7.3.

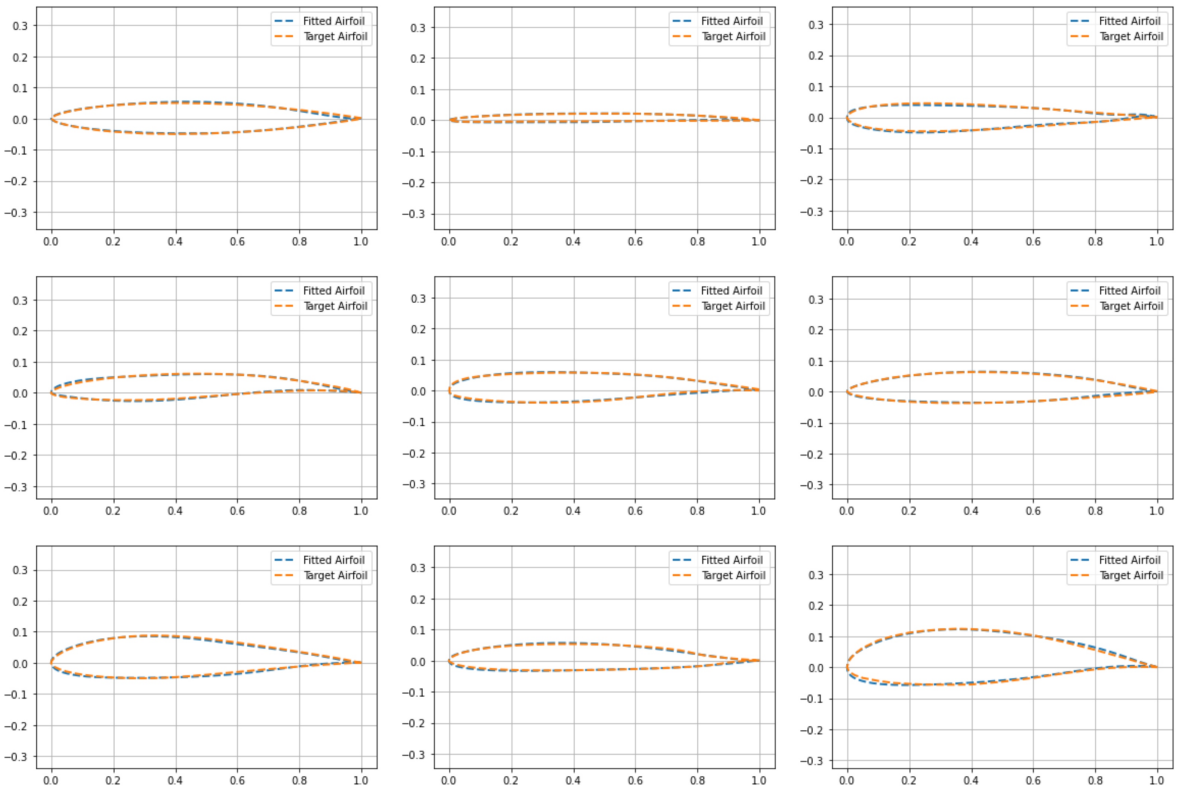


Figure F.1: Actual versus fitted airfoils sampled from 100 random UIUC airfoil fittings as done with the airfoil shape mapping module. The average fitting error in MSE displayed is 2.87e-8 versus the actual average fitting error over all 100 fittings of 2.08e-8.

In Figure F.2 we provide the error distribution calculated between the generated and target airfoil's 23 parsimonious variables as discussed in Section 7.7.3. Here, an average fitting error rate of 0.69% is observed across all 23 variables with the most accurate prediction being for the maximum bottom upper location with an average error of 0.48% and the most inaccurate for the perimeter with an average error of 1.10%.

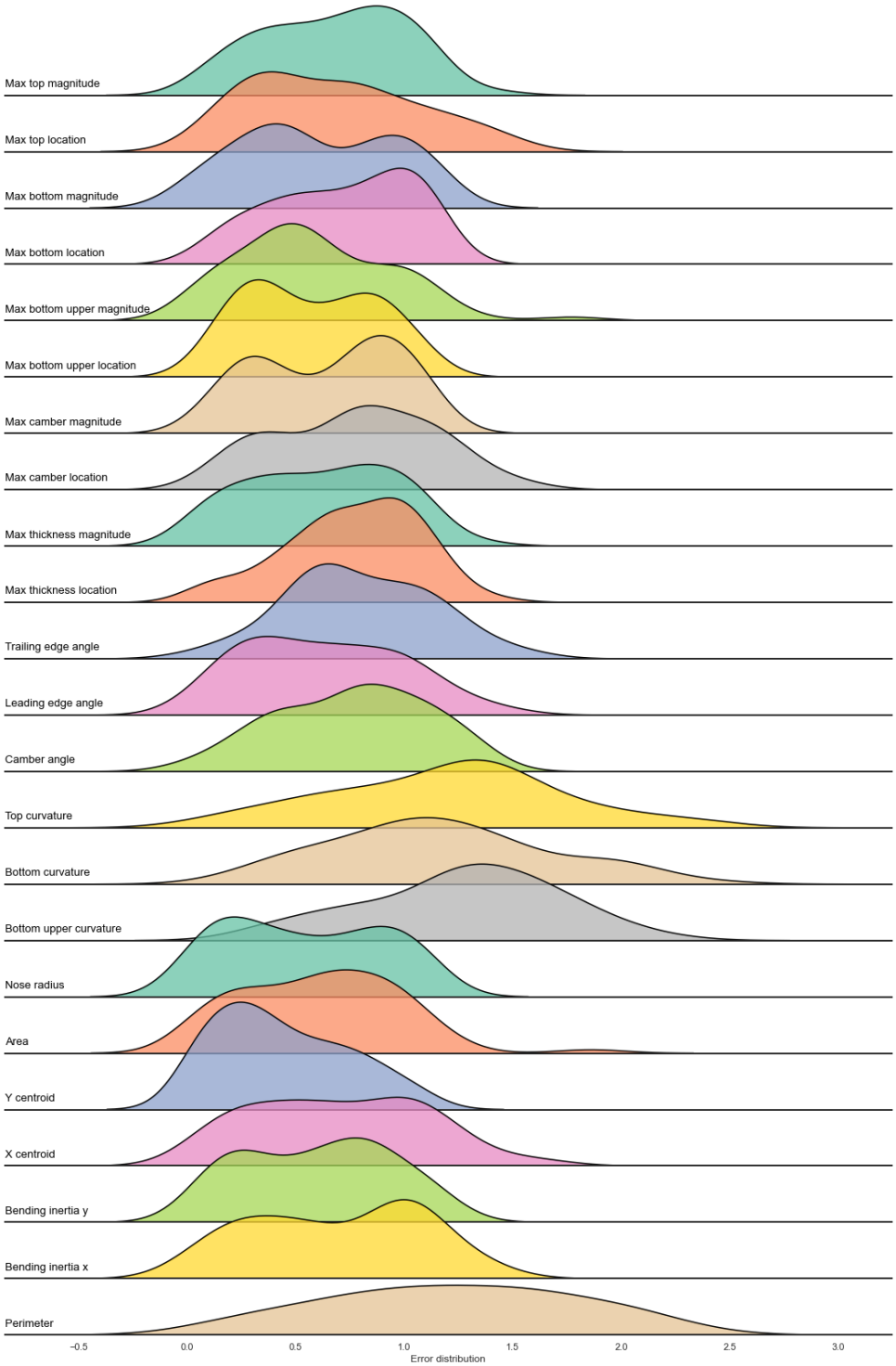
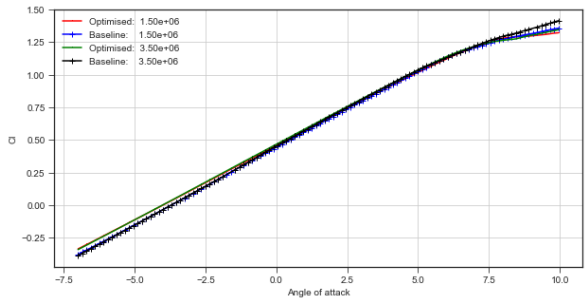


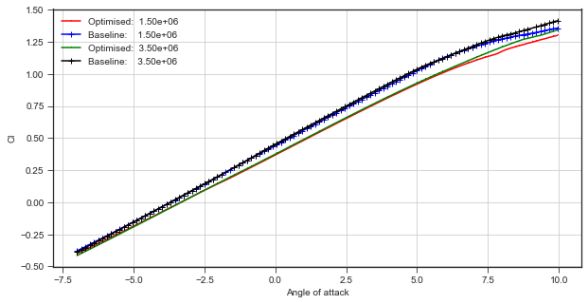
Figure F.2: Distribution of error as calculated between the 23 parsimonious variables of airfoil shape mapping module and the actual values

Appendix G

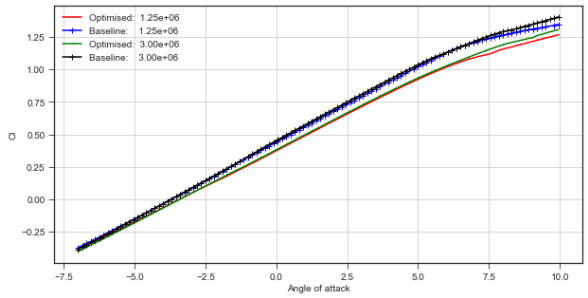
Supplementary analysis of JS4 optimised airfoils



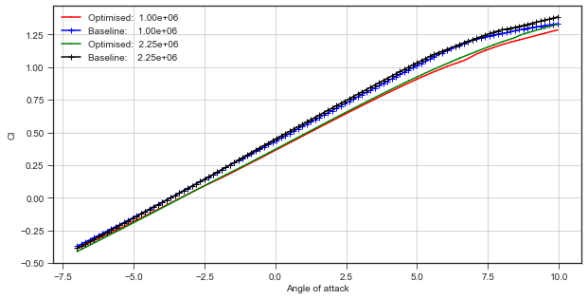
(a) Lift performance for the baseline and optimised airfoil at station 1



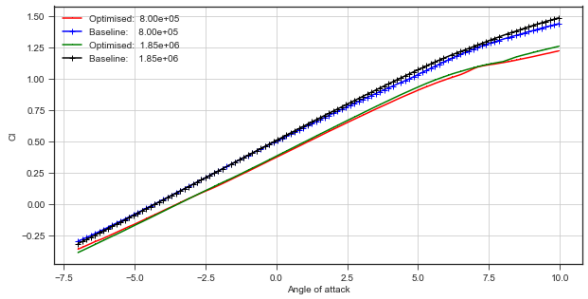
(b) Lift performance for the baseline and optimised airfoil at station 2



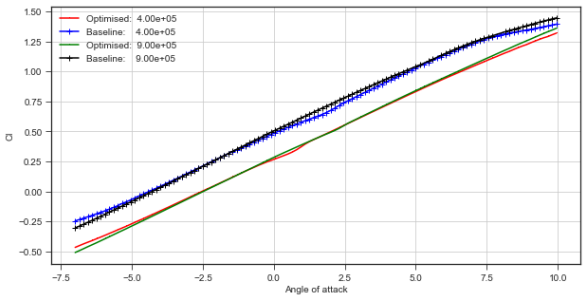
(c) Lift performance for the baseline and optimised airfoil at station 3



(d) Lift performance for the baseline and optimised airfoil at station 4

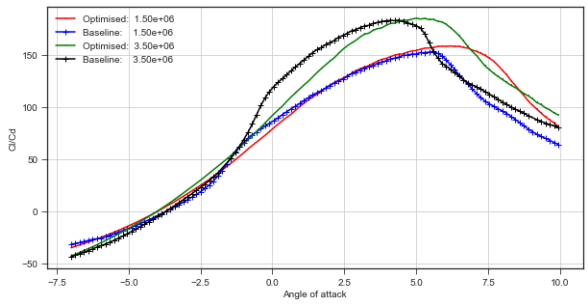


(e) Lift performance for the baseline and optimised airfoil at station 5

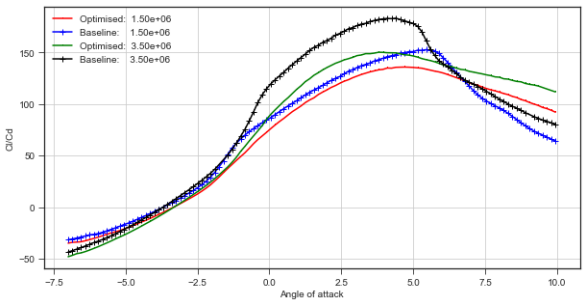


(f) Lift performance for the baseline and optimised airfoil at station 6

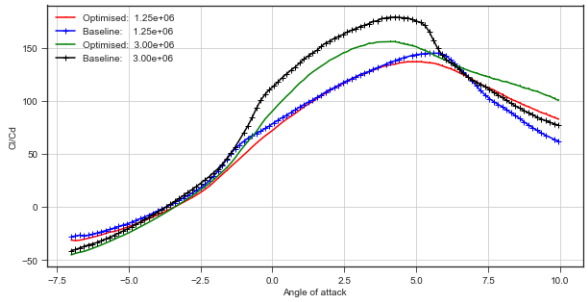
Figure G.1: Lift performance of the optimised and baseline airfoils simulated for each respective stations' applicable upper and lower Reynolds numbers



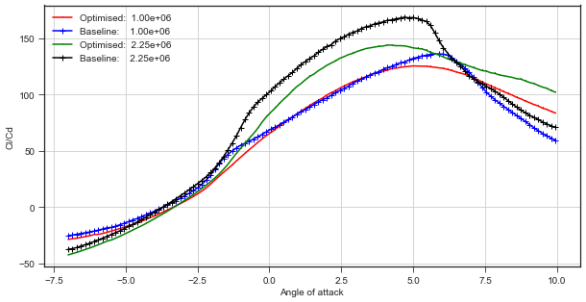
(a) Lift-to-drag performance for the baseline and optimised airfoil at station 1



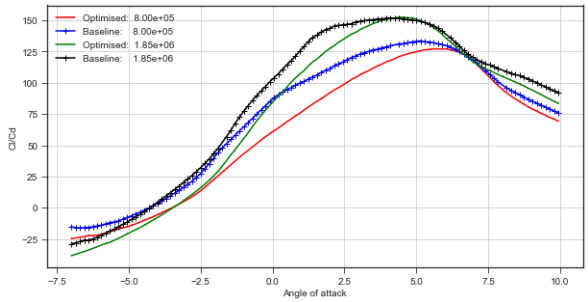
(b) Lift-to-drag performance for the baseline and optimised airfoil at station 2



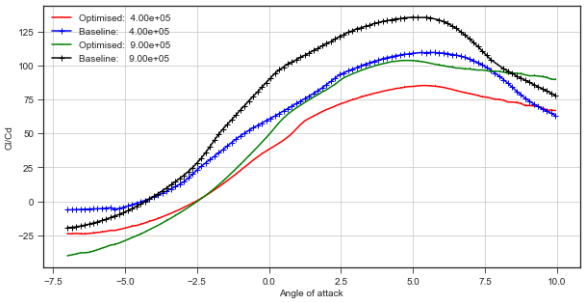
(c) Lift-to-drag performance for the baseline and optimised airfoil at station 3



(d) Lift-to-drag performance for the baseline and optimised airfoil at station 4

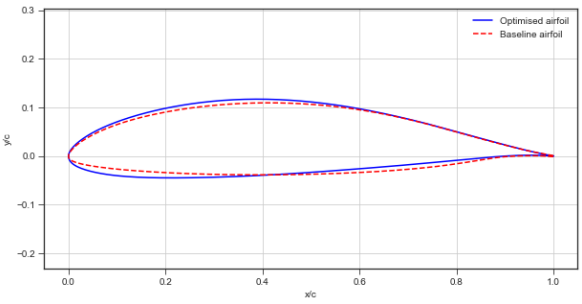


(e) Lift-to-drag performance for the baseline and optimised airfoil at station 5

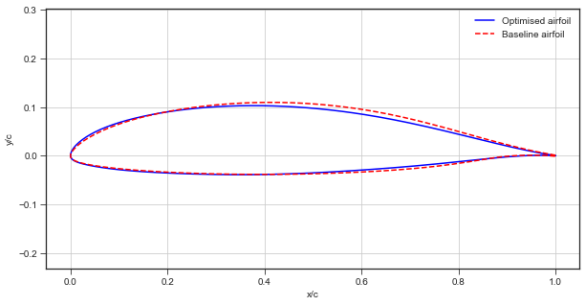


(f) Lift-to-drag performance for the baseline and optimised airfoil at station 6

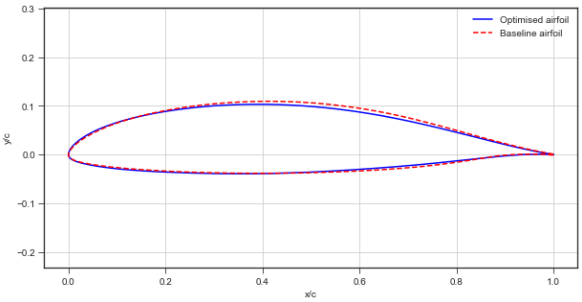
Figure G.2: Lift-to-drag performance of the optimised and baseline airfoils simulated for each respective stations' applicable upper and lower Reynolds numbers



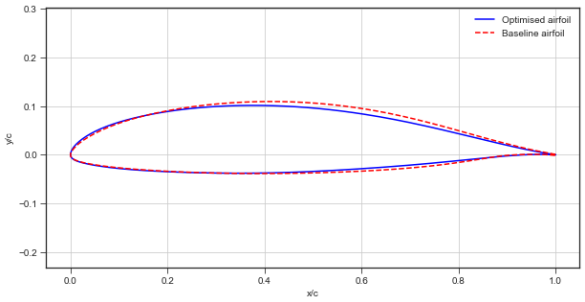
(a) Baseline and optimised airfoil at station 1



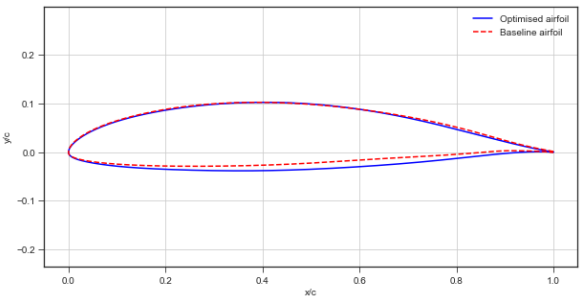
(b) Baseline and optimised airfoil at station 2



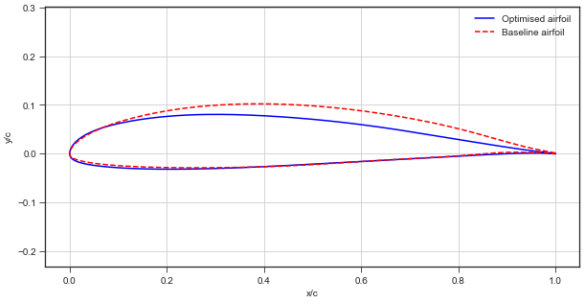
(c) Baseline and optimised airfoil at station 3



(d) Baseline and optimised airfoil at station 4



(e) Baseline and optimised airfoil at station 5



(f) Baseline and optimised airfoil at station 6

Figure G.3: Comparison of the optimised and baseline airfoil profiles present at each of the six stations

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Abutunis, A., Hussein, R., & Chandrashekhara, K. (2019). A neural network approach to enhance blade element momentum. *Renewable Energy*, 1281–1293.

Achour, G., Sung, W. J., Pinon-Fischer, O. J., & Mavris, D. N. (2020). Development of a conditional generative adversarial network for airfoil shape optimization. *Aiaa scitech 2020 forum*. https://doi.org/10.2514/6.2020-2261

AF, M., AS, B., & JR, C. (2004). Measurements and predictions of forces, pressures and cavitation on 2-d sections suitable for marine current turbines. *Proceedings of the Institution of Mechanical Engineers Part M. J Eng Maritime Environ*, *218*, 127–138.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu).

Andersson, K. (2009). Extending endurance for small uavs by predicting and searching for thermal updrafts. *Mechanical and Astronautical Engineering Dept., Naval Postgraduate School, Monterey, California*.

Andres, E., Salcedo-Sanz, S., & Monge, F. (2012). Efficient aerodynamic design through evolutionary programming and support vector regression algorithms. *Expert Syst. Appl.*, 10700–10708.

Andrés-Pérez, E., & Paulete-Periáñez, C. (2021). On the application of surrogate regression models for aerodynamic coefficient prediction. *Complex & Intelligent Systems*, *7*(4), 1991–2021. https://doi.org/10.1007/s40747-021-00307-y

Anitha, D., Shamili, G., Ravi, K. P., & Sabari, V. R. (2018). Air foil shape optimization using cfd and parametrization methods. *Materials Today: Proceedings*, *5*, 5364–5373.

Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J., & Morlier, J. (2019). Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and Technology*, *90*, 85–102. https://doi.org/https://doi.org/10.1016/j.ast.2019.03.041

Barton, R. (1992). Metamodels for simulation input-output relations. *Proceedings of the 1992 Winter Simulation Conference*.

Bedon, G., S, D. B., & Benini, E. (2016). Performance-optimized airfoil for darrieus wind turbines. *Renewable Energy*, *94*, 328–340. https://doi.org/http://dx.doi.org/10.1016/j.renene.2016.03.071

Boyd, E. A. (1967). Principles of ideal-fluid aerodynamics.k. karamcheti. john wiley amp; sons inc.1966. 636 pp. illustrations. 160s. *The Aeronautical Journal*, *71*(680), 588–588. https://doi.org/10.1017/S0001924000055093

Bozorg-Haddad, O., Solgi, M., & A., H. (2017). *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. Wiley.

Bramesfeld, G., & Krebs, T. (2016). Using an optimisation process for sailplane winglet design. *Aeronautical Journal*, 1–20.

Carmichael, B. H. (1954). What price performance. *Soaring*.

Catalani, G., Costero, D., Bauerheim, M., Zampieri, L., Chapin, V., Gourdain, N., & Baqu©, P. (2023). A comparative study of learning techniques for the compressible aerodynamics over a transonic rae2822 airfoil. *Computers Fluids*, *251*, 105759. https://doi.org/https://doi.org/10.1016/j.compfluid.2022.105759

Chen, H., He, L., Qian, W., & Wang, S. (2020). Multiple aerodynamic coefficient prediction of airfoils using a convolutional neural network. *Symmetry*, *12*(4), 544.

Chen, W., & Fuge, M. (2018). Béziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters. https://doi.org/10.48550/ARXIV.1808.08871

Chen, Y., Zhang, D., Li, X., Peng, Y., Zhang, X., Han, Z., Cao, Y., & Dong, Z. (2023). Surrogate models for twin-vawt performance based on kriging and artificial neural networks. *Ocean Engineering*, *273*, 113947. https://doi.org/https://doi.org/10.1016/j.oceaneng.2023.113947

Coder, J., & Maughmer, M. (2014). Comparisons of theoretical methods for predicting airfoil aerodynamic characteristics. *J. Aircr*, *51*, 183–191. https://doi.org/https://doi.org/10.2514/1.c032232

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). L2 regularization for learning kernels. *CoRR*, *abs/1205.2653*. http://arxiv.org/abs/1205.2653

D'Alessandro, V., Montelpare, S., Ricci, R., & Zoppi, A. (2017). Numerical modeling of the flow over wind turbine airfoils by means of spalart-allmaras local correlation based transition model. *Energy*, *130*, 402–419. https://doi.org/http://dx.doi.org/10.1016/j.energy.2017.04.134

Daróczy, L., Janiga, G., & Thévenin, D. (2018). Computational fluid dynamics based shape optimization of airfoil geometry for an h-rotor using a genetic algorithm. *Eng Optim*, *50*, 1483–1499.

De Gaspari, A., & Ricci, S. (2010). Combining shape and structural optimization for the design of morphing airfoils. https://doi.org/10.13140/RG.2.1.4823.2404

Drela, M. (1989). An analysis and design system for low reynolds number airfoils. *T.J. Mueller (Ed.), Low Reynolds Number Aerodynamics, Springer-Verlag, Berlin*, 1–12.

Du, Q., Liu, T., Yang, L., Li, L., Zhang, D., & Xie, Y. (2022). Airfoil design and surrogate modeling for performance prediction based on deep learning method. *Physics of Fluids*, *34*(1), 015111.

Du, X., He, P., & Martins, J. R. (2021). Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. *Aerospace Science and Technology*, *113*, 106701. https://doi.org/https://doi.org/10.1016/j.ast.2021.106701

El-Shahat, S. A., Li, G., Lai, F., & Fu, L. (2020). Investigation of parameters affecting horizontal axis tidal current turbines modeling by blade element momentum theory. *Ocean Engineering*, *202*. https://doi.org/https://doi.org/10.1016/j.oceaneng.2020.107176

Fang, K., Li, R., & Sudjianto, A. (2006). Design and modeling for computer experiments. *Chapman Hall/CRC, Taylor Francis Group, Blades Court London SW15 2NU UK*, 23–25.

Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate modeling*. John Wiley Sons.

Goel, T., Haftka, R., Shyy, W., & Queipo, N. (2007). Ensemble of surrogates. *Struct. Multidiscip. Optim*, *33*, 199–2006.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures [IJCNN 2005]. *Neural Networks*, *18*(5), 602–610. https://doi.org/https://doi.org/10.1016/j.neunet.2005.06.042

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. https://doi.org/10.48550/ARXIV.1502.01852

Hicks, R., & Henne, P. (1978). Wing design by numerical optimization. *Journal of Aircraft*, *35*, 407–412.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366.

Horstmann, K. (1976). Neue modellaufwindverteilungen and ihr einflub auf die aiuslegung von segelflugzeugen [german]. *XV OSTIV Congress*.

Houghton, E., Carpenter, P., Collicott, S. H., & Valentine, D. T. (2016). *Aerodynamics for engineering students*. Butterworth-Heinemann.

Huan, Z., Zhenghong, G., Yuan, G., & Chao, W. (2017). Effective robust design of high lift nlf airfoil under multi-parameter uncertainty. *Aerospace Science and Technology*, *68*, 530–542. https://doi.org/http://dx.doi.org/10.1016/j.ast.2017.06.009

Hui, X., Bai, J., Wang, H., & Zhang, Y. (2020). Fast pressure distribution prediction of airfoils using deep learning. *Aerospace Science and Technology*, *105*, 105949. https://doi.org/https://doi.org/10.1016/j.ast.2020.105949

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. https://doi.org/10.48550/ARXIV.1502.03167

J, M., R, V., & MAR, S. (2016). Xfoil vs cfd performance predictions for high lift low reynolds number airfoils. *Aerospace Science and Technology*, *52*, 207–214. https://doi.org/https://doi.org/10.1016/j.ast.2016.02.031

Jeong, J., & Kim, S. (2018). Optimization of thick wind turbine airfoils using a genetic algorithm. *J Mech Sci Technol*, *32*, 3191–3199.

Jin, R., Wei, C., & Simpson, T. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct. Multidiscip. Optim*, *23*, 1–13.

J.Morgadoa, R.Vizinhob, M.A.R.Silvestrea, & J.C.Páscoab. (2016). Xfoil vs cfd performance predictions for high lift low reynoldsnumber airfoils. *Aerospace Science and Technology*, *52*, 207–204. https://doi.org/http://dx.doi.org/10.1016/j.ast.2016.02.031

Joukowski, N. E. (1906). Sur les tourbillons adjionts. *Travaux de la Section Physique de la Société Imperiale des Amis des Sciences Naturales*, *13*, 261–284.

Karali, H., Yukselen, M. A., & Inalhan, G. (2019). A new non-linear lifting line method for 3d analysis of wing / configuration aerodynamic characteristics with application to uavs. *AIAA SciTech 2019 Forum, San Diego, CA*, 2119. https://doi.org/doi:10.2514/6.2019-2119

Karali, H., Demirezen, M., Yukselen, M., & Inalhan, G. (2020). Design of a deep learning based nonlinear aerodynamic surrogate model for uavs. *American Institute of Aeronautics and Astronautics*.

Ken Peffers, M. A. R., Tuure Tuunanen, & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, *30*(1-2), 81–93. https://doi.org/10.1093/biomet/30.1-2.81

Khurana, M., & Massey, K. (2015). Swarm algorithm with adaptive mutation for airfoil aerodynamic design. *Swarm and Evolutionary Computation*, *20*, 1–13. https://doi.org/http://dx.doi.org/10.1016/j.swevo.2014.10.001

Khurana, S., Winarto, H., & Sinha, K. (2008). Application of swarm approach and artificial neural networks for airfoil shape optimization. *American Institute of Aeronautics and Astronautics*.

Kim, M. I., & Yoon, H. S. (2022). Geometric modification for the enhancement of an airfoil performance using deep cnn. *Ocean Engineering*, *266*, 113000. https://doi.org/https://doi.org/10.1016/j.oceaneng.2022.113000

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. https://doi.org/10.48550/ARXIV.1412.6980

Kubrynski, K. (2006). Aerodynamic design and cross-country flight performance analysis. *XV OSTIV Congress*.

Kulfan, B. (2007). A universal parametric geometry representation method - cst. *45th AIAA Aerospace Sciences Meeting and Exhibit*.

Kutta, M. W. (1902). Auftriebskraft in strömenden flüssigkeiten. *Illustrierte Aeronautische Mitteilungen*, *6*, 133–135.

Kwon, H., & Choi, S. (2019). A trended kriging model with r2 indicator and application to design optimization. *Aerospace Science and Technology*, 111–125.

Lalonde, E. R., Vischschraper, B., Bitsuamlak, G., & Dai, K. (2021). Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model. *Journal of Wind Engineering and Industrial Aerodynamics*, *216*, 104696. https://doi.org/https://doi.org/10.1016/j.jweia.2021.104696

Lee, D. H., Lee, D., Han, S., Seo, S., Jik Lee, B., & Ahn, J. (2023). Deep residual neural network for predicting aerodynamic coefficient changes with ablation. *Aerospace Science and Technology*, 108207. https://doi.org/https://doi.org/10.1016/j.ast.2023.108207

Lei, R., Bai, J., Wang, H., Zhou, B., & Zhang, M. (2021). Deep learning based multistage method for inverse design of supercritical airfoil. *Aerospace Science and Technology*, *119*, 107101. https://doi.org/https://doi.org/10.1016/j.ast.2021.107101

Li, J., Du, X., & Martins, J. R. (2022). Machine learning in aerodynamic shape optimization. *Progress in Aerospace Sciences*, *134*, 100849. https://doi.org/https://doi.org/10.1016/j.paerosci.2022.100849

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2016). Hyperband: A novel bandit-based approach to hyperparameter optimization. https://doi.org/10.48550/ARXIV.1603.06560

Li, X., Zhang, L., Song, J., Bian, F., & Yang, K. (2020). Airfoil design for large horizontal axis wind turbines in low wind speed regions. *Renewable Energy*, *145*, 2345–2357. https://doi.org/https://doi.org/10.1016/j.renene.2019.07.163

Li, Z., & Zheng, X. (2017). Review of design optimization methods for turbomachinery aerodynamics. *Progress in Aerospace science*, *93*, 1–23. https://doi.org/https://doi.org/10.1016/j.paerosci.2017.05.003

Luo, X., Zhu, G., & Feng, J. (2014). Multi-point design optimization of hydrofoil for marine current turbine. *Journal of Hydrodynamics Ser B*, *26*, 807–817. https://doi.org/10.1016/S1001-6058(14)60089-5

Mao, J., Hu, D., & Li, D. (2017). Novel adaptive surrogate model based on lrpim for probabilistic analysis of turbine disc. *Aerosp. Sci. Technol.*, 76–87.

Martin, C. H., & Mahoney, M. W. (2021). Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *J. Mach. Learn. Res.*, *22*(1).

Masters, D., Taylor, N., & Rendall, T. (2017). A geometric comparison of aerofoil shape parameterisation methods. *AIAA*. https://doi.org/https://doi.org/10.2514/1.J054943

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163–191. https://doi.org/http://dx.doi.org/10.1016/j.advengsoft.2017.07.002

Mohammadi, S., Hassanalian, M., Arionfard, H., & Bakhtiyarov, S. (2020). Optimal design of hydrokinetic turbine for low-speed water flow in golden gate strait. *Renewable Energy*, *150*, 147–155. https://doi.org/https://doi.org/10.1016/j.renene.2019.12.142

Morgado, J., Vizinho, R., Silvestre, M., & Páscoa, J. (2016). Xfoil vs cfd performance predictions for high lift low reynolds number airfoils. *Aerospace Science and Technology*, 207–214.

Nemati, M., & Jahangirian, A. (2020). Robust aerodynamic morphing shape optimization for high-lift missions. *Aerospace Science and Technology*, *103*. https://doi.org/10.1016/j.ast.2020.105897

Padulo, M., Maginot, J., Guenov, M., & Holden, C. (2009). Airfoil design under uncertainty with robust geometric parameterization. *50th aiaa/asme/asce/ahs/asc structures, structural dynamics, and materials conference*. https://doi.org/10.2514/6.2009-2270

Pires, O., Munduate, X., Ceyhan, O., Jacobs, M., & Snel, H. (2016). Analysis of high reynolds numbers effects on a wind turbine airfoil using 2d wind tunnel test data. *Journal of physics: conference series*, *753*(2).

Prandtl, L. (1921). Appplications of modern hydrodynamics to aeronautics. *Tech. Rep. TR-116, NACA*.

Prandtl, L. (1918). Tragfl¨ugel theorie. *Nachricten von der Gesellschaft der Wissenschaften zu G¨ottingen*, 451–477.

Raghu, M., & Schmidt, E. (2020). A survey of deep learning for scientific discovery.

Ram, K., Lal, S., & Ahmed, M. (2019). Design and optimization of airfoils and a 20 kw wind turbine using multi-objective genetic algorithm and harp$_o$*ptcode*. *Renewable Energy*, *144*, 56–67. https://doi.org/https://doi.org/10.1016/j.renene.2018.08.040

Ram, K. R., Lal, S. P., & Ahmed, M. R. (2019). Design and optimization of airfoils and a 20kw wind turbine using multi-objective genetic algorithm and harp opt code. *Renewable Energy*, *144*, 56–67. https://doi.org/https://doi.org/10.1016/j.renene.2018.08.040

Reid, J. T., & Hunsaker, D. F. (2021). General approach to lifting-line theory, applied to wings with sweep. *Journal of Aircraft*, *58*(2), 334–346. https://doi.org/10.2514/1.C035994

Renganathan, S. A., Maulik, R., & Ahuja, J. (2021). Enhanced data efficiency using deep neural networks and gaussian processes for aerodynamic design optimization.

Aerospace Science and Technology, *111*, 106522. https://doi.org/https://doi.org/10.1016/j.ast.2021.106522

Reynolds, D. A. et al. (2009). Gaussian mixture models. *Encyclopedia of biometrics*, *741*(659-663).

Roberts, S. J., Husmeier, D., Rezek, I., & Penny, W. (1998). Bayesian approaches to gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1133–1142.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7

Saleem, A., & Kim, M.-H. (2020). Aerodynamic performance optimization of an airfoil-based airborne wind turbine using genetic algorithm. *Energy*, *203*, 117841. https://doi.org/https://doi.org/10.1016/j.energy.2020.117841

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, *48*(4), 1875–1897. https://doi.org/10.1214/19-AOS1875

Sekar, V., Zhang, M., Shu, C., & Khoo, B. C. (2019). Inverse design of airfoil using a deep convolutional neural network. *AIAA Journal*, *57*(3), 993–1003. https://doi.org/10.2514/1.J057894

Selig, M. (2003). Low reynolds number airfoil design lecture notes. *VKI Lecture Series, November*, 24–28.

Selig, M. (2022). UIUC airfoil data site. https://m-selig.ae.illinois.edu/ads/coord_database.html

Selig, S., & Guglielmo, J. (1997). High-lift low reynolds number airfoil design. *J. Aircr*, *234*, 72–79. https://doi.org/http://dx.doi.org/10.2514/2.2137

Sharpe, P. (2023). NeuralFoil: An airfoil aerodynamics analysis tool using physics-informed machine learning.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*. https://doi.org/10.1016/j.physd.2019.132306

Shoshan, A., Bhonker, N., Kviatkovsky, I., & Medioni, G. (2021). Gan-control: Explicitly controllable gans. https://doi.org/10.48550/ARXIV.2101.02477

Skinner, N., & Zare-Behtash, H. (2018). State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*, *62*, 933–962. https://doi.org/https://doi.org/10.1016/j.asoc.2017.09.030

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf

Sobieczky, H. (1999). Parametric airfoils and wings. *Recent Development of Aerodynamic Design Methodologies*, 71–87.

Sun, G., Sun, Y., & Wang, S. (2015). Artificial neural network based inverse design: Airfoils and wings. *Aerospace Science and Technology*, *42*, 415–428. https://doi.org/10.1016/j.ast.2015.01.030

Sun, Z., Mao, Y., & Fan, M. (2021). Performance optimization and investigation of flow phenomena on tidal turbine blade airfoil considering cavitation and roughness. *Applied Ocean Research*, *106*, 102463. https://doi.org/https://doi.org/10.1016/j.apor.2020.102463

Tao, J., & Sun, G. (2019). Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. *Aerospace Science and Technology*, *92*, 722–737. https://doi.org/https://doi.org/10.1016/j.ast.2019.07.002

Tao, J., Sun, G., Guo, L., & Wang, X. (2020). Application of a pca-dbn-based surrogate model to robust aerodynamic design optimization. *Chinese Journal of Aeronautics*.

Tao, J., Sun, G., & Guo, X., Liqiang; Wang. (2020). Application of a pca-dbn-based surrogate model to robust aerodynamic design optimization. *Chinese Journal of Aeronautics*.

Thomas, F. (1999). *Fundamentals of sailplane design*. College Park Press.

Timmer, W., & Bak, C. (2013). 4 - aerodynamic characteristics of wind turbine blade airfoils. In P. Brndsted & R. P. Nijssen (Eds.), *Advances in wind turbine blade*

design and materials (pp. 109–149). Woodhead Publishing. https://doi.org/https://doi.org/10.1533/9780857097286.1.109

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(86), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

Van Treuren, K. W. (2015). Small-Scale Wind Turbine Testing in Wind Tunnels Under Low Reynolds Number Conditions [051208]. *Journal of Energy Resources Technology*, *137*(5). https://doi.org/10.1115/1.4030617

Wang, J., Li, R., He, C., Chen, H., Cheng, R., Zhai, C., & Zhang, M. (2022). An inverse design method for supercritical airfoil based on conditional generative models. *Chinese Journal of Aeronautics*, *35*(3), 62–74. https://doi.org/10.1016/j.cja.2021.03.006

Wang, S., Sun, G., & Chen, Y., Wanchunhong. (2018). Database self-expansion based on artificial neural network: Anapproach in aircraft design. *Aerospace Science and Technology*, 77–83.

Wang, Z., Liu, X., Yu, J., Wu, H., & Lyu, H. (2023). A general deep transfer learning framework for predicting the flow field of airfoils with small data. *Computers Fluids*, *251*, 105738. https://doi.org/https://doi.org/10.1016/j.compfluid.2022.105738

Weia, X., Wanga, X., & Chena, S. (2020). Research on parameterization and optimization procedure of low-reynoldsnumber airfoils based on genetic algorithm and bezier curve. *Advances in Engineering Software*, *149*. https://doi.org/https://doi.org/10.1016/j.advengsoft.2020.102864

Weihong, Z., Linying, Z., Tong, G., & Shouyu, C. (2017). Topology optimization with closed b-splines and boolean operations. *Comput. Methods Appl. Mech. Engineering*, 652–670. https://doi.org/doi.org/10.1016/j.cma.2016.11.015

WU, H., LIU, X., AN, W., & LYU, H. (2022). A generative deep learning framework for airfoil flow field prediction with sparse data. *Chinese Journal of Aeronautics*, *35*(1), 470–484. https://doi.org/https://doi.org/10.1016/j.cja.2021.02.012

Xiaoqiang, K., Jun, H., Lei, S., & Jing, L. (2018). An improved geometric parameter airfoil parameterization method. *Aerospace Science and Technology*, 241–247. https://doi.org/https://doi.org/10.1016/j.ast.2018.04.025

Xuesong, W., Xiaoyang, W., & Songying, C. (2020). Research on parameterization and optimization procedure of low-reynoldsnumber airfoils based on genetic algorithm and bezier curve. *Advances in Engineering Software*, *149*. https://doi.org/https://doi.org/10.1016/j.advengsoft.2020.102864

Yondo, R., Andries, E., & Valero, E. (2018). A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Progress in Aerospace Sciences*, *96*, 23–61. https://doi.org/https://doi.org/10.1016/j.paerosci.2017.11.003

Yonekura, K., Wada, K., & Suzuki, K. (2022). Generating various airfoils with required lift coefficients by combining naca and joukowski airfoils using conditional variational autoencoders. *Engineering Applications of Artificial Intelligence*, *108*, 104560. https://doi.org/https://doi.org/10.1016/j.engappai.2021.104560

Yuan, Z., Huo, S., & Ren, J. (2019). A rapid aeroelasticity optimization method based on the stiffness characteristics. *Scipedia*, *35*. https://doi.org/10.23967/j.rimni.2018.12.002

Zhang, S., Li, H., & Abbasi, A. (2019). Design methodology using characteristic parameters control for low reynolds number airfoils. *Aerospace Science and Technology*, *86*, 143–152. https://doi.org/https://doi.org/10.1016/j.ast.2019.01.003

Zhu, G., Feng, J., Li, P., Wang, Z., Wu, G., & Luo, X. (2023). Multi-condition optimisation design of a hydrofoil based on deep belief network. *Ocean Engineering*, *272*, 113846. https://doi.org/https://doi.org/10.1016/j.oceaneng.2023.113846

Zhu, W. J., Shen, W. Z., & Sorensen, J. N. (2014). Integrated airfoil and blade design method for large wind turbines [Special issue on aerodynamics of offshore wind energy systems and wakes]. *Renewable Energy*, *70*, 172–183. https://doi.org/https://doi.org/10.1016/j.renene.2014.02.057

Ziemkiewicz, D. (2017). Simple analytic equation for airfoil shape description. *AIAA J.* https://doi.org/doi.org/10.2514/1.J055986

Zuo, K., Bu, S., Zhang, W., Hu, J., Ye, Z., & Yuan, X. (2022). Fast sparse flow field prediction around airfoils via multi-head perceptron based deep learning architecture. *Aerospace Science and Technology*, *130*, 107942. https://doi.org/https://doi.org/10.1016/j.ast.2022.107942