Copyright Statement

University of Southampton

Faculty of Engineering and Physical Sciences

School of Engineering

Aerodynamics and Experimental Optimisation of Automotive Underbody Diffusers in the Presence of Rake

by

Pawel Kekus

Thesis for the degree of Doctor of Philosophy

November 2021

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Engineering

Doctor of Philosophy

Aerodynamics and Experimental Optimisation of Automotive Underbody Diffusers in the Presence of Rake

by Pawel Kekus

This research project was focused on two related topics—hardware-in-the-loop aerodynamic optimisation, and aerodynamics of automotive underbody diffusers in the presence of rake, defined as an inclination of the underfloor with respect to the ground. Two experimental systems were used for automatic, closed-loop optimisation trials, and for mapping of aerodynamic performance. Each consisted of an Ahmed-type body with a diffuser, with three controlled degrees of freedom, i.e. the model's height above the ground, and inclinations of the underfloor and diffuser plates. The systems were equipped with force acquisition for optimisation and performance quantification purposes, and with surface pressure measurements to inspect the underlying flow patterns.

The high-speed system was used for real-time optimisation runs using a range of algorithms in order to determine their suitability to problems of this type. Population-based algorithms, and genetic algorithms in particular, were found to provide the most reliable convergence in spite of the noise and hysteresis in the measurements. Reductions in pre-sampling delay and sampling time decreased the average function evaluation time without negatively impacting convergence performance, whereas combinatorial optimisation was used to minimise actuation overheads. Subsequently, both methods were shown to improve overall optimisation efficiency during experimental trials.

Finally, the impact of rake on diffuser aerodynamics was investigated through quasi-static variations of the three degrees of freedom. Introducing rake was found to induce significant pressure recovery beneath the underfloor, causing strong suction under the front of the body and increased downforce. Furthermore, two counter-rotating vortices were observed along the edges of the underfloor, whose formation and strength depended on the configuration of the model, and which significantly affected the stall characteristics of the diffuser.

Contents

List of Tables

List of Figures

Declaration of Authorship

I, Pawel Kekus, declare that this thesis and the work presented in it are my own, and have been generated by me as the result of my own original research.

I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- Parts of this work have been published as:

  P. Kekus and D. Angland, "Automatic wind tunnel-based optimisation of an automotive underbody diffuser," in *2018 AIAA Aerospace Sciences Meeting*, (Kissimmee, FL, USA), American Institute of Aeronautics and Astronautics, 2018. AIAA 2018-0045.

Signed: Date:

Acknowledgements

The completion of this work would not have been possible without the continuous support I have received over the course of the project. I would like to extend my sincere gratitude, and simultaneously dedicate this work, to the following individuals:

- First of all, to my supervisor, David Angland, who provided support infallibly, regardless of the nature of my needs, openly shared his knowledge and experience, and advised or criticised, as appropriate, whenever a need arose;
- To my partner-in-life, Dominika, who walked with me every step of the way, bringing comfort and resupplying motivation;
- To my colleagues from the ANTC, with whom I shared this perilous journey, and who had the ability to ask just the right questions to stimulate my work in the right direction;
- To Dave Marshall, who selflessly shared his expertise in order to maximise the quality of my work;
- And finally, to every single person studying or working at the University of Southampton, who collectively gave me a home without me even realising it, and an environment that was simultaneously safe, comforting, inspiring, and entertaining.

May the force be with you, always.

Nomenclature

Roman Symbols

AR	Maximum aspect ratio of a design space
B_0	Baseline dimension length of a design space
B_n	Length of the n-th dimension of a design space
C, $\mathfrak{C}$	Cost in optimisation
C_D	Drag coefficient
C_F	Force coefficient
C_L	Downforce coefficient
C_M	Moment coefficient
C_p	Pressure coefficient
$E(\{r_i\})$	System energy in simulated annealing
F	Force
F_x	Drag force
F_y	Side force
F_z	Downforce
H	Model height
I	Relative cost reduction due to sorting
$I_{\max}$	Maximum possible cost reduction due to sorting
L	Permutation length
L/D	Lift-to-drag ratio
L_M	Model length
M	Moment
M_x	Rolling moment
M_y	Pitching moment
M_z	Yawing moment
N	Number of dimensions of a design space

R	Random number
Re	Reynolds number
S	Model frontal area
T	Simulated annealing temperature
U_∞	Freestream velocity
V	Actuator voltage
W	Model width
c	Hyperplane equation coefficient
c_{max}	Maximum permitted hyperplane equation coefficient
d	Model half-width
f	Fitness in particle swarm optimisation
h_1	Ride height
h_2	Ride height at diffuser inlet
h_3	Ride height at diffuser exit
k_B	Boltzmann constant in simulated annealing
k_c	Cognitive constant in particle swarm optimisation
k_s	Social constant in particle swarm optimisation
n_{swap}	Number of pairs of cities to swap in simulated annealing
p	Design point
q_∞	Freestream dynamic pressure
t	Time
t_{max}	Sorting time limit
v	Particle velocity in particle swarm optimisation
v_{act}	Actuator speed
w	Inertia weight in particle swarm optimisation
x	Drag force axis
y	Side force axis
z	Downforce axis

Greek Symbols

Δ	Difference
γ	Rake angle
θ	Diffuser angle
μ	Mean
σ	Standard deviation

Acronyms

ACO	Ant colony optimisation
ADC	Analogue-to-digital converter
CDF	Cumulative distribution function
CFD	Computational fluid dynamics
CTSP	Chebyshev travelling salesman problem
DAC	Digital-to-analogue converter
DAQ	Data acquisition
DC	Direct current
DoF	Degree of freedom
FEA	Finite element analysis
GA	Genetic algorithm
HIL	Hardware-in-the-loop
LDA	Laser Doppler anemometry
LE	Leading edge
PAM	Pneumatic artificial muscle
PC	Personal computer
PIV	Particle image velocimetry
PSO	Particle swarm optimisation
SA	Simulated annealing
SD	Standard deviation
SMA	Shape memory alloy
SMP	Shape memory polymer
SUS	Stochastic universal sampling
TE	Trailing edge
TSP	Travelling salesman problem
UAV	Unmanned air vehicle
USB	Universal Serial Bus

Chapter 1

Introduction

Optimisation problems are present in engineering disciplines of all kinds and magnitudes, from microchips to space telescopes, and from nanotechnology to megastructures. Whenever a device, vehicle or structure is designed, it needs to satisfy certain requirements while maximising value, and at the same time to minimise negative impacts on stakeholders. The requirements might be dictated by safety or market demand and value typically stems from performance metrics, aesthetics, or perceived quality, whereas the negative impacts usually include upfront and long-term costs and environmental pollution.

In order to tackle the omnipresent optimisation challenges, engineers and scientists have devised a myriad of algorithms and methods. By formalising and automating the process of optimisation, greater gains can be achieved at a lower cost compared to trial-and-error or manual techniques. The aerospace industry has largely contributed to the establishment of multidisciplinary optimisation, where several aspects of a design, e.g. its aerodynamic and structural performance, are optimised simultaneously [1, 2]. This is crucial in the endeavour to maximise profit while remaining socially and environmentally conscious.

The field of aerodynamic optimisation encompasses all problems involving relative motion of an object and a fluid. In most cases, the object in question is a vehicle, whether terrestrial or aerial, which moves through the atmosphere. However, aerodynamic optimisation is also increasingly utilised in the design of structures, for example high-rise buildings or bridges, which have to withstand wind forces in addition to structural loads.

Aerodynamic development of all types of vehicles is driven by the need to maximise energy efficiency and reduce susceptibility to wind by minimising drag. However, race car aerodynamics is concerned with an additional element: maximising downforce in order to increase cornering speed. Whereas drag reduction is typically achieved through streamlining the geometry and reducing flow separation, downforce is produced with specifically designed elements. Typical examples include a wing or a splitter mounted at the front of the vehicle, a wing or a spoiler at the back, and a flat underfloor with a diffuser between and behind the rear wheels.

There are three key problems that are of particular significance to the aerodynamic optimisation process, viz. model parametrisation, optimisation method selection, and performance quantification. Parametrisation, or describing a geometry with a set of numerical variables, allows the algorithm to establish new configurations or shapes without the aid of a designer. The chosen method must consistently provide as high performance as possible, while using minimal time and resources. Finally, once a promising configuration is created, its performance, i.e. the forces and moments generated, surface pressure distributions, wake characteristics, etc., must be evaluated in order to inform the algorithm of the characteristics of the problem. Performance evaluation is typically carried out through computational fluid dynamics (CFD), which is a group of numerical methods that simulate fluid flow around an object. The main reasons for this are relatively low upfront costs, ease of geometry manipulation, and the resulting breadth of information about the flow. However, accurate simulation of complex, unsteady flows, as in an automotive diffuser, involves great computational cost, as well as uncertainty inherent to the approximations and models used.

1.1 Rationale

An optimisation process requires dozens of function evaluations, or even hundreds or thousands for larger problems, in order to determine the optimal or near-optimal solution. In the case of aerodynamics, a function evaluation corresponds to either a CFD simulation, iterating until satisfactory convergence is attained, or an experimental measurement, e.g. time-averaged forces or pressure, which is typically carried out in a wind tunnel, but may also be performed in motion in the open air. The process between the measurements involves updating the geometry and meshing in the case of numerical methods, and redesigning or manually adjusting the model in the case of wind tunnel testing. When the former is used, it is the computation itself that is most time-intensive, especially in unsteady problems, which require time-resolved CFD, although meshing can be resource intensive as well. In the case of wind tunnel experiments, individual data points take in the order of seconds to acquire, but it is the intermediate process that takes the most time, i.e. stopping the wind tunnel, rebuilding or adjusting the model, and restarting the wind tunnel.

The goal of this research is to combine the two fast processes: quick data acquisition enabled by experimental measurement systems, and rapid geometry modifications. This can be achieved by placing a body capable of changing its shape in a wind tunnel and giving control of the geometry to an optimisation algorithm, similar to the procedure of optimisation using numerical methods. Finally, the algorithm needs to be able to automatically acquire and process measurements, which closes the loop. The whole process, previously termed hardware-in-the-loop (HIL) aerodynamic optimisation, promises to reduce the total time required to optimise an aerodynamic geometry from days or weeks to hours. Furthermore, sufficiently fast morphing may enable a more thorough exploration of the search space, or even an exhaustive search, reducing the likelihood of omitting the global optimum. On the other hand, the need for a physical model restricts the ability to modify the shape to only the degrees of freedom included in model design, which limits the accessible search space relative to CFD-based optimisation. Although systems of this kind have been created before, they frequently suffered from poor speed and reliability. This work attempts to address those issues, while also exploring ways to improve the efficiency of the process. This is achieved firstly by minimising operational overheads, such as sampling time during data acquisition, and secondly by utilising combinatorial optimisation to reduce the time spent actuating the model.

The automotive underbody diffuser is used as the primary geometry for this study, as it represents a complex, multidimensional aerodynamic optimisation problem that may be easily reproduced through a physical model. Due to the inherently three-dimensional and unsteady flow physics involved, this problem is difficult and expensive to solve computationally, making it ideally suited for an experimental investigation.

Furthermore, the morphing capability of the models built for this study is used for a detailed investigation into diffuser aerodynamics, which attempts to fill key knowledge gaps in published literature. Crucially, the behaviour of diffusers in the presence of rake, or a positive incidence of the underfloor with respect to the ground, is poorly documented, despite the frequent use of rake in racing cars, most notably in Formula 1. The use of an actuated geometry permits high-resolution mapping of diffuser performance with respect to ride height, rake angle, and diffuser angle. Quasi-static variations of rake and diffuser angles during continuous wind tunnel operation are a novel application of automation, and enable previously unexplored phenomena to be investigated.

The specific objectives of this work are:

- To devise a fully-functioning hardware-in-the-loop aerodynamic optimisation system with sufficient reliability to enable prolonged, continuous tests without operator input;
- To assess the suitability of several types of optimisation algorithms, and to identify their most advantageous characteristics;
- To investigate whether reducing the settling and sampling times in wind tunnel-based optimisation is a viable method for improving the efficiency of the process, and to examine the impact on convergence characteristics;
- To further improve the operational efficiency of hardware-in-the-loop optimisation by utilising combinatorial optimisation, and to devise a generalised methodology for maximising the gains;
- And to investigate the effect of rake on the aerodynamics of a diffuser-equipped bluff body in ground effect, in terms of performance as well as flow patterns.

1.2 Thesis Structure

The thesis begins with a literature review in Chapter 2, which places this work in the context of previously published research, and provides an essential background for the discussion of results in the subsequent chapters. Chapter 3 introduces the methodologies used in this study, with an emphasis on design and procedures, as well as the limitations of the experiments.

The results presented in Chapter 4 pertain to the comparative hardware-in-the-loop aerodynamic optimisation tests, where the downforce generated by a diffuser-equipped bluff body in ground effect was maximised in a wind tunnel, using several distinctive algorithms. Average convergence performance is presented for all the algorithm configurations used, and the trends are linked to particular characteristics of the algorithms. Then, the results of the sampling time study are presented, where the impact of reducing the settling and sampling times on convergence and operational efficiency was investigated numerically, with subsequent experimental validation.

Chapter 5 presents the results of the population sorting study, where combinatorial optimisation was utilised to reduce total actuation time during fitness evaluation of a group of configurations. The effectiveness of several algorithms was tested using a broad range of problem parameters, in order to make the recommendations applicable to any problem with a similar cost metric. Furthermore, a methodology for maximising operational gains is presented, as well as the results of its application to the hardware-in-the-loop optimisation problem from Chapter 4.

Chapter 6 is focused on the aerodynamics of automotive diffusers, and discussed therein are the force and surface pressure measurements acquired using the same wind tunnel models as in the optimisation tests in Chapters 4 and 5. Sweeps of ride height, rake and diffuser angles were carried out using various model configurations, enabling a detailed analysis of the performance of diffusers in the presence of rake, and notably, of the evolution of flow patterns under the model as rake is applied.

Finally, Chapter 7 presents the conclusions drawn from this work, highlighting the key findings and their limitations, as well as the suggestions for future work, which would complement the results presented in this thesis, and fill the key gaps recognised in the existing knowledge.

Chapter 2

Literature Review

The following chapter presents a review of previous research on the topic of hardware-in-the-loop aerodynamic optimisation, with a focus on practical aspects of the implementations, such as actuation methods, and on other important considerations, such as tackling hysteresis. This is followed by a review of novel morphing systems and the types of problems a HIL system might be applied to. Next, a review is conducted of alternative actuation methods that could improve shape articulation beyond what traditional mechanical systems might achieve. Afterwards, detailed descriptions of the optimisation methods used are presented, including existing variations and theoretical characteristics, which helps to understand the behaviour of the algorithms. An overview of the travelling salesman problem (TSP) is also given, which forms the background for the combinatorial optimisation part of this work. It is supplemented by a brief overview of the sorting algorithms used in this study. Finally, a review of research on aerodynamics of automotive underbody diffusers is presented.

2.1 Aerodynamic Optimisation

2.1.1 Hardware-in-the-Loop Aerodynamic Optimisation

Possibly the first documented example of research on automatic wind tunnel-based optimisation is credited to Levinsky & Palko [3], who back in 1982 published an article describing the optimisation of a flexible wing and tail configuration with 13 degrees of freedom (DoFs), using a hardware-in-the-loop system. They presented firm reasoning on why CFD is prohibitively costly, and why automatic experimental optimisation is limited in terms of wing articulation and shape control. Although both of these issues have since diminished in magnitude with the growth of computational power and the development of new materials and actuation technologies, they remain crucial factors for practical implementations. The authors optimised an aircraft's configuration for minimised drag at a particular lift coefficient, while trimming the aircraft using the tail. This represents the typical design objective of any aircraft, i.e. providing sufficient lift to overcome the weight and ensuring longitudinal stability by the means of a horizontal stabiliser, while minimising fuel consumption due to aerodynamic drag. The testing conditions were representative of a typical passenger cruise flight—measurements were taken at velocities up to Mach 0.85, which is the common cruise Mach number for long-haul passenger airliners. The article describes tests carried out on an improved model, which was meant to overcome the structural and reliability shortcomings of the authors' previous build [4, 5]. They suggest that the most significant issue was shape deformation under aerodynamic loads, which led to biased results and poor repeatability. The latter issue was also experienced by the hydraulic actuation system of the original model. In order to alleviate the bias resulting from profile deformation, the authors applied photogrammetric shape verification. A significant focus was placed on model safety, with imposing constraints to prevent impossible or risky configurations identified as a critical issue. The optimisation algorithm applied was gradient projection-based and was not capable of sufficient exploration of the multidimensional search space, which resulted in very small improvements over baseline configurations. Although the results of the experiments were described as successful, the potential of the morphing aircraft model could be utilised further with a more capable algorithm.

Another hardware-in-the-loop system based in a wind tunnel was presented in 1994 by Rioual et al. [6] of the University of Southampton, UK. Their system involved active closed-loop control of transition over a flat plate with uniformly distributed suction for the purpose of drag reduction. The authors developed a reliable method of localising the transition point over the plate by measuring pressure fluctuations, and used it as feedback for the controller driving the suction, which allowed them to successfully control the location of transition. Although the experiments did not involve design optimisation, they are relevant due to their similar nature—a closed-loop measurement and actuation system running in a wind tunnel.

The first investigation using hardware-in-the-loop evolutionary aerodynamic optimisation might be the work of Hunt et al. [7] of the NASA Ames Research Center, who successfully optimised the flapping rate and tail position of a 4-foot ornithopter, shown

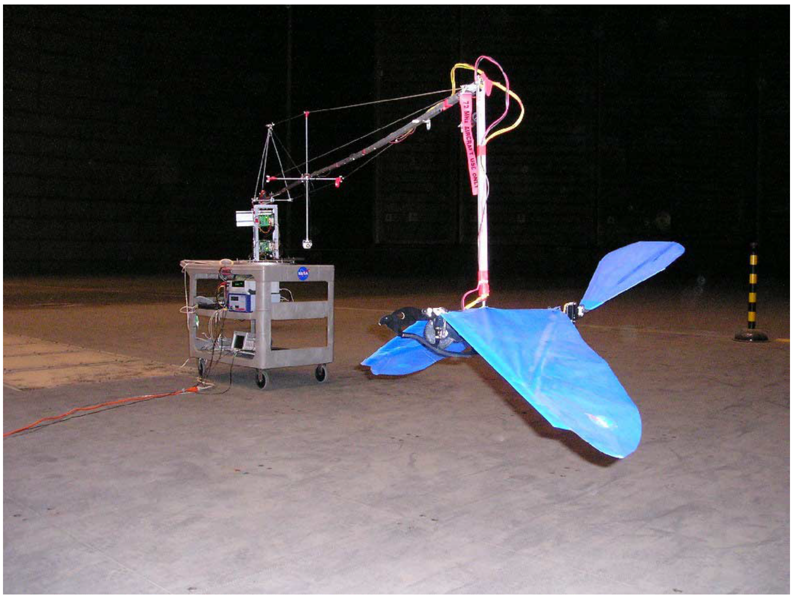


FIGURE 2.1: The ornithopter of Hunt et al. [7] mounted on a rotating test platform.

in Fig. 2.1, for maximum speed and efficiency using a genetic algorithm (GA). The authors explained why this type of optimisation might be preferable to using CFD, and considered a number of aspects important for the process, such as flow settling time, repeatability of measurements, relative performance measures, wear, and other hardware issues. They incorporated a null parameter into the control system in order to monitor the GA's sensitivity to a variable that did not have any measurable effect. Consecutive steps in the automation process were also discussed, raising issues such as the degree of control over the model, which is linked to the number of DoFs in the system. The closed-loop optimisation run was successful, with an unexpected optimal configuration reported. Importantly, tests were not carried out in a wind tunnel—the ornithopter performed tethered flight, supported by a rotating structure that was used for load acquisition, as seen in Fig. 2.1. This article is particularly significant, as it gives an example of effective large-scale evolutionary optimisation, while applying it to a problem uncommon in the field.

Boria et al. [8] designed and built a thin-airfoil wing with variable camber, controlled at two actuation points—near the quarter chord position and at the trailing edge (TE). They devised a real-time control system and used a GA to optimise camber for maximum lift coefficient and lift-to-drag ratio. Actuation was done with servos, and force measurements were collected with a strain gauge balance, although alternative actuation methods were also suggested. Thorough validation of the system was shown, but no insight into the efficiency of the process was presented, with issues such as algorithmic optimisation and flow settling omitted. The authors suggested that a major difficulty with the application of GAs to aerodynamic optimisation is experimental repeatability. Although this is an important issue, flow settling tests and hysteresis alleviation could be used to mitigate it. Finally, this publication highlights the potential applications of similar systems, including active wing morphing to adjust the geometry for an array of flight manoeuvres, such as high-speed transit, efficient loiter, or agile manoeuvring through urban environments.

A significantly more complex morphing wing was devised by Cosin et al. [9], which although rectangular in planform shape, could vary its camber and twist in 10 DoFs through modified leading edge (LE) and trailing edge positions at 5 spanwise locations, as seen in Fig. 2.2. Their objective was to optimise the wing's lift-to-drag ratio for the purpose of adaptation to various flight conditions using a GA in a closed-loop control system. A thorough investigation using finite element analysis (FEA) was carried out in order to establish the mechanical and structural properties of the flexible wing. Unfortunately, the authors did not explore the potential of their wing and optimisation system, with no analysis of optimisation performance, repeatability, hysteresis or convergence.

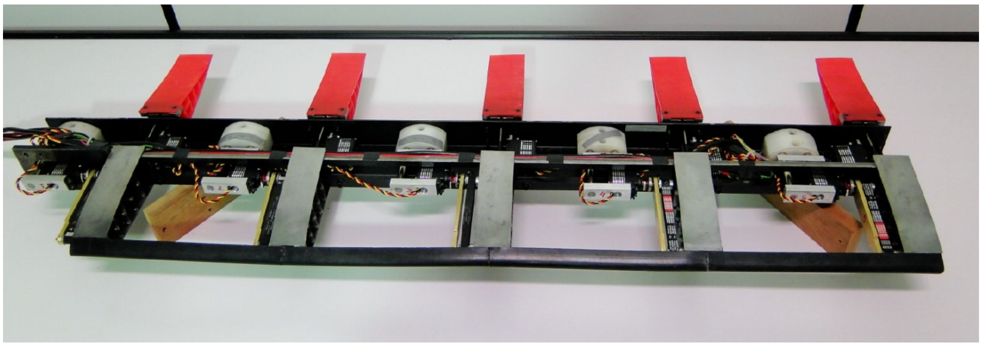


FIGURE 2.2: Internal view of the morphing wing by Cosin et al. [9].

Arguably the most extensive research on the topic of hardware-in-the-loop aerodynamic optimisation was carried out at the École de technologie supérieure in Montréal. Popov et al. [10] presented the design of a rectangular wing with a morphing suction surface for the purpose of extending the laminar flow region over the wing, and complemented it with a comprehensive overview of drag reduction and wing morphing techniques. The wing surface was actuated vertically at two chordwise positions by shape memory alloy (SMA) actuators, which were controlled in a closed loop by a hybrid algorithm, combining simulated annealing (SA) and hill climbing. The actuation system, aside from its complexity, was reported to be unreliable, slow and power inefficient. Moreover, it had a tendency to overheat, which prevented the authors from carrying out prolonged continuous tests. Its slow pace was due to the need to cool down the SMA actuators, and means that this actuation method would significantly diminish the benefits of hardware-in-the-loop aerodynamic optimisation. The high voltage and power requirements might be a further disadvantage for wind tunnel applications due to the need for atypical electricity provision, and the related safety procedures. Although the system worked according to expectations and yielded performance improvements, it was not entirely independent, as it was supported with an airfoil database. In a further publication, Popov et al. [11] described the validation process of the measurement system and the morphing wing. This article provided more detail about the SMA actuators, the flexible skin and the hardware used, as well as on the data acquisition process. Coutu et al. [12] presented a numerical analysis of the system, accompanied by further experimental validation. Sensitivity and actuator displacement limits were also discussed, raising the important issue of the global optimum potentially lying outside of the physically available search space.

Perseghetti et al. [13] recently applied closed-loop GA-based optimisation to improve the flapping pattern of a flapping wing air vehicle of their own construction. Although emphasis was placed on the bird's low weight and modularity, it also performed consistently and reliably. Two flight conditions were considered—gliding and flapping—and weighted objective functions were used for optimisation. The goal for gliding was to maximise the lift-to-drag ratio and minimise aerodynamic moments by controlling wing incidence, and in the case of flapping motion, power consumption and drag were minimised while maximising lift, by modifying the flapping rate and the ratio of the up and down strokes. The authors reported successful completion of tests, with trends in the results confirming predictions. They also highlighted repeatability issues, emphasising that identical configurations in subsequent GA generations should be retested due to noise, instead of using fitness from older generations. This publication provides further validation of the concept of HIL aerodynamic optimisation, while forming a good basis for future investigations on more complex systems.

A chronological summary of research on the concept of hardware-in-the-loop aerodynamic optimisation, including the number of degrees of freedom and optimisation method used in each study, is presented in Table 2.1.

Table 2.1: Summary of prior hardware-in-the-loop aerodynamic optimisation research.

Authors	Year	DoFs	Optimisation algorithm	Closed-loop
Levinsky & Palko [3–5]	1982	8	Gradient projection-based	Yes
Rioual et al. [6]	1994	-	-	Yes
Hunt et al. [7]	2005	3	Genetic algorithm	Yes
Boria et al. [8]	2009	2	Genetic algorithm	Yes
Cosin et al. [9]	2010	10	Genetic algorithm	Yes
Popov et al. [10, 11], Coutu et al. [12]	2010-2011	2	Hybrid: hill climbing and simulated annealing	Partly
Perseghetti et al. [13]	2015	2	Genetic Algorithm	Yes

2.1.2 Aerodynamic Morphing

Shape morphing has been inherently present in aeronautics since its birth in the 19th century. Traditionally, it constitutes any change of shape of an aerodynamic body, most frequently of an aircraft. As such, control surface deflection may be considered morphing, but more recently the term has been associated with more advanced shape changes, such as variable wing twist, span or camber. The morphing process, often inspired by the capabilities of birds' wings [7, 8, 14, 15], can be carried out either through rigid body motion, or by transforming flexible surfaces. The first can be subdivided into three categories:

- Rotation, which includes any hinged motion, such as variable wing sweep, simple flaps, variable angle of attack of horizontal stabilisers, rudders, deployable landing gear, foldable wings, etc. [15–17];
- Sliding, which is mainly applied for altering aircraft wing span [14, 18];
- And a combination of sliding and rotation, the most common example being flap and slat deflections to modify wing camber [14, 18, 19].

Applications of flexible surfaces have been found in variable fixed-wing aircraft wing camber [14, 20–26] and rotorcraft blade camber and twist [21, 27]. However, this type of morphing is not yet common among commercial or military aircraft for several reasons:

- Difficulty in predicting structural properties of elastic materials due to hysteresis and the influence of aerodynamic loads [20];
- Complex shape response, which makes precise shape prediction challenging [25];

- High complexity and mass of the required systems;
- And Poisson ratio issues (extending a flexible surface in one direction typically makes it shrink in the perpendicular directions) [18, 22].

Furthermore, flexible morphing structures must be simultaneously thin to minimise the required energy input, and strong to withstand high external loads due to aerodynamic pressure [27, 28]. Another key aspect is compliance, or the ability to respond to actuation input with changing shape, but without hinges (either conventional or flexural) [20, 22–24]. This carries the advantages of ease of manufacture, resistance to fatigue, no backlash and eliminated need for lubrication [20]. Finally, the structure should be optimised for maximum stiffness countering out-of-plane external (aerodynamic) loads, and minimum stiffness resisting in-plane actuator displacement [18, 20, 22–24].

Going beyond traditional shape morphing that has been widely used for decades, a number of researchers and engineers attempted to devise innovative, more complex or more efficient systems for shape adaptation. These studies are seen as crucial background for automatic hardware-in-the-loop aerodynamic optimisation. Some examples of successful work are presented in the following sections, with focus on multidimensionality and optimisation potential.

2.1.2.1 Novel Morphing Systems

Moosavian et al. [17] designed a morphing mechanism (“a variable geometry truss manipulator”) comprised of cells with “fully variable geometry”. The cells, pictured in Fig. 2.3, had airfoil profiles at both ends, and were morphed through the application of 4 linear actuators and 4 passive, lockable members, which collectively were the only structural link between the two ends. This allowed full control of the orientation of one profile with respect to the other (i.e. relative sweep, twist, and dihedral). The authors proposed the root of a wing tip fin as a potential application, where two adjacent cells would allow the wing tip to morph from parallel to perpendicular to the wing, through a range of cant angles. The fin’s sweep and relative angle of incidence could also be controlled. This application, with snapshots of four different configurations, or poses, is shown in Fig. 2.3. According to the authors, the main strength of the system is that when the structure is passive, all the limbs act as structural members, which provides stiffness and redundancy. They proposed an under-actuated mechanism, where the number of controllable DoFs is larger than the number of actuators. Therefore, the

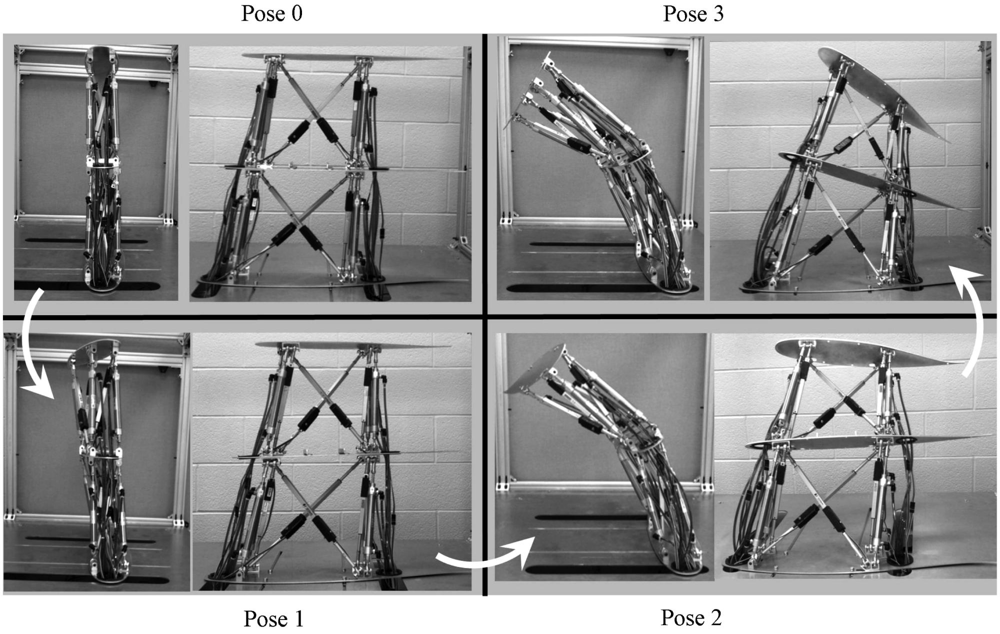


Figure 2.3: Side and front views of the morphing cell prototype by Moosavian et al. [17].

mechanism has to go through a series of stages where the passive members are sequentially locked and unlocked in order to achieve the final position. An analytical method for deriving a minimum-energy actuation strategy was devised through a mathematical analysis of the statics and kinematics of the system. The focus of this publication was on refining the system's performance and optimising the actuation strategy, which led to a fully operational model. It is therefore relevant as a feasibility proof for a morphing concept, where a body's shape and orientation may be modified in many DoFs by several parallel actuators, which could be applied to automatic, multidimensional optimisation, for example of a wing tip fin.

Maki and Hirabayashi [19] designed and tested a multi-element unmanned air vehicle (UAV) rectangular wing with 5 chordwise segments, whose separation could be controlled by a single actuator and a linkage system. In this arrangement, the wing could transition from a single-element wing, through a slat-flap configuration, to a 5-element slotted wing. Expanded and unexpanded configurations using two different linkage systems are shown in Fig. 2.4. The system's purpose was to delay stall, and wind tunnel measurements indicated a much higher stall angle and lift coefficient for the slotted configurations. The single actuator design was lightweight and small in size, but limited the freedom of morphing of the multi-element wing. An expanded concept, with a greater number of controlled degrees of freedom, could be a viable hardware-in-the-loop optimisation problem. It is also noteworthy that the wing's actuation mechanism was

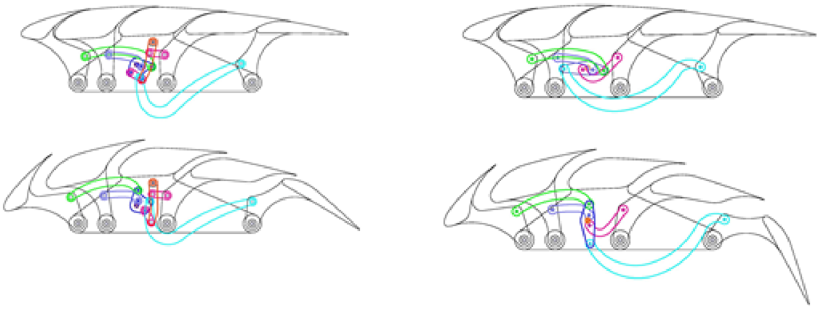


Figure 2.4: Schematics of the multi-element wing by Maki and Hirabayashi [19] using two different linkage systems.

contained within the winglets. A similar solution may be of use in any situation where an external actuation system would induce undesirable aerodynamic effects.

Abdulrahim et al. [28] applied torque rods to the membrane wings of a 60 cm-span micro air vehicle for the purpose of improving lateral control of the vehicle, which in this case was previously controlled by the elevator and the rudder. Whereas the investigation was focused on the impact of twisting the wings on control and stability, the concept is also a good candidate for automatic optimisation. A UAV with an atypical morphing or control mechanism that does not have well-established dynamic properties could be evaluated and optimised in flight by a stochastic algorithm in order to find the most suitable response of control surfaces to either control input or airflow disturbance. However, the structural dynamics of the vehicle would need to be considered, to ensure that the vehicle does not morph into a configuration that promotes structural instability.

Neal et al. [16] described the design of their fully reconfigurable aircraft, which is a simple research model incapable of flight, but can change the span and sweep of its wings, wing tip twist, tail moment arm, as well as the span and incidence of the tail. The model's main purpose was modelling and optimisation of morphing aircraft quasi-statics and dynamics. This represents a complex problem that could benefit from hardware-in-the-loop optimisation. Unfortunately, although the article thoroughly describes the design and structural tests of the model, no optimisation was presented.

2.1.2.2 Shape Memory Actuators

Shape memory actuators are wires made of specific metal alloys that expand and contract under the influence of changing temperature through crystalline phase transformation, with strain reaching up to 10%. A major advantage, from the point of view of aerospace applications, is their strength, which allows them to carry structural loads, but a number of issues make their application difficult. Long-term reliability and performance are frequently reported issues, as is their dependence on fatigue. Fatigue life is affected by the manufacturing process and initial heat treatment, as well as the type and magnitude of stressing and heating cycles [10, 21, 26]. Their slow response in a trade-off with fatigue is also a known issue [21], which makes them unattractive for hardware-in-the-loop optimisation, where a quick response is paramount. Popov et al. [10] commented on the "very long time response" of the SMA actuators and had to adapt their optimisation algorithm to compensate for this (the time needed to transform between extreme positions was claimed to be 2 minutes). In addition, the behaviour of the actuators is non-linear, making it difficult to establish the exact extension, necessitating the use of linear potentiometers or photogrammetric shape verification. The application of SMA actuators may be viable for quasi-static aircraft morphing, where structural strength is preferable to quick response; this has previously been demonstrated for twist and camber morphing [21, 26, 29]. However, despite their commercial availability, it is unlikely that they could be successfully implemented into a hardware-in-the-loop optimisation system due to their slow response, low strain, poor reliability, and high power and cooling requirements.

An alternative to SMA actuators are shape memory polymers (SMPs) [30]. This material radically varies its elasticity under thermal stimuli through the process of thermo-molecular relaxation, and is able to recover an initial, memorised state in the flexible form. Perkins et al. [30] described two types of adaptive materials utilising SMPs. Their dynamic modulus composites employ shape memory polymers as the resin in fibrous composite materials, allowing them to be reshaped above the activation temperature, as shown in Fig. 2.5. Dynamic modulus foam, on the other hand, is a low-density composite foam, which can undergo similar thermal reshaping to dynamic modulus composites. Moreover, it has the ability to fill cavities, but it lacks the composites'

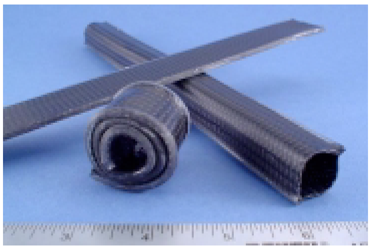


Figure 2.5: Dynamic modulus composite tubes in different shapes [30].

structural strength in the rigid state. It was suggested that the level of development of these novel materials renders them ready for aerospace applications [30]. However, the descriptions suggest that the tests were not representative of the target conditions, reducing their applicability to a wind tunnel-based optimisation system.

2.1.2.3 Piezoelectric Actuators

Another type of adaptive materials are piezoelectric actuators. The name encompasses all actuators based on materials that exhibit a direct relationship between strain and electric potential. Commonly called piezoelectrics, they can be used in two ways:

- As displacement or vibration sensors, where applied strain generates voltage, which in turn may be directly measured;
- Or as actuators, where applied voltage induces mechanical strain.

Many different types of actuators incorporating piezoelectric materials exist, and they have been widely applied in the aerospace industry, usually to alter the camber or twist of rotor blades, or to modify the camber or surface of a wing (through hingeless flaps, airfoil tailoring, etc.), as well as for flow control [27]. Some of the commonly quoted advantages are actuation with a high force output and sensing over a wide frequency range [27, 31, 32], low current drain [27], good repeatability [27, 33], and the fact that they are inherently embedded in the external surface. However, there are also numerous issues, most notably a high voltage requirement [23, 27, 31], low strain [27, 31, 32, 34], significant hysteresis [23, 27, 31–34], and noticeable creep [31]. The actuators' response is described as non-linear [23, 33], although a linear regime was also observed [27].

Whereas piezoelectrics' low strain prevents their application in many fields, it is known that minute changes in shape or control input can induce drastic changes in the aerodynamic characteristics of a geometry [27]. A potential application of piezoelectrics is fine-tuning of the curvature of a body that had been coarsely optimised using different, large-strain actuation methods. The advanced development stage and wide commercial availability of this type of material makes it attractive for the purpose of hardware-in-the-loop optimisation. Outlined below are prior aerospace applications of piezoelectric actuators, which form the background for their applicability to this research.

Kota et al. [20] designed FEA and Xfoil models of compliant leading and trailing edge flaps, and built a physical model of the compliant TE flap. The main proposed application of this system was expanding the flight envelope of high-endurance aircraft without deploying conventional flaps, in order to provide lift variation without introducing flow separation. The authors also created a system where piezoelectric actuators deployed vortex generators through a surface at frequencies up to 90 Hz, and used it for the purpose of flow control.

Usher et al. [33] bonded piezoelectric actuators to the surface of an inflatable wing in order to induce camber change for increased lift. Although no wind tunnel testing was carried out, the authors reported successful wing curvature manipulation. The actuators were bonded to the wing surfaces in two ways—directly, and indirectly through a steel or aluminium substrate. It was found that indirect bonding diminishes the flexibility of the wing, but provides greater displacements than direct bonding. Finally, the authors presented a high level of detail on the practical aspects of preparing the system, including actuator model, adhesive type, and curing method.

Molinari et al. [23, 35] designed and built compliant wings with morphing trailing sections for the purpose of roll control, with the aim of replacing traditional ailerons. The bending was driven by sets of piezoelectric actuators on both sides of the wings, and enabled by compliant ribs and corrugated skin on the pressure side, as illustrated in Fig. 2.6. Evolutionary optimisation of the connectivity and geometry of the truss-like ribs was crucial for obtaining desired aerodynamic properties under bending, as well as for tuning the compromise between morphing ability and resistance to aerodynamic loads. Furthermore, the thickness ratio of the actuators and the underlying skin was optimised to maximise rolling moment while minimising the mass of the system. The hysteresis and creep of the actuators necessitated a closed-loop system based on strain gauges to control the bending of the wing. Despite higher-than-predicted stiffness of

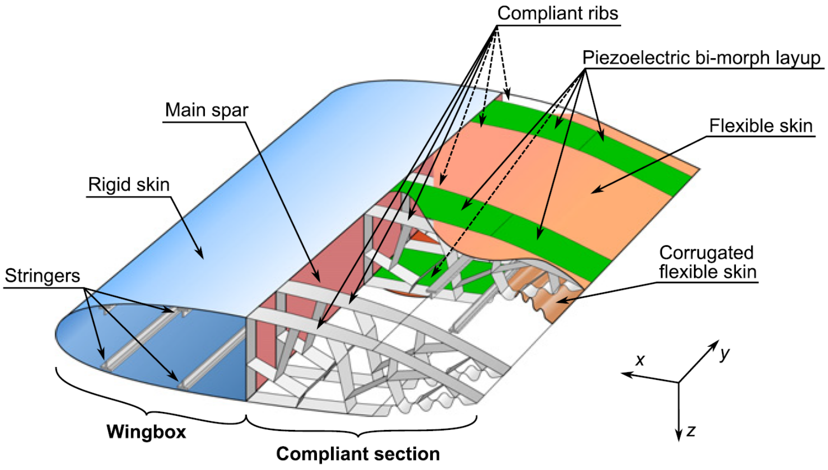


Figure 2.6: A compliant, piezoelectric wing design by Molinari et al. [23].

the manufactured wing, the authors found that their system provided sufficient roll authority, and enabled a significant variation of the lift coefficient in cruise.

Pinkerton and Moses of NASA [31] applied piezoelectric actuators to manipulate airfoil surfaces for the purpose of transition delay. Significant emphasis was placed on the practical properties of piezoelectrics. The authors commented extensively on creep, which was notable, but partly caused by the tension of the flexible membrane, and on hysteresis, which was very significant and consistent across all experiments. They also observed that the actual membrane position and actuator displacement were dependent on wind tunnel velocity and the angle of attack of the wing, as both of these factors increase suction on the top surface of the wing, effectively pulling the actuator upwards. This suggests that additional actuation points on the wing surface, which would modify the pressure distribution, would likely influence the actual displacements of neighbouring actuators. In other words, the position of an actuator would depend not only on the input signal, but also on the entire geometry of the wing. This is a significant issue for multidimensional systems, for instance where piezoelectrics are positioned between points displaced with linear actuators. Long-term repeatability studies suggested that significant discrepancies may appear over time. The authors concluded that this was a result of a drop in capacitance, and that repoling the actuator, i.e. applying a voltage to repolarise it, resolved the issue. It was also suggested that feedback control should be employed in order to maintain the positions of the actuators, mitigating the effects of creep, hysteresis, and any long-term variations of actuator response, although this significantly increases system complexity.

Angelino and Washington [34] employed piezoelectric actuation with stick-slip motion to modify the shape of a satellite antenna's subreflector, in order to alter its radiation pattern. To achieve this, electrical signals were used to slowly extend a piezoelectric stack and rotate the connected screw, followed by fast contraction, which allowed the screw to slip and maintain position due to its relatively high inertia and low friction. This type of motion was necessary, as the small displacements of the piezoelectric stack would not be sufficient to achieve the desired change in shape in continuous motion. Emphasis was placed on a numerical electromechanical model of the actuator, which was deemed essential to predict its response. The resulting surface shape of the subreflector was verified using photogrammetry.

Giurgiutiu [32] presented a review of applications of piezoelectrics to rotor blades for twist and flap actuation. The existing types of piezoelectrics were introduced, with a focus on energy and power density, and the linear piezoelectricity theory was described. Examples of practical applications and experimental tests were also outlined. The author briefly described the technical details of the actuation systems and test results, but concluded that despite the advanced development of piezoelectric technologies, their application in commercial or industrial applications remained challenging.

2.1.2.4 Self-Folding Bilayers

A novel and promising type of actuation are thin bilayers of materials with differing expansion coefficients in response to a stimulus. The concept is based on the principle of residual stress formed at the interface of two materials when one of the materials undergoes expansion or contraction in response to a change in a particular condition, while the other material remains passive, restricting the transformation and leading to bending [36–38]. Whereas the bilayers are usually polymer-based, a range of different material combinations were shown to be applicable, including liquid crystal elastomer-polystyrene [39], polypyrrole-gold [40] and polydimethylsiloxane-silicon carbide [41]. The stimuli activating the bilayers can be changing temperature [37–39, 41, 42] or pH [36], a solvent [36, 43], or an electrical signal [36, 40, 44].

The main advantage of this class of actuation methods is full reversibility [36, 37, 39–41, 43], the lack of hysteresis [41], and the possibility of creating complex 3D shapes, such as tubes, spheres, or cubes, through patterning of the films [36, 38–40, 43], as seen in Fig. 2.7. The radius of curvature can be controlled by varying aspect and thick-

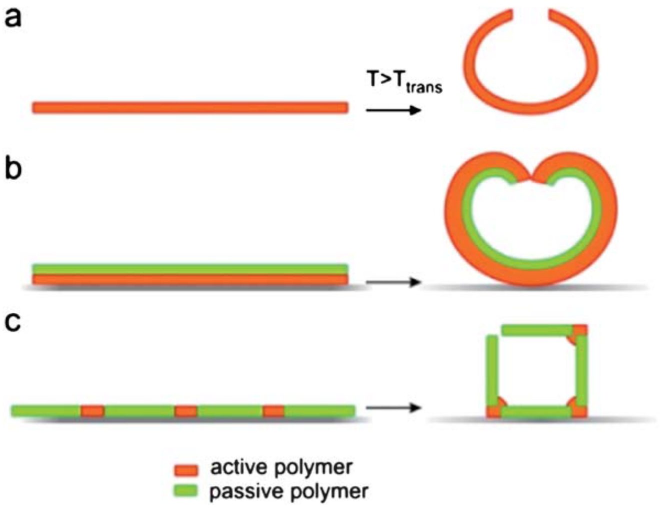


Figure 2.7: Different types of self-folding concepts, including a shape memory polymer (a), a simple polymer bilayer (b), and a patterned polymer bilayer (c) [36].

ness ratios of the layers, although care needs to be taken when selecting the respective thicknesses, as some ratios may cause the bilayer to remain passive due to insufficient or excessive resistance from passive layer [38, 39, 41]. Figure 2.8 shows the thermal response of a self-folding bilayer for a range of layer thickness ratios, where increasing the active layer thickness causes transitions from wrinkling, to folding, to passive response.

Whereas, in theory, the thermostat effect may occur for any two dissimilar materials, controlled bending with a large magnitude is difficult to achieve and requires in-depth understanding of material properties and geometrical characteristics [41]. Reliability of manufacture, which involves advanced chemical and physical processes, is also challenging, even for specialists in the field [36].

In addition, bilayer actuation with significant bending was so far only demonstrated on the microscale (up to 2 mm) [36, 37, 39–41, 43, 44], with suggested applications focused on controlled capture and release of drugs, particles and cells [36, 38, 41, 42]. Issues with wrinkling were also demonstrated [39, 45], and the response and resistance to external forces acting on the bilayers is unknown, which is a significant impediment to their application in aerodynamics.

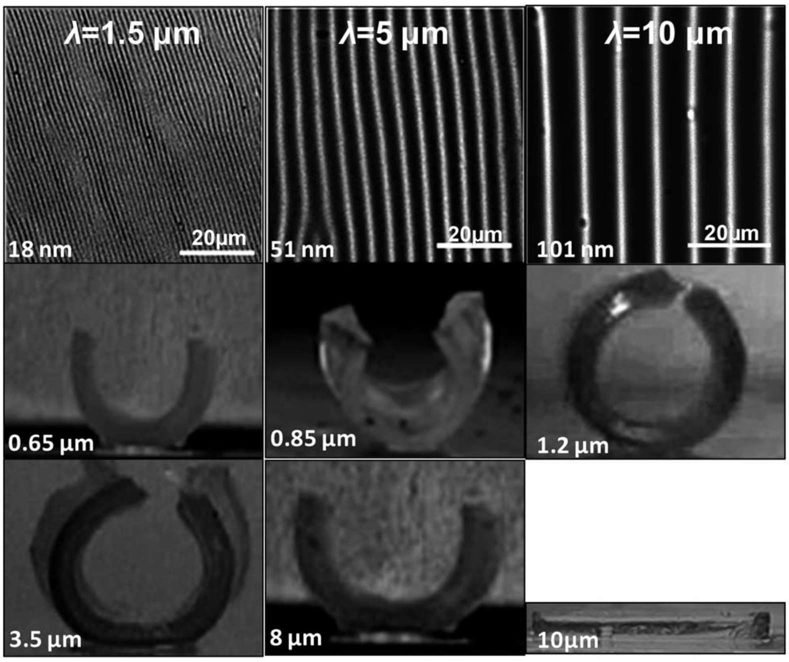


FIGURE 2.8: Thermal response of liquid crystal elastomer-polystyrene bilayers for different polystyrene film thicknesses, as indicated in the bottom left of each image. Liquid crystal elastomer thickness was 480 μm in all cases, and λ denotes wrinkle wavelength for the top three images [39].

2.1.2.5 Electroactive Polymer Actuators

An alternative actuation method—electroactive polymer actuators—was applied by Madden et al. [46] to modify the camber of hydrodynamic propeller blades. This type of actuation was chosen for its high energy and work density, which allowed the actuators to be mounted within the blades. This characteristic is desirable for hardware-in-the-loop aerodynamic optimisation, where many applications may require embedding of actuation systems within a streamlined body, e.g. a thin aircraft wing. Low voltage requirements and zero current draw in the static state are further advantages. However, the authors reported that the forces attained were not sufficient to achieve the desired deflections, which is a hindrance for high-tension flexible systems. Moreover, strains typically do not exceed 2%, and the actuators are comprised of two electrodes that must be submerged in an electrolyte, which poses an additional practical challenge.

2.1.2.6 Zero-Poisson-Ratio Materials

Bubert et al. [18] investigated the design of a material with a zero in-plane Poisson ratio. The authors created a grid of cells, called a pneumatic artificial muscle (PAM) actuator, that could mechanically expand in one direction without contracting in the other, and covered it with a highly extendible Elastomeric Matrix Composite skin. A theoretical model, based on the classical laminated plate theory, was applied to calculate the elastic modulus and other properties of the skin. The PAM actuation system was used to deform a honeycomb, where chordwise rigid beams were connected with V-shaped beams mounted between them, that would expand and contract under in-plane load, effectively making the V's narrower or wider. The force-deflection and stress-strain relationships were determined analytically. The honeycomb was then 3D-printed, as shown in Fig. 2.9, and test results agreed with the analytical predictions.

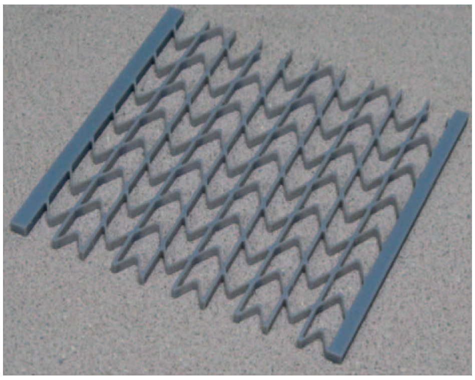


Figure 2.9: A rapid-prototyped zero-Poisson honeycomb by Bubert et al. [18].

Low bending stiffness of the structure was observed, so spanwise carbon fibre rods were added, which stick out of the honeycomb when it is contracted. The authors highlighted the resulting resistance to bending and low friction on the carbon fibre-plastic interface, but did not report extension tests under bending loads, when friction is greatly increased.

2.1.3 Algorithms for Aerodynamic Optimisation

The following section presents a brief introduction to the algorithms used in hardware-in-the-loop optimisation in this work, placing them in the broader context of optimisation methods, followed by an outline of their principles of operation, as well as existing variations and modifications. Algorithm choice was influenced by abundance of existing applications, as well as the algorithms' ability to cope with noise, which is inherent in experimental measurement systems. However, the final selection was dictated by the desire to use algorithm implementations that have been developed, optimised and validated to a similar extent, eliminating the impact of poor coding or unreported modifications. Consequently, only algorithms that were part of MATLAB's Global Optimization Toolbox [47] were used in the experiments.

Crucially, this work is not focused on identification or development of highly-optimised algorithms, but rather on testing a sample of well-understood methods to determine their suitability to wind tunnel-based optimisation problems, and on evaluating and comparing their performance characteristics in relation to their specific attributes. For this reason only individual algorithms were used, and hybrid algorithms, which combine two or more methods, either sequentially or in parallel, in order to improve the overall performance, were not investigated.

One of the simplest methods of numerical optimisation is the steepest descent method, which approximates system behaviour by derivatives and finds the minimum or maximum by gradient inspection [2, 48, 49]. However, there are a number of issues that prevent its application to complex systems, most crucially its poor ability to cope with local extrema, noise, or hysteresis [48]. A closely related alternative is pattern search, which although similar in principle, does not require any gradient information [50–52]. In this method, a set of points is formed at every iteration around the starting point according to a specific mesh, and the one providing the largest improvement in the objective function becomes the next point. The search directions for bound constraint problems are typically orthogonal (normal to constant-variable hyperplanes), in order to ensure convergence, but may also be complemented with directions rotated by 45° [52]. Pattern search was selected as a simple yet effective optimisation method, and is the only non-stochastic algorithm used in the current investigation.

Surrogate-assisted techniques were initially developed in geostatistics, where they were used to identify optimal locations for ore mining, but are also widely used in other fields, including aerodynamic optimisation [49]. They are based on methodical sampling of an area (or a range of solutions) to create what is known as a surrogate, or a low-fidelity model [53]. Good solutions are then readily identified, followed by updating of the surrogate in the vicinity of these solutions. The search and update strategy is then repeated until a satisfactory result is attained [54]. Common surrogate modelling approaches include polynomial response surfaces and Kriging [54]. However, despite being commonly used in aerodynamic optimisation, surrogate modelling was not applied to the current automated system, as it was not available in the Global Optimisation Toolbox at the time of experiment preparation.

Numerous stochastic optimisation algorithms, or ones that utilise randomness, have been developed to tackle complex, multidimensional problems with multiple local extrema, or ones that cannot be differentiated, are non-continuous, or noisy. They are often the best way to find a solution reasonably close to the optimum without examining a substantial number of all possibilities [55]. These techniques include parallel chaos optimisation, particle swarm optimisation (PSO), and the gravitational search algorithm, all of which have applications in engineering [56–60]. However, the type that has been particularly advantageous in aerodynamic optimisation are evolutionary algorithms. Whereas differential evolution and simulated annealing [61–63] are both widely used, genetic algorithms are the most popular [55, 64, 65]. Simulated annealing was selected due to its combination of stochastic evolution and local search characteristics, whereas particle swarm optimisation and genetic algorithms were chosen as two competing, global, population-based methods. Importantly, the GA implementation in the Global Optimisation Toolbox is integer-encoded, which considerably affects the algorithm's behaviour. Unfortunately, MATLAB's implementation of particle swarm optimisation could not be used due to software compatibility issues, therefore a different implementation [66] was employed instead. Its code was analysed thoroughly to ensure consistency with the version included in the Global Optimisation Toolbox and with existing literature [58, 67, 68], thus it is assumed that any observed performance trends are due to the algorithm itself, rather than this particular implementation.

2.1.3.1 Genetic Algorithms

Complex search and optimisation problems can often benefit from an effective use of parallelisation, where many possibilities are explored simultaneously. They may also require the search algorithm used to be adaptive or innovative, i.e. to respond to a changing environment, or to be able to create new solutions, rather than to follow a set of instructions from a fixed starting point. Each of the above arguments justifies the appeal of evolution as an inspiration for computational problem solving and optimisation methods [55]. The process of evolution encompasses the mechanisms through which genes of successive generations of organisms are altered, tending towards an optimal genetic configuration, which constitutes a highly fit organism—an individual well suited to surviving and reproducing in its environment. Crucially, as the fitness criteria for an organism change, evolution follows and adapts new generations to those alterations. Genetic algorithms are a subset of evolutionary algorithms that are based specifically on natural selection and are characterised by three main stages—selection according to fitness, crossover, and random mutation [55, 64].

As genetic algorithms are so closely inspired by natural processes, it is useful to employ biological terminology to describe the various entities and mechanisms of the algorithms. A summary of these terms is presented in Table 2.2.

The functioning of a typical GA is outlined in a flowchart in Fig. 2.10. First, an initial generation is formed, which constitutes creating individuals up to the predefined

Table 2.2: Summary of biological terminology used in reference to genetic algorithms.

Term	Meaning
Generation	Set of individuals at a particular stage of evolution
Population	Individuals within a generation
Genome or individual	Solution defined by a set of variables
Gene; locus	Bit or number within a genome; position of that gene
Genotype	An individual's genetic composition
Phenotype	Characteristics of an individual resulting from its genotype
Mating or reproduction	Pairing of solutions that will produce offspring
Parents	Original solutions from which offspring is derived
Offspring	Members of a subsequent generation formed from their parents
Crossover	Exchange of genetic material during reproduction
Mutation	Perturbations at random loci

population size L, where each individual is comprised of N variables. Each variable is usually, but not necessarily, constrained by bounds, and can be encoded as an integer, a real number, a binary string, etc. The first generation, which forms the starting point for the algorithm, may be initialised either randomly, or through a more advanced sampling method, such as Latin hypercubes[1] [69] or Sobol sequences[2] [70], with the aim of uniform distribution of points in the N-dimensional design space. Next, fitness of the entire population is evaluated using the chosen objective function. When creating the next generation, elitism might be applied first, where a predefined number of fittest individuals are carried through unaltered, to prevent the loss of favourable gene

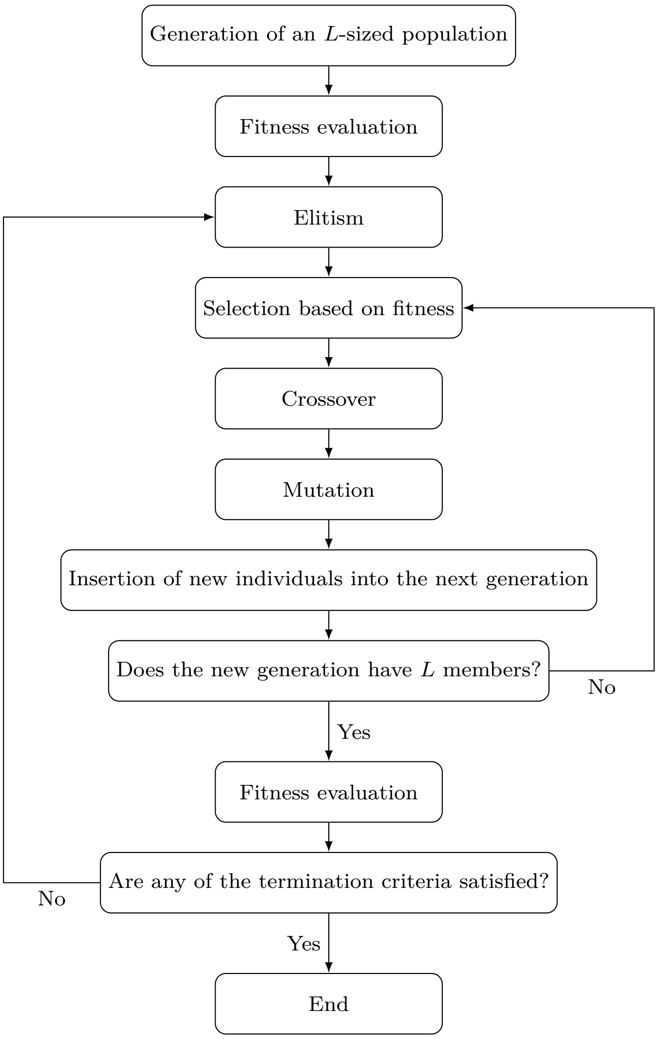

Generation of an L-sized population
Fitness evaluation
Elitism
Selection based on fitness
Crossover
Mutation
Insertion of new individuals into the next generation
Does the new generation have L members?
No
Yes
Fitness evaluation
Are any of the termination criteria satisfied?
No
Yes
End

FIGURE 2.10: Flow chart presenting the functioning of a typical genetic algorithm.

[1] Latin hypercube sampling is a multidimensional extension of the Latin square, where each row, column, and any subsequent hyperplane must contain only one sample.

[2] Sobol sequences are generated deterministically for a given number of dimensions, ensuring minimised empty spaces between samples, and therefore high sample uniformity.

combinations. Then, the GA selects pairs of individuals for reproduction, where an individual's probability for selection is dependent on its fitness. The chosen pair, or mates, are then subjected to crossover, which involves exchanging parts of their genetic material (variables), and subsequently to mutation, where random perturbations are introduced, e.g. bit flips. Both of these operations occur according to their respective probabilities, or ratios, defined by the user. Finally, the population is checked for any members that violate the predefined criteria, which would result in either replacing them or applying a penalty factor, and then is evaluated for fitness, which completes the new generation. The process of elitism, selection, crossover and mutation is then repeated until at least one termination criterion is satisfied. Termination criteria may be expressed in terms of wall-clock time, function evaluation count, fitness threshold, convergence, etc. The key operations and parameters of genetic algorithms are described in greater detail in the following paragraphs, which aids the discussion in Chapter 4.

Population Size The size of the population, denoted by L, is one of the most crucial parameters of genetic algorithms. At low population size GAs tend to be highly prone to errors of decision, and their convergence to be influenced by chance or large numbers of flawed solutions through extensive diversity. On the other hand, a large population size allows the algorithm to work reliably and, due to parallel processing and effective recombination of solutions, quickly solve even complex problems [71]. Population size is usually increased with a growing number of dimensions N, but a trade-off between efficiency and effectiveness needs to be optimised, as increasing L too much effectively slows the evolution without additional benefits [72]. Typical values range from 20 to 1000 [64], and although some equations for the optimal value have been conceived, they are not widely used, as they are not able to fully capture the specificity of optimisation problems [71].

Fitness Evaluation Every individual's probability for selection and reproduction is established through an objective function, which quantifies fitness. This function determines the quality or performance of an individual, and is either maximised or minimised. Fitness is often proportional to a particular property of an individual or its performance, such as force, potential, distance, or energy [55]. In aerodynamic optimisation, common objectives include minimisation of the drag coefficient and maximisation of the lift-to-drag ratio.

Elitism A predefined number of fittest solutions may be retained in each generation in order to prevent them from being either discarded at selection, or altered through crossover or mutation [55]. Although maintaining good solutions in the population is often found to be beneficial, occasionally discarding them to slow the algorithm and allow it to further explore the design space may have a positive influence on efficiency, so a balance between the two approaches should be sought [64].

Selection A broad range of selection methods have been developed since the conception of genetic algorithms, which determine how individuals are selected for reproduction.

- Roulette wheel selection: the original and one of the simplest selection methods. First, each individual is assigned a portion of the roulette wheel equal to the ratio of its fitness to the total fitness of the entire population. Then, selectors are generated randomly, with the probability of hitting an individual proportional to its fitness [64, 72, 73]. However, for small population sizes the outcome is largely dependent on chance, increasing the likelihood of mating weak individuals.
- Stochastic universal sampling (SUS): a slightly modified version of roulette wheel selection, which makes the probabilities of mating less random by selecting from the roulette at even separations, rather than randomly, which corresponds to several, evenly-spaced selectors in the roulette [55, 72].
- Fitness-proportionate selection: individuals are selected at random, and are accepted for mating at a probability equal to the ratio between their fitness and the maximum fitness among the population.
- Tournament selection: a small group of individuals is chosen at random in each mating operation and either the fittest individual is mated, or fitness-proportional selection is performed within the group [72]. This technique is computationally efficient, since it requires a smaller number of fitness evaluations [55, 72, 73].
- Steady-state selection: a few least fit individuals are replaced by offspring resulting from crossover and mutation of the fittest individuals [55].

Fitness Scaling In order to modify the relationship between fitness and probability for selection, fitness values may be scaled according to a specific formula.

- Rank, or linear scaling: each individual's probability of mating is proportional to its position in the ranking of the population according to fitness, rather than to fitness itself. Consequently, a situation in which the majority of offspring originates from a small group of highly fit individuals is unlikely to occur [55, 64, 72, 73].

- Boltzmann scaling: the scaling coefficient varies during the optimisation process, usually decreasing with time. The fitness distribution is brought closer together in early optimisation stages for increased diversity, and is pushed further apart in the latter generations, in order to increase selection pressure [64].
- Exponential scaling: a modification of rank scaling, where scaled fitness is proportional to rank position raised to a constant exponent.
- Top scaling, where a fixed number of the fittest individuals are scaled equally.

Crossover Arguably the most distinctive feature of genetic algorithms is crossover, or exchange of genetic material between offspring. The operation takes place according to the crossover ratio, which is equivalent to the probability that crossover should take place for a particular pair of offspring [55, 64, 72].

- Single-point crossover: a single crossover point is chosen at random, and all bits or variables after this locus are swapped between the two offspring.
- Two-point crossover: genetic material between two randomly selected loci is exchanged. This method often increases the effectiveness of recombination of genes.
- Parametrised uniform crossover: crossover happens at each locus with a predefined probability. This method has no positional bias, which means that any combinations can be formed, or re-formed, and the endpoints are not always exchanged.
- Heuristic crossover: a single offspring is created on a straight line passing through its two parents in the search space, a small distance away from the fitter parent in the direction away from the less fit parent, as illustrated in Fig. 2.11.
- Arithmetic crossover: if the genetic algorithm is encoded with integers or real numbers, crossover may be achieved by averaging the positions of the parents in the search space.

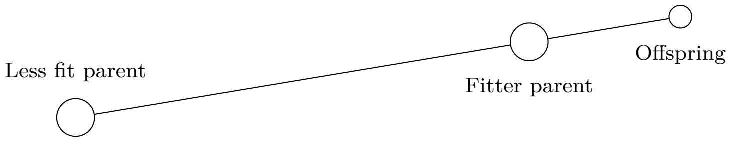


Figure 2.11: Illustration of heuristic crossover in genetic algorithms.

Mutation The operation that prevents the loss of diversity in a population is mutation [64]. In the simplest binary form, it causes every bit in each of the offspring to flip according to the mutation rate, or mutation probability, which is typically between 10^{-4} and 10^{-2} [64, 74]. When the GA is real-number-encoded, mutation might cause replacement of a given variable by a random number within its bounds, or by a number from a Gaussian distribution centred on the parent's position. Mutation may also be biased towards less fit individuals, towards early generations, or used only after a certain generation has been reached to slowly alter the fittest members, but only if the bit flips are constructive, i.e. increase maximum fitness [64].

Termination The algorithm is usually terminated when a predefined number of function evaluations have been carried out, when a solution above a certain fitness value has been found, or when the diversity of the population has fallen below a preset threshold [64, 72]. The latter requires a definition of population diversity, which could relate to fitness, genotypes, or phenotypes [72].

2.1.3.2 Particle Swarm Optimisation

Particle swarm optimisation is another stochastic method that operates on a population of solutions, but moves them around the design space, instead of manipulating the position description (genetic content), as is the case in genetic algorithms. This method aims to simulate the spatial behaviour of a bird flock or a fish school, where all individuals are aware of the actions of the entire swarm, in order to find the region of maximum performance (food availability) in the search space [58], as outlined in Fig. 2.12. Crucial for the algorithm are the positions and velocities of the particles, both of which are generated randomly for the initial swarm. In each iteration, the algorithm evaluates the fitness of all the particles, and then adjusts their velocities and positions, pulling them simultaneously towards their previous personal best and towards the current global best design. Velocities are updated according to the following formula:

$$\hat{\mathbf{v}} = \mathbf{v} + k_s \cdot R_s \cdot (\mathbf{p}_{g,\text{best}} - \mathbf{p}_{\text{current}}) + k_c \cdot R_c \cdot (\mathbf{p}_{p,\text{best}} - \mathbf{p}_{\text{current}}) \,, \tag{2.1}$$

where $\mathbf{v}$ and $\hat{\mathbf{v}}$ are the current and updated velocities of a given particle, k_s and k_c are the social and cognitive factors (determining the influence of the global and personal maxima respectively), R_s and R_c are random numbers between 0 and 1, $\mathbf{p}_{\text{current}}$ is the current position of the particle, and $\mathbf{p}_{g,\text{best}}$ and $\mathbf{p}_{p,\text{best}}$ are the positions of the current global and personal maxima respectively. This strategy results in coherent and efficient movement towards the optimal regions, while sufficient exploratory behaviour is maintained through simple stochastic parameters. The social and cognitive factors k_s and k_c are commonly assigned values of 2, giving the resulting factors $k_s \cdot R_s$ and $k_c \cdot R_c$ an average value of 1, which ensures an appropriate balance between slow, conservative motion towards the target, and risky, exploratory movement [58]. Furthermore, swarm size is a key factor for the performance of particle swarm optimisation. Typical values range from 10 to 60 [67, 75], but larger sizes may be desirable for highly multidimensional problems [76].

A later modification to the globally-oriented algorithm was the addition of inertia weight w, which makes velocity dependent on its previous values, in an attempt to balance the global versus local nature of the search [77]. Increasing this parameter, which can be either a positive constant or a positive function of time, results in more exploratory behaviour, whereas decreasing it has the opposite effect [75]. Early tests

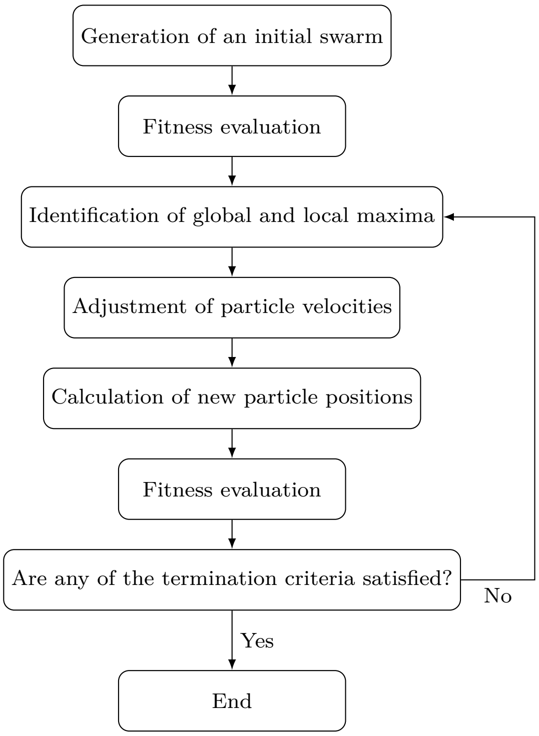

Generation of an initial swarm
Fitness evaluation
Identification of global and local maxima
Adjustment of particle velocities
Calculation of new particle positions
Fitness evaluation
Are any of the termination criteria satisfied?
No
Yes
End

FIGURE 2.12: Flow chart presenting the functioning of particle swarm optimisation.

showed that values of w between 0.9 and 1.2 yield the highest chance for finding the global optimum, and that gradually decreasing the parameter over the course of a run provides the best effectiveness and efficiency [77, 78]. Furthermore, it was found that progressively decreasing inertia weight eliminates the need for a particle velocity limit, which is otherwise required to control the behaviour of the swarm [78]. However, it was later reported that the desirable progression of inertia weight is problem dependent, as exceedingly rapid loss of the exploratory characteristic of the algorithm may be detrimental for large and complex problems [76].

An alternative, locally-oriented paradigm was also implemented, where instead of the global maximum, particles are only aware of the local maximum in their neighbourhoods of a certain number of particles. Early tests showed that whereas the local method is less efficient in finding the global maximum, a variant with a neighbourhood of 2 is more resistant to local maxima [67].

2.1.3.3 Simulated Annealing

Optimisation by simulated annealing, which utilises the connection between statistical mechanics and combinatorial optimisation, was first devised by Kirkpatrick et al. [61]. The state of any material is governed by its temperature, or the local average kinetic energy of its particles, but the exact arrangement of atoms or particles is, to an extent, random. The probability for any given state (set of atomic positions) is described by the Boltzmann probability factor, $\exp(-E(\{r_i\})/k_BT)$, where $E(\{r_i\})$ is the energy of the system described by parameters r_i, k_B is Boltzmann's constant, and T is temperature. Crucially, at low temperatures the probability distribution collapses to the lowest energy states, therefore in order to find the ground state of a matter, its temperature must be lowered, which corresponds to annealing. First, the material is heated to above its recrystallisation temperature, followed by gradual cooling, ensuring that it remains in quasi-equilibrium, in order to avoid defects and maintain stability. These principles may be readily applied to any optimisation problem, where the energy of the system, $E(\{r_i\})$, is replaced by a fitness function, and the state of matter r_i becomes the set of parameters to be optimised.

The procedure of simulated annealing, as displayed in Fig. 2.13, begins with the generation of an initial configuration, which corresponds to a point in the design space, and then a randomised disturbance is applied. The magnitude of the offset can be either selected from a uniform distribution [61–63], or based on a probability distribution with a

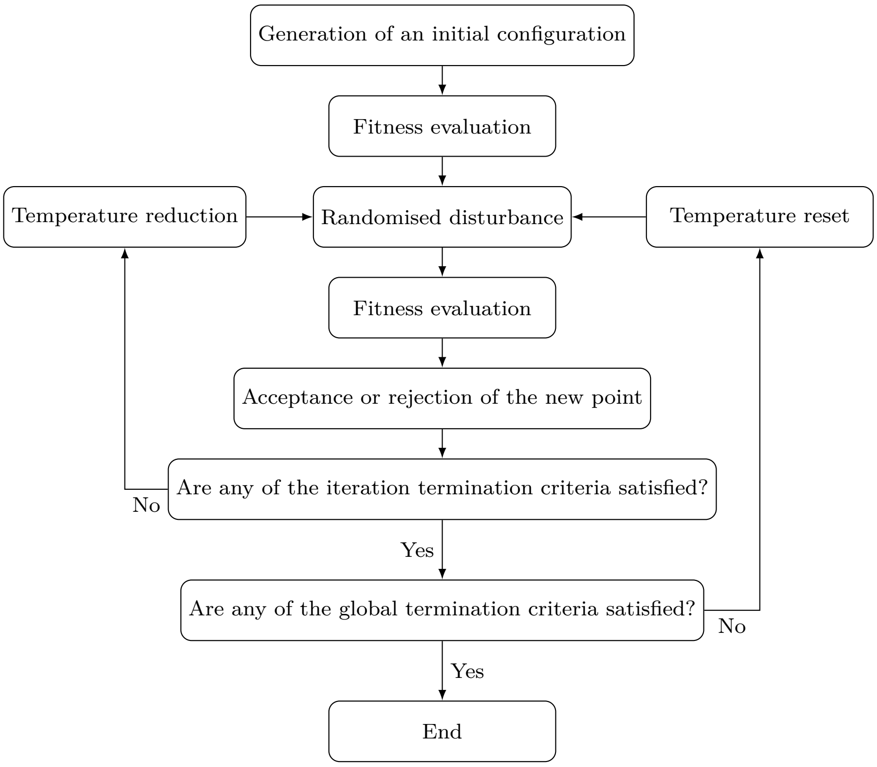

Generation of an initial configuration
Fitness evaluation
Temperature reduction
Randomised disturbance
Temperature reset
Fitness evaluation
Acceptance or rejection of the new point
Are any of the iteration termination criteria satisfied?
No
Yes
Are any of the global termination criteria satisfied?
No
Yes
End

FIGURE 2.13: Flow chart presenting the functioning of simulated annealing.

scale proportional to temperature, i.e. the higher the temperature, the greater the magnitude of the disturbance. Next, the new energy (fitness) of the system is determined; if an improvement is made, the new configuration is accepted as the next starting point. However, in order to prevent the system from premature convergence, adverse changes are also occasionally accepted, at a probability equal to $\exp(-\Delta E/k_B T)$ for fitness maximisation problems—the larger the increase in energy ΔE (or decrease in fitness), and the lower the temperature, the smaller the chance of accepting the new point. An annealing schedule defines how quickly temperature is decreased, which determines how fast the algorithm narrows down its search. A number of schedules have been proposed, including:

- Exponential [61, 62]: $T_n = T_0 \cdot k^n$, where T_n is the temperature after n iterations, T_0 is the initial temperature, and k is a constant ($k < 1$);
- Boltzmann: $T_n = T_0/\ln(n)$, where $\ln(n)$ is the natural logarithm of n;
- Fast: $T_n = T_0/n$;
- And adaptive cooling, proportional to a heuristic estimator of the global maximum [63].

Temperature may be updated every time a new point is generated, or only when a new global maximum has been found. Typically, once sufficient convergence has been achieved, or the temperature has decreased to a limit, the temperature is raised and a new iteration begins, which corresponds to reheating the material and reannealing it.

2.1.3.4 Pattern Search

Pattern search is a type of gradient-free, local optimisation algorithms. The first of the kind—direct search—was introduced by Hooke and Jeeves [50]. Generalised pattern search, which forms a subgroup of direct search methods, was originally developed for unconstrained problems [51], and later extended to bound constrained and linearly constrained problems [52, 79]. The functioning of a typical pattern search algorithm is outlined in Fig. 2.14. Fundamentally, in each iteration the algorithm places a pattern of points around the initial location, and evaluates their fitness [51]. Whereas the shape of the pattern is dependent on the implementation, a general rule is that the steps may not be arbitrary. If a point is found that improves the objective function, it becomes the starting position for the next iteration; otherwise, the starting point remains the same. A key characteristic of this type of algorithm is the way the step length (i.e. the scale of the pattern) is updated. Following a successful iteration (one that yielded an

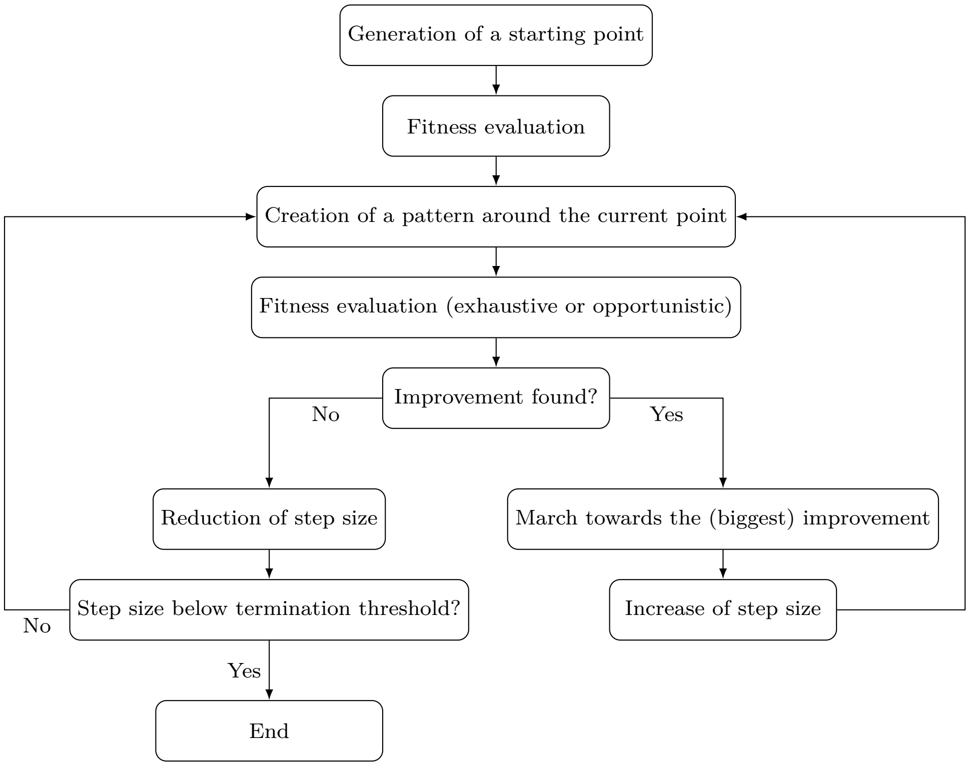

Generation of a starting point
Fitness evaluation
Creation of a pattern around the current point
Fitness evaluation (exhaustive or opportunistic)
Improvement found?
No
Yes
Reduction of step size
March towards the (biggest) improvement
Step size below termination threshold?
Increase of step size
No
Yes
End

FIGURE 2.14: Flow chart presenting the functioning of pattern search.

improvement), the step size may increase, but is not allowed to decrease. Conversely, if no improvement has been found, the step size is forced to decrease [51]. A common practice is to multiply the step size by a constant factor k, where $k > 1$ after a successful iteration, and $0 > k > 1$ after an unsuccessful iteration.

Different strategies exist for polling, or finding an improvement in a given pattern. One possibility is to evaluate fitness at all points in the pattern (exhaustive search), but it might be more efficient to only evaluate points until one improvement is found (opportunistic search) [80]. Search directions are typically orthogonal, i.e. normal to constant-variable hyperplanes, in order to ensure convergence. This is also used in bound constraint problems, and results in $2N$ points per iteration for an N-dimensional problem. However, these may be complemented with 45° directions, which yields $3^N - 1$ points per iteration [51, 52]. For linearly constrained problems, search directions in proximity to active constraints are set to be parallel and normal to the constraint hyperplanes [79]. In addition, for any constrained problems, the step size must be such that feasibility is maintained (i.e. that the search does not exceed the bounds).

2.1.3.5 Applications and Comparative Tests

Obayashi and Tsukahara [81] presented the results of a comparison between a gradient-based algorithm, simulated annealing, and a genetic algorithm. They tested these algorithms on a typical aerodynamic problem—lift maximisation of an airfoil. The airfoil shape was defined by a weighted sum of four different airfoils, where the weights were optimised by the algorithms, and the panel method was used for lift evaluation. The authors concluded that the gradient-based algorithm is unlikely to perform well, unless the optimum is in the neighbourhood of the initial point, indicating sensitivity to starting position. Simulated annealing was also found to converge on a local maximum, whereas the genetic algorithm, despite requiring 6 times more iterations, was the only algorithm from the group to attain the global maximum. Furthermore, because both simulated annealing and gradient-based search are sensitive to starting points and need several trials for comparison, the authors concluded that the GA was the best method for their problem. The authors also noted that genetic algorithms would be particularly useful for transonic and supersonic problems, due to their global nature and the ability to cope with discontinuities in performance, such as those caused by shock waves.

A real-number-encoded genetic algorithm was applied to a range of problems by Holst and Pulliam [82], including the optimisation of a 3D transonic wing using a potential flow solver. Their algorithm was atypical for several reasons. Firstly, it operated on real number variables, rather than on binary strings. Secondly, only one of the four GA operators (elitism, crossover, perturbation or mutation) was applied to each individual, i.e. each operator created a fixed proportion of every new generation. The authors investigated the effect of changing operator significance and population size, noting that elitism ratios of 10% or less and small population sizes led to quicker convergence in terms of computational time. Adding a symmetry constraint for the airfoil or changing the initialisation of CFD simulations were also considered, with the aim of reducing the computational effort. The authors concluded that the GA provided satisfactory convergence and robustness, but noted that gradient-based methods are likely to be superior for this class of problems, and that the benefits of a GA would be more pronounced for a non-smooth design space or for multi-objective optimisation.

Wetter and Wright [83] used a building energy consumption optimisation problem to compare the performance of the Hooke-Jeeves pattern search algorithm and a genetic algorithm. The task involved numerical optimisation of window area, window overhang size, set points for external shading devices, and temperature thresholds for night cooling for a simulated office floor at three locations at different latitudes. Both algorithms yielded significant improvements over the baseline case in terms of total annual energy consumption. In two of the three cases, the genetic algorithm demonstrated superior performance, achieving 37% and 50% larger improvements than the Hooke-Jeeves algorithm. The authors attributed this difference to the discontinuities present in the fitness function, which pattern search algorithms are not able to overcome. In the final test case, both algorithms demonstrated performance within 1% of each other, and the authors concluded that although both methods are good candidates for solving similar problems, a hybrid algorithm may be a more efficient alternative. A proposed hybrid method would first perform a global search using a GA to ensure that the search is not trapped in a local optimum, and then carry out a fine, localised search around the global optimum using a pattern search algorithm.

Kirkpatrick et al. [61] applied simulated annealing to circuit design optimisation and the travelling salesman problem, reporting significant improvements even for very large problems, and highlighting the advantage of the adaptive procedure, where the scope of the search is gradually narrowed. The authors also noted that best results were obtained when computational effort, determined by the annealing schedule, scaled with the number of degrees of freedom or its small power. Furthermore, Kirkpatrick [62] performed a quantitative study on the performance and efficiency of simulated annealing, comparing it with the greedy sorting algorithm on the travelling salesman problem, reporting not only improved performance, but also reduced computational cost for sufficiently large samples. Two different types of permutation perturbations were tested—reversal and displacement of sections—but their efficiency was not quantitatively compared. Suggestions for optimising the annealing schedule were also presented.

A review of methods for parallelisation of simulated annealing was presented by Wang and Damodaran [84], along with the results of the application of their methods to several aerodynamic optimisation problems. The design cases were optimisation of an axisymmetric diffuser, one wall of a supersonic nozzle, and a supersonic axisymmetric nozzle, and the algorithm was found to obtain the desired pressure distributions and maximise nozzle thrust by optimising the cross-sectional area distributions. In addition, a near-linear time saving due to parallelisation was observed, with 30 processors reducing wall-clock time by a factor of up to 20. However, it was also noted that efficiency improvements are negligible above a certain number of processors.

2.2 Combinatorial Optimisation

2.2.1 Travelling Salesman Problem

The travelling salesman problem is a popular heuristic combinatorial optimisation problem, where a number of locations on a two-dimensional plane, e.g. cities on a map or components on an electronic circuit board, are to be connected with a line that passes through all the points at the lowest possible cost [85–88]. In the classical TSP, the cost is directly proportional to the geometric distance between the points. However, numerous variants of the TSP exist, where the cost function is modified to suit particular applications. Examples include asymmetric TSP, where the cost of travel between any two points depends on the direction of travel [89], time-dependent TSP, where the cost also depends on the position of the pair in the sequence [90], multi-TSP, where all points must be visited in several sub-tours, possibly with limited capacity per tour [87], or the dial-a-bus problem, where the order of several two-point deliveries must be optimised [87].

One such variant is the Chebyshev TSP (CTSP), where the cost metric is evaluated as the maximum of the absolute values of differences of the points' x and y coordinates [88, 91, 92]. This variant is useful for situations where multiple dimensions, or degrees of freedom, can be traversed simultaneously. Whereas the highest practical number of dimensions in the classical TSP is 3, as it involves geometric distances in physical domains, the CTSP is not bound by this constraint, as the points to be connected can represent the state of any system with multiple DoFs. A real-life example of the multidimensional CTSP is a robotic arm with multiple joints, whose tool has to go through a set of spatial positions and orientations. In this case, every degree of freedom of the arm (each of its linear and rotary motors) represents a dimension of the CTSP problem, and their number can exceed 3.

Another application, which was the motivation for this research and is discussed further in Chapter 4, is hardware-in-the-loop aerodynamic optimisation, where the objective function is evaluated using a wind tunnel model with a variable geometry, position or orientation. Most studies on this type of optimisation utilised genetic algorithms, which operate on populations of configurations. Once a set of solutions is generated, the system has to evaluate fitness for each one before creating a new generation. Crucially, the order in which the solutions are tested is inconsequential for the optimisation method, although it may impact the readings due to aerodynamic hysteresis. This can be exploited by evaluating the population in an order that minimises the total time spent on actuating the system. The time needed to transform the model between two configurations is determined by the actuator that takes the longest to complete its motion, therefore the Chebyshev cost metric is applicable. The number of degrees of freedom of the system is equal to the number of independent actuators, and is in principle unlimited. As the travelling salesman problem, as well as the Chebyshev variant, have been investigated primarily in 2 dimensions, there is a need to explore how optimisation effectiveness changes for higher numbers of dimensions.

2.2.2 Sorting Algorithms

Crucially, the study presented in this work does not focus on algorithmic developments, or on identifying optimal sorting methods for particular cases. Instead, it investigates the impact of the number of DoFs and the number of points to be sorted on the effectiveness of approximating the CTSP, utilising a range of well-established, readily available algorithms. They include open-source implementations of a genetic algorithm [93], simulated annealing [94], ant colony optimisation (ACO) [95], and tabu search [96], as well as two algorithms written specifically for the purpose, viz. greedy sorting and another genetic algorithm variant. The principles of operation of these methods are briefly outlined below.

2.2.2.1 Greedy Sorting Algorithm

Greedy sorting is initialised at a predefined or random starting point, and in each step the algorithm moves to the nearest unvisited point in the design space. Although the final order of points depends on the starting point, the computational cost of this algorithm is small enough that in most cases it is affordable to initialise it from all points and select the best solution. Despite that, this method is unlikely to find the global optimum in large problems, as always travelling to the nearest neighbour may result in long distances between remaining points towards the end of sorting, whereas occasionally travelling to a slightly more distant point may result in an overall shorter route. However, adequate solutions can be obtained at a very low cost, making greedy sorting a good reference for more advanced algorithms.

2.2.2.2 Sorting Genetic Algorithm

The sorting genetic algorithm uses typical GA operations, i.e. selection, crossover and mutation, as described in detail by Mitchell [55]. This implementation uses a population size of 50, where each individual corresponds to one permutation of design points. Fitness-proportionate selection is used, where individuals are picked randomly and the probability of acceptance for mating is equal to the ratio of the lowest fitness (Chebyshev cost) in the population to its own fitness (100% for the fittest individual). Crossover is applied to pairs of selected individuals with a probability of 40%. In this operation, sections of random length and position are selected from each individual and placed into the other in the same position, with the remainder of points in both individuals repositioned around the inserted section, maintaining their initial order. Mutation is then applied to each selected individual at a probability of 15%, and swaps two randomly selected points within the permutation. Elitism is also applied, which carries through one best individual to the next generation.

2.2.2.3 TSP Genetic Algorithm

The second GA, developed by Kirk [93], operates on groups of four permutations at a time, from a population of 100. The algorithm takes the first four permutations, picks the best of the group (best-of-four), generates two random integers i and j between 1 and the length of the permutation L, and creates four offspring according to the following operations:

- The first offspring is a direct copy of the best-of-four;
- The second offspring is a copy of the best-of-four, with the order of the fragment between i and j reversed;
- The third offspring is a copy of the best-of-four, with points i and j swapped;
- And the fourth offspring is a copy of the best-of-four, with point i moved to position j, and points $i+1$ to j shifted to positions i to $j-1$.

This process is then repeated for individuals 5 to 8, 9 to 12, etc., until the offspring population is completed.

2.2.2.4 TSP Ant Colony Optimisation

The TSP ant colony optimisation algorithm by Yarpiz [95] simulates the behaviour of a colony of ants searching for food. Its main principle is that ants deposit pheromones as they travel, and the more pheromone an ant can sense, the more likely it is to follow its scent. A key element is pheromone evaporation, which means that the longer the path, the more time pheromones have to evaporate, and so the less likely ants are to follow it due to the reduced pheromone intensity [95]. The algorithm used a population size of 40, and an evaporation rate of 0.05.

2.2.2.5 TSP Simulated Annealing

The TSP simulated annealing implementation was developed by Seshadri [94], and is based on similar principles to the original algorithm described by Kirkpatrick et al. [61]. Working on a single permutation, the algorithm randomly swaps a predefined number of pairs of points in each iteration. If the swaps result in a reduction in cost C, then the new permutation is accepted. If the swaps result in an increase in cost, then the new configuration is accepted at a probability equal to $\exp(\frac{-|\Delta C|}{T})$, where ΔC is the change in cost due to swapping, and T is the current temperature. Temperature is initialised at 2000, and reduced by 3% after every 10 successful iterations. The number of pairs of points to swap is initialised at 2, and updated after each successful iteration according to Eq. 2.2:

$$\hat{n}_{\text{swap}} = \text{round}\left(n_{\text{swap}} \cdot \exp\left(\frac{-|\Delta C|}{n_{\text{it}} \cdot T} \right) \right), \tag{2.2}$$

where n_{swap} and $\hat{n}_{\text{swap}}$ are the original and updated numbers of pairs of points to swap, and n_{it} is the number of completed iterations that were accepted.

2.2.2.6 Tabu Search

Tabu search is a local search method that, in each iteration, applies a predefined number of operations to the current permutation. The implementation developed by Yarpiz [96] uses three types of operations for two points within the permutation, i and j:

- Swap, where points i and j are swapped;
- Reversion, where the order of points between i and j, including the endpoints, is reversed;
- And insertion, where point i is moved to position j, and points $i+1$ to j are moved to positions i to $j-1$.

The operations are carried out for every valid pair of points within the permutation. For swap and reversion, j must be greater than i, as the pairs i & j and j & i result in identical operations, and in addition, for reversion, the difference between j and i has to be greater than 2. For insertion, the pairs i & j and j & i result in different operations, so both are permitted, but the difference between j and i has to be greater than 1. As a result, an L-sized permutation will undergo $\frac{L(L-1)}{2}$ swap operations, $\frac{(L-1)(L-4)}{2}+1$ reversion operations, and $(L-1)(L-2)$ insertion operations, or a total of $n_{\text{op}} = 2(L-1)(L-2)+1$ operations in every iteration. Furthermore, the algorithm does not permit reusing operations (swap, reversion or insertion for particular i and j) that resulted in an improvement of the global optimum within $n_{\text{op}}/2$ iterations, in order to discourage the search from returning to already visited configurations.

2.3 Automotive Diffuser Aerodynamics

This section presents a review of literature on automotive underbody diffusers, supporting the discussions on diffuser optimisation and aerodynamics presented in Chapters 4 and 6 respectively. It begins by placing diffusers in the context of race car aerodynamics, followed by an outline of how their performance and underlying flow physics were gradually understood through academic studies. The review is intertwined with explanations of how various factors affect diffuser aerodynamics, including ride height, diffuser and rake angles, ground simulation, and particular geometry modifications.

2.3.1 Historical Context

The potential of car underfloors for downforce generation was first realised in the 1960s. Some of the first prototypes, notably Chaparral 2J (1969) and Brabham BT46B (1978), involved suction fans drawing air from beneath the car. The sides of these cars were sealed using flexible skirts, hence large regions of low pressure were generated underneath [97]. As this solution gave the cars an immense advantage in their respective competitions (Can-Am and Formula 1), it was shortly outlawed by both regulations [97]. The first Formula 1 car to utilise passive ground effect was Lotus 78 in 1977, which used two small aspect ratio inverted wings as sidepods. These, again accompanied by sliding skirts in order to seal the flow under the floor, generated large amounts of downforce compared to flat underfloors, which allowed the car to win the world championship [97]. After sliding skirts were banned for safety reasons and flat-bottomed underfloors became mandatory [98], renewed interest in employing ground effect into car design appeared and the concept of a diffuser—an upswept section of the underfloor at the rear of the car—was introduced for the first time.

2.3.2 Principles of Downforce Generation

Cooper et al. [99], the authors of one of the most comprehensive studies on automotive diffuser aerodynamics, established through wind tunnel testing that a diffuser-equipped, flat-roofed Ahmed body [100] in ground effect generates downforce through three interrelated mechanisms [99]:

- Surface upsweep, which effectively cambers the body, resulting in a downward force.
- Ground interaction, where the flow underneath the body is accelerated due to the ground constraint and its static pressure decreases. The magnitude of the pressure reduction increases with decreasing ground separation, up until a critical ride height, where viscous forces become dominant and the effect is reversed.
- Diffuser pumping, where the increasing cross-sectional area results in slowing the airflow down and increasing its static pressure. Crucially, in the case of a bluff body or a car with a fixed base pressure at the diffuser exit, the pressure recovery manifests itself as a suction peak at the diffuser inlet, which propagates upstream, towards the front of the body [99, 101–105]. This phenomenon is illustrated in Fig. 2.15.

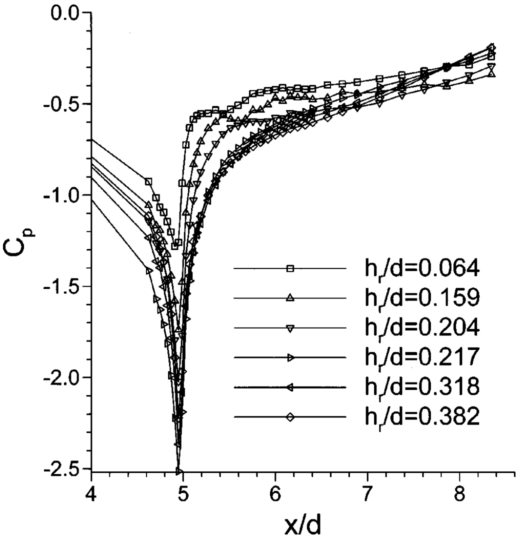


FIGURE 2.15: Surface pressure coefficient along diffuser centreline at a range of non-dimensional ride heights. Diffuser inlet at $x/d = 4.95$ [102].

A key milestone in the understanding of ground effect diffuser flows was the discovery, aided by flow visualisation, of counter-rotating vortices near the side edges of the diffuser [106]. It was shown that these vortices, depicted in Fig. 2.16, not only help to prevent or delay flow separation at the sharp diffuser inlet edge [103, 105–109], but also directly contribute to downforce generation by inducing low-pressure regions on the sides of the diffuser surface [102, 105, 107, 110]. The vortices were also shown to grow in size while moving inboard and upwards as they propagate downstream through the diffuser channel, and in some cases to detach from the diffuser surface [104, 109, 111, 112].

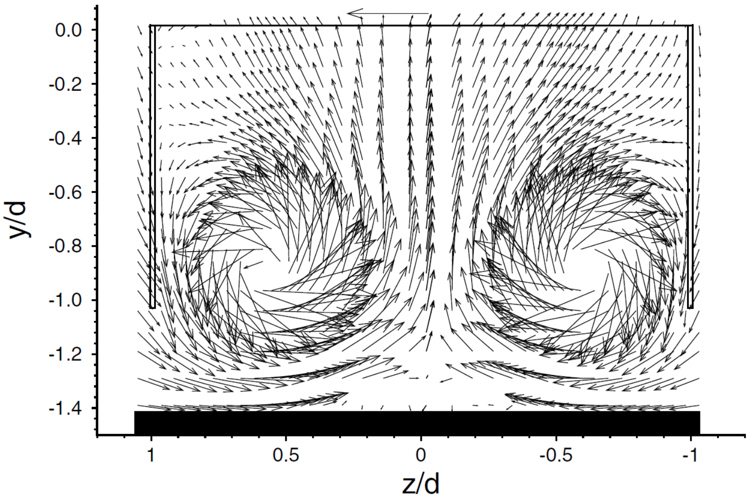


FIGURE 2.16: Mean cross-flow velocity vectors across a cross-section of the diffuser, measured using laser Doppler anemometry at $x/d = 8.476$ (15mm downstream of the model), at a non-dimensional ride height of $h_r/d = 0.382$ [110].

2.3.3 The Impact of Ride Height and Diffuser Angle

Further experiments and CFD simulations confirmed that the effect of reducing the ride height of a diffuser-equipped bluff body in ground effect is accelerating downforce increase [99, 102–105, 110, 111, 113–115]. This is mainly the result of increased pressure recovery, which results in a stronger suction peak at the diffuser inlet, as shown in Fig. 2.17, and stronger vortices [102–105, 107, 110, 114]. However, Cooper et al. [99] and Jowsey and Passmore [114] also observed a sharp change in the streamwise pressure recovery rate in the vicinity of the leading edge of the underfloor. At high ground separations, the pressure recovery rate changed suddenly, as seen in Fig. 2.17 around ports 12 and 13. This was caused by a separation bubble, which occurred as the inertia of the flow prevented it from following the curvature of the nose. As ride height was reduced, the favourable pressure gradient under the nose increased, diminishing the localised separation. This resulted in more gradual pressure recovery in that area, and therefore reduced pressure under the front half of the underfloor and increased net downforce.

As ride height is reduced further, the increasing adverse pressure gradient in the diffuser causes the flow to separate. However, the counter-rotating vortices prevent separation near the sides of the diffuser, and help to reattach the separated flow in the centre of the diffuser further downstream, resulting in a localised separation bubble. Simultane-

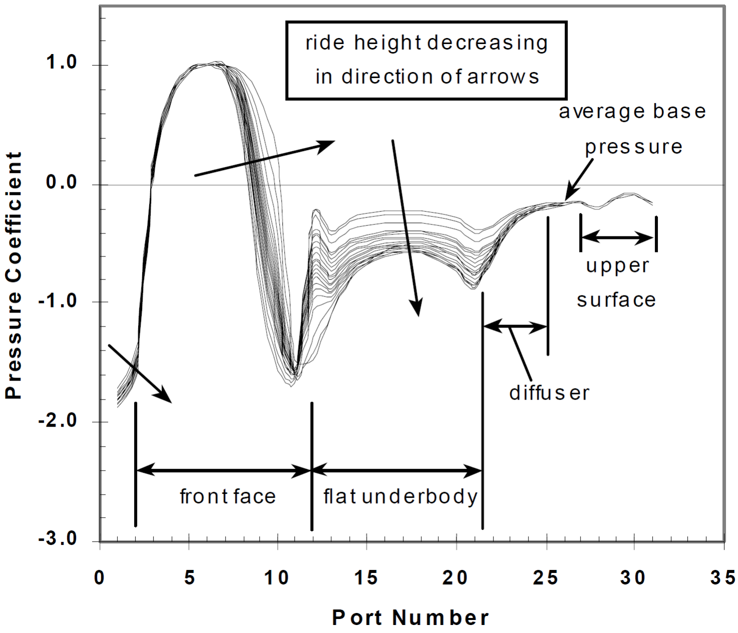


FIGURE 2.17: Pressure distributions along model centreline with varying ride height, at $\theta = 9.64°$ [99].

ously, an onset of vortex breakdown occurs, characterised by increased vortex size, and substantially reduced axial and cross-flow velocities and vorticity [110]. These changes cause a reduction of the downforce enhancement rate, or a small decrease in downforce, jointly referred to as the downforce plateau [102–105, 110, 111, 114].

At even lower ride heights, downforce continues to increase despite the presence of the separation bubble, which gradually moves towards the diffuser inlet. At the critical ride height, the separation bubble is swept to one side, causing asymmetric vortex breakdown and flow separation at the inlet over a substantial part of the diffuser width, resulting in a large recirculation region and a significant loss of downforce [102–105, 107, 110, 111]. Ehirim et al. [104, 105, 112] also observed that the direction of the asymmetric stall depends on the relative strength of the vortices prior to breakdown, with the stronger vortex surviving the subsequent stall. Furthermore, Ruhrmann and Zhang [103], Zhang et al. [110] and Ehirim [116] observed that the process of downforce loss is subject to significant hysteresis at diffuser angles of 15° and above, as the vortices re-form and reattach the flow at higher ride heights than when they break down.

Cooper et al. [99] initially attributed this downforce loss to boundary layer merging between the underfloor and the ground, assuming that the phenomenon would not occur in inviscid flow, and therefore must be the result of viscous effects. This theory was later disproven, as it was shown that Reynolds number has no effect on downforce variations with ride height or on the critical ride height [102, 116]. However, Senior and Zhang [102], Ruhrmann and Zhang [103] and Zhang et al. [110] later suggested that the boundary layer merging phenomenon is indeed present, but at ground separations even lower than where the downforce loss occurs. Both conclusions were later confirmed by laser Doppler anemometry (LDA) measurements of velocity profiles between the underfloor and the ground, which revealed a region of constant velocity between the boundary layers, even at ride heights where asymmetric vortex breakdown and downforce loss have already occurred, followed by apparent boundary layer merging at even lower ground separation [104, 105].

The accelerating downforce increase, the initial downforce reduction (only at diffuser angles above 10°), and the subsequent downforce loss can be seen in Fig. 2.18, which presents downforce coefficient curves with respect to non-dimensional ride height at a range of diffuser angles, as obtained by Cooper et al. [99]. These curves, as well as those obtained experimentally by Howell [117] and Ruhrmann and Zhang [103] and computationally by Knight et al. [115], also show that, as the diffuser angle is increased, the initial downforce reduction occurs at progressively higher ride heights. Importantly, ride height and diffuser angle jointly determine the diffuser area ratio, which is defined as the ratio of cross-sectional areas of the outlet and the inlet of the diffuser. This non-dimensional parameter directly controls the pressure recovery, and hence the adverse pressure gradient in the diffuser. Therefore, as the diffuser angle is increased, so is the area ratio and the resulting adverse pressure gradient, causing flow separation at a higher ride height [103, 111]. At diffuser angles below 10°, the adverse pressure gradient is small enough that the separation bubble does not form, and downforce keeps increasing until the asymmetric stall.

Vitally for automotive applications, wind tunnel measurements showed that the diffuser-associated drag penalty was minimal [106]. This is explained partly by ground effect and partly by the diffuser pumping phenomenon, both of which induce low pressure mainly under the flat section of the underfloor, which is parallel to the freestream and therefore does not contribute to drag. In addition, upward deflection of the flow at the back of the body may reduce the thickness of the wake, which in some cases leads to an overall reduction in drag force [99, 117].

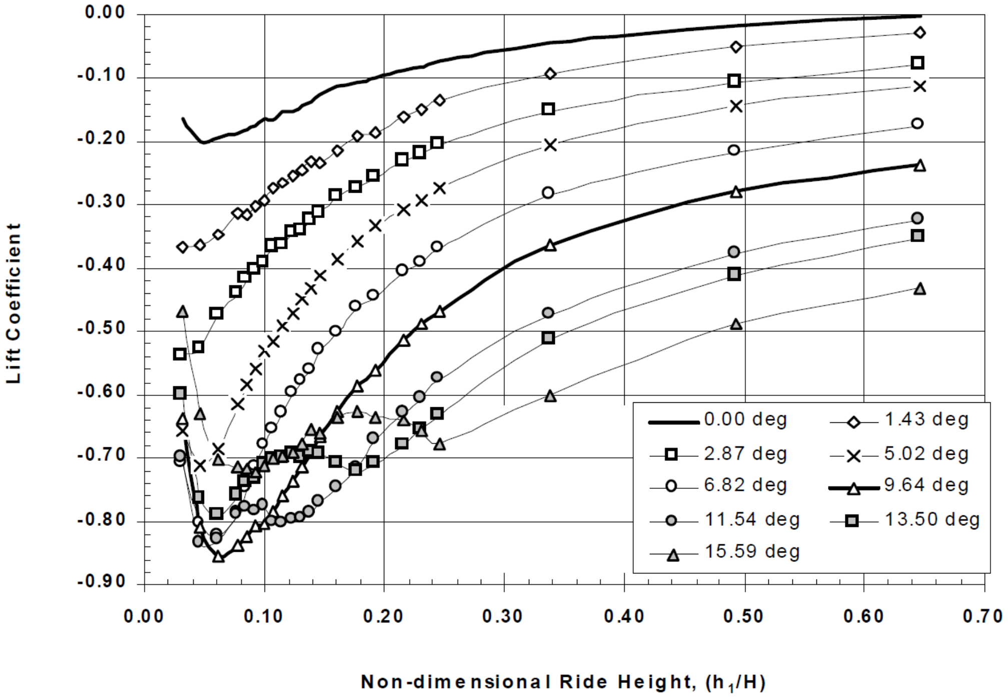


FIGURE 2.18: Lift coefficient curves against non-dimensional ride height at a range of diffuser angles [99].

2.3.4 The Impact of Rake

Rake, defined as the angle between the underfloor and the ground, is commonly used in contemporary race cars, yet the published body of knowledge regarding the impact of rake on diffuser performance is limited. The majority of existing studies on rake examined its effect by pitching the entire model, rather than deflecting just the underfloor. Although this method is analogous to raking an entire race car by modifying suspension geometry, it potentially obscures the changes in aerodynamic performance of the bottom of the model by also inclining the top surface.

George [106] found that the relationship between downforce and pitch angle is approximately linear for angles between ±30°, for a body in the freestream with a 20° diffuser and no end plates. Furthermore, significant changes to the flow patterns were reported, including the formation of counter-rotating vortices at pitch angles of 10° and above. However, the high pitch angles used in that study are not applicable to racing cars, and no results were presented at pitch angles lower than 5°.

Cooper et al. [99] varied the ride height of the model at nose-down pitch angles of 1.60° and 2.75°, and found that the increase in downforce coefficient due to a 1.60° pitch grew from 0.1 at a high ground separation to 0.4 near the critical ride height, suggesting that ground effect significantly amplifies the effect of rake. However, the effect of increasing pitch to 2.75° was much weaker. Furthermore, the downforce loss phenomenon at low ride heights was eliminated at both pitch angles, suggesting that the benefits of rake are two-fold—not only does it generate additional downforce, but it also broadens the diffuser's performance envelope. However, the causes behind this trend were not discussed.

Zhang et al. [98] presented downforce distributions of a generic open-wheeled race car on the front and rear axles with respect to front and rear ride height, which jointly define ride height and rake angle. Front downforce coefficient was shown to increase monotonically with decreasing front and increasing rear ride height, or increasing rake angle. This is a combination of two effects—increased efficiency of the front wing in ground effect, and diffuser effect induced by the entire underfloor due to positive rake. Rear downforce coefficient, however, was shown to have a maximum at a low ride height and a very small rake angle. This is due to the gradual forward shift of the centre of pressure due to increasing rake. These considerations demonstrate why optimising aerodynamic components for downforce, while maintaining a favourable aerodynamic balance, is a significant challenge.

A major shortcoming of the rake studies outlined above, and indeed the gap in the existing research, is the lack of data on performance trends and flow pattern variations when the underfloor is raked in isolation, with the rest of the model parallel to the ground, in order to separate the effects of inclining the different surfaces.

2.3.5 The Impact of Geometry Modifications

Mahon et al. [111], who used the same bluff body that was used for several other investigations [102, 103], but without diffuser end plates, reported lower critical ride heights, and suggested that the absence of end plates induces a less severe adverse pressure gradient within the diffuser, hence delaying vortex breakdown. Cooper et al. [99] carried out similar tests and showed that removing end plates increases downforce slightly, but only at near-critical ride heights, and observed no effect on the value of the critical ride height.

In a related study, Cooper et al. [118] investigated the effect of changing diffuser length, showing that maximum downforce is generated when the diffuser length is equal to or shorter than half the length of the entire body. Huminic and Huminic [119] presented computational results showing logarithmic growth of downforce with increasing diffuser length up to 50%, for diffuser angles up to 7°.

George [106] added wheels to his model, and observed strengthened vortical flow and an increase of the critical diffuser angle up to about 20°, whereas adding roughness to the flat underfloor had the opposite effect. In a follow-up study, George and Donis [107] showed that as the underbody was sealed with side skirts, the vortices could not form, leading to flow separation inside the diffuser and a decrease in downforce. However, this only occurred at diffuser angles of 10° and above, as at lower angles the flow inside the diffuser remained attached despite the absence of vortices, due to the sufficiently weak adverse pressure gradient. In further studies, Breslouer and George [113], Desai et al. [108] and Huminic and Huminic [120] observed reduced downforce for a diffuser-equipped bluff body with wheels, due to disrupted vortex formation and reduced flow beneath the underfloor. These contradicting results suggest that the impact of placing additional bluff objects underneath or to the side of the model strongly depends on the locations and shapes of those objects, as well as on the ride height and geometric characteristics of the diffuser.

Vortex generators are a common method of delaying flow separation in the presence of an adverse pressure gradient. They achieve this by inducing increased mixing of streamwise momentum between the freestream and the boundary layer, hence reducing the tendency of the flow to separate [121]. Kuya et al. [121] applied vortex generators to the suction surface of a single-element inverted wing in ground effect. Their results show that pairs of sub-boundary layer, counter-rotating vortex generators may result in an increase in downforce and aerodynamic efficiency of up to 26% and 10% respectively. The authors observed that, since no improvement occurred at large ride heights, the downforce increase was solely due to flow separation delay at low ride heights, as opposed to vortex-induced suction. This was confirmed by surface pressure measurements, which revealed stronger suction peaks and evidence of suppressed separation compared to a clean wing. The flow patterns were later examined using oil flow visualisation and wake surveys obtained by particle image velocimetry (PIV) [122], which showed that the delay in separation was most pronounced downstream of the centres of vortex generator pairs, due to the downwash generated by the resultant vortices. It was also suggested that vortex generators may be used to eliminate hysteresis in the downforce reduction phenomenon [121]. This could be beneficial in the case of diffuser flows, where hysteresis is a significant issue.

An alternative to additional vortex generators is splitting the diffuser into multiple channels. Jowsey and Passmore [114, 123] investigated the performance of 2-, 3- and 4-channel diffusers using a geometry similar to that of Cooper et al. [99], but without a moving ground. The force results showed that splitting the diffuser into 2 to 4 channels at diffuser angles up to 13° decreased downforce marginally due to the smaller effective surface area of the diffuser. However, a downforce increase of up to 17% was observed at diffuser angles between 13° and 19° and at low ride heights. These configurations correspond to the case where the diffuser is not yet stalled, but some separation is present due to the adverse pressure gradient. In this situation, channel splitters increase the strength of edge vortices by constraining them into narrower channels, making them more resistant to breakdown and increasing their ability to reattach the flow. This was confirmed by pressure measurements on the diffuser surface, which showed strengthened suction peaks at the inlets of the outer channels and lower pressure near the end plates, but negligible change in the inner channels [114]. Moreover, at a diffuser angle of 25°, which would result in loss of downforce in a single-channel diffuser, some downforce was retained in the 3- and 4-channel diffusers. This is due to the presence of vortices, and hence gradual pressure recovery, in the outer channels [114].

In 2019, Ehirim et al. [124] presented the most comprehensive review of ground-effect diffuser aerodynamics to date. In this review, the authors discussed existing experimental and numerical studies, but also described possible future directions for diffuser research. In addition to passive and active flow control measures for enhanced flow attachment and pressure recovery, they also discussed the potential use of curved surfaces near the diffuser exit, in order to encourage a second-stage pressure recovery and enhance the overall diffuser performance. This concept was studied further by Ehirim [116], who implemented several modifications to the diffuser geometry and investigated their effects on flow patterns and performance trends. A spanwise convex bump on the diffuser surface near the exit was shown to result in improved flow attachment over the bump and in slightly increased suction on its surface, which resulted in a downforce increase of up to 5% at the maximum ride height. However, the flow upstream of the bump was not noticeably affected [104, 116]. A small inverted wing placed at a similar location, parallel to the diffuser surface, resulted in an increase in downforce of up to 12% due to additional suction on the wing's surface, but had otherwise little effect on the diffuser flow [105, 116].

Most recently, Huminic and Huminic [125] studied the aerodynamics of curved, specifically convex, diffusers, using computational methods. Two types of diffuser plate curvature were studied, in addition to the plane diffuser—circular and elliptical. Both curves were tangential to the flat, horizontal underfloor, and in addition the ellipse was tangential to the vertical back plate of the model. The angle of all diffuser types—planar, circular and elliptical—was defined by the line between the points of contact of the diffuser plate with the underfloor and the vertical back plate. A significant number of configurations of all 3 diffuser shapes were tested, including four diffuser lengths (10%, 20%, 30% and 40% of body length) and four diffuser angles (2°, 4°, 6° and 8°). The results showed a significant increase in downforce due to curvature, of 25% and 53% on average for the circular and elliptical diffusers respectively, as well as an increase in drag, although less significant, and only at high diffuser angles. The enhanced downforce is caused by a shift and elongation of the suction peak at the diffuser inlet. Whereas flat plane diffusers experience a sharp peak followed by rapid pressure recovery, the suction peaks generated by curved diffusers are of similar magnitude, but with smaller gradients, increasing total suction under the body. In addition, the elliptical diffuser was found to strengthen the edge vortices, contributing even more downforce.

Chapter 3

Methodology

3.1 Initial Low-Fidelity Experimental System

The first experimental system comprised a small-scale, low-cost model of a diffuser-equipped bluff body with an actively controlled geometry. This system was used chiefly for the hardware-in-the-loop optimisation tests and the sampling time study in Chapter 4, as well as for a number of supporting tests. Furthermore, it was used to collect high-resolution performance maps with respect to its three degrees of freedom: ride height, rake and diffuser angles, which aided the design of the experimental campaign on diffuser aerodynamics, whose results are discussed in Chapter 6.

3.1.1 Wind Tunnel Facility

The wind tunnel used for this system was a low-speed, open-circuit wind tunnel at the University of Southampton. The driving fan was located at the inlet, followed by a series of flow-straightening honeycomb meshes, a contraction with a cross-sectional ratio of 7:1, and a rectangular test section with a cross-section of $600 \times 450\,\text{mm}$ and fixed walls. The freestream velocity was calibrated against fan frequency to a precision of $\pm 1\,\text{m/s}$ prior to the experiments, and the predetermined required fan frequency was maintained by a dedicated controller during the tests. The open circuit, the lack of real-time velocity control and unknown turbulence intensity meant that the flow quality, in terms of steadiness and alignment, was inferior to the facility used later in this work. Moreover, the lack of a moving ground introduces a quantitative skew into aerodynamic measurements of objects in ground effect. However, this facility was appropriate for the initial proof-of-concept study, and was used instead of the larger wind tunnel due to its ample availability, which was required for the time-consuming development and acquisition campaigns.

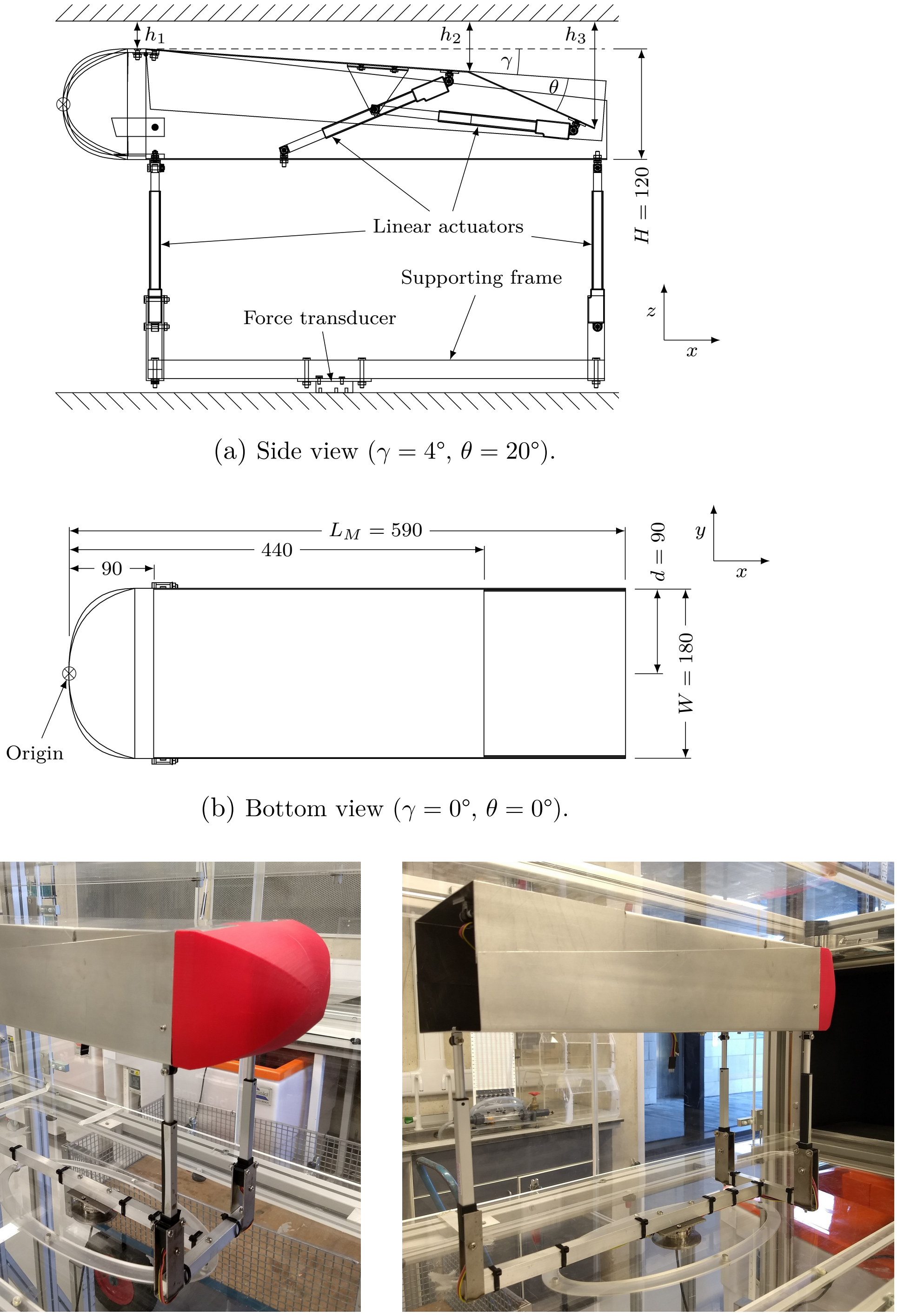
3.1.2 Model Design

The main aim of the design of this model was facilitating rapid data acquisition and automated optimisation through the use of remotely-controlled geometry manipulation. Although novel actuation methods and morphing concepts, such as those outlined in Section 2.1.2, could be used to advance shape articulation in HIL aerodynamic optimisation, emphasis in this study was placed chiefly on pursuing high efficiency and reliability, which were identified as the key shortcomings of existing studies. Therefore, mechanical actuation coupled with rigid-body motion and deflection were deemed to be most suitable for this system.

This solution permitted on-the-fly adjustments of the ride height, as well as the rake and diffuser angles of the model, as illustrated in Fig. 3.1, without the need to stop the wind tunnel to manually adjust or replace components. Data collection could be performed at up to 4800 configurations per day, resulting in much higher sampling resolutions than would be possible with a conventional system.

The diffuser model used in this system was a cuboid inspired by the Ahmed body [100], but with a more conical nose, and smaller width ($W = 180\,\mathrm{mm}$) and height ($H = 120\,\mathrm{mm}$) relative to its length ($L_M = 590\,\mathrm{mm}$), in order to maximise the aerodynamic forces while minimising wind tunnel blockage. The dimensions are shown in the schematics in Fig. 3.1, as are the coordinate system, ride heights h_1, h_2 and h_3, and rake and diffuser angles: γ and θ respectively. The model was installed upside down due to the existing layout of mounting interfaces, with the wind tunnel roof acting as the ground. The frontal area of the model was equal to $S = 0.0216\,\mathrm{m}^2$, which together with the supporting structure ($S \approx 0.0112\,\mathrm{m}^2$) resulted in a total blockage ratio of $\sim 12\%$.

The main part of the model's body was composed of two 1 mm-thick, U-shaped aluminium plates, one forming the roof and outer sidewalls, and the other forming the underfloor and inner sidewalls. A double-sidewall configuration was used to ensure that no end plates would be present along the underfloor when deflected. The concept is illustrated in Fig. 3.2, which shows the model's rear end for two configurations: zero and positive rake. This solution was used to obtain an analogous configuration to



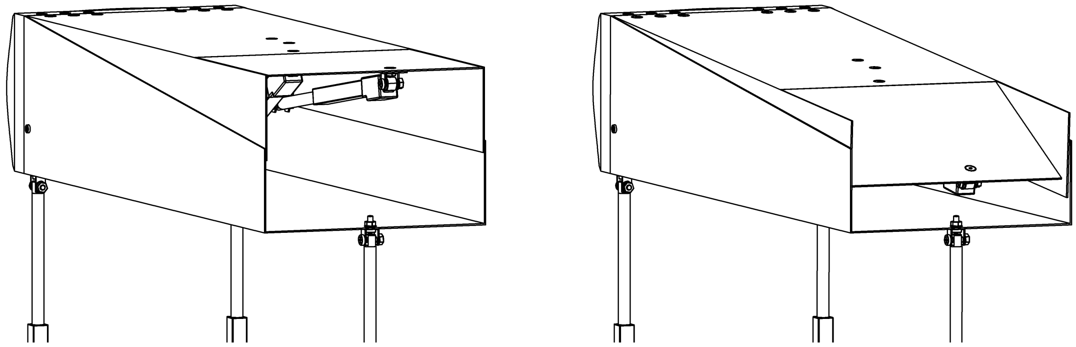
(a) Side view ($\gamma = 4°$, $\theta = 20°$).

(b) Bottom view ($\gamma = 0°$, $\theta = 0°$).

(c) Front view.

(d) Rear view.

FIGURE 3.1: Schematics and pictures of the small-scale model. The model was installed upside down, with the wind tunnel roof acting as the ground. All dimensions in mm.



(a) $\gamma = 0°$, $\theta = 0°$.

(b) $\gamma = 4°$, $\theta = 20°$.

FIGURE 3.2: Schematics of the rear end of the small-scale model.

previous studies [99, 106], where the models were pitched, and therefore no end plates were present at the sides of the underfloor. However, the impact of end plates was later investigated using the large-scale model, as detailed in Section 3.2.3.

The roof plate of the small-scale model was bolted directly to a single-element, 3D-printed nose cone, whereas the underfloor plate was hinged to the bottom edge of the nose, using conventional steel butt hinges. As underfloor deflection was limited to 6°, which covers the range of rake angles typically used in race cars, the gap between the hinged nose and the underfloor surface did not exceed 0.5 mm.

A third aluminium plate was used for the diffuser plate, which was mounted directly to the trailing edge of the underfloor. The two plates were hinged using a single strip of aluminium tape mounted on the outside, to ensure a smooth transition regardless of the angle between the plates. The maximum diffuser angle was limited by the roof of the model, and varied between $\theta = 46°$ at $\gamma = 0°$ and $\theta = 21°$ at $\gamma = 6°$. Furthermore, all mounting points on the underfloor and diffuser plates and on the underside of the nose cone were countersunk, in order to reduce their interference with the flow.

The model was attached to a supporting frame through three Actuonix L12-P-210 linear actuators, which were used to control the ride height, as shown in Fig. 3.1. The lowest ride height used in the tests was only 2 mm, or $h_1/H = 0.017$ in the non-dimensional form, which was made possible by the lack of a moving ground. The maximum ride height was limited to 92 mm ($h_1/H = 0.767$), yielding a 90 mm range.

Two further actuators of the same type were located within the bluff body, which independently controlled the rake and diffuser angles. The first connected the underfloor with the roof of the model, whereas the second connected the underfloor and diffuser plates, offset from the underfloor by a triangular mounting. This configuration of internal actuation, as displayed in Fig. 3.1a, ensured that only the angle of the underfloor was modified when rake was desired, rather than the pitch angle of the entire body.

The simple mechanical actuation system provided fast morphing (15 s between extreme positions), high positional precision (±0.2 mm), and good stability and reliability. The positional precision resulted in a resolution of 0.4 mm for the ride height, and average resolutions of 0.15° and 0.43° for the rake and diffuser angles respectively.

3.1.3 Force Measurements

An ATI Mini40 force transducer was fixed between the supporting frame and the wind tunnel wall, as seen in Fig. 3.1, and was the only point of contact between the model and the wind tunnel, apart from actuator cables. It was used to measure the forces and moments exerted on the model in all tests. Positive drag force, side force and downforce are acting along the x, y and z axes respectively (see Fig. 3.1). Positive rolling, pitching and yawing moments are acting about the three respective axes according to the right-hand rule, and their origin, which coincides with the origin of the coordinate system, is at the centre of the nose tip, as seen in Fig. 3.1a and 3.1b. The sensing range and resolution of the transducer are presented in Table 3.1.

Table 3.1: Specifications of the ATI Mini40 force transducer.

Load	Sensing Range	Resolution	Overload Range
F_x, F_y	±40 N	0.01 N	±810 N
F_z	±120 N	0.02 N	±2400 N
M_x, M_y	±2 Nm	0.000 25 Nm	±19 Nm
M_z	±2 Nm	0.000 25 Nm	±20 Nm

3.1.4 Closed-Loop Control System

The actuation system was controlled by a MATLAB script running on a personal computer (PC), which sent target positions for each actuator to a Simulink control system built on a dSPACE DS1006 processing board. The target positions were compared to positions reported by the actuators' built-in potentiometers, and the required direction of travel was selected if the positional error was greater than the specified precision window. Appropriate signals were then sent to the actuators through a digital-to-analogue converter (DAC) in the dSPACE chassis, and through a custom control board powered by a direct current (DC) supply. Position feedback was transmitted from the actuators through the control board, through an analogue-to-digital converter (ADC), and back to MATLAB.

An additional filter was implemented in the Simulink system to prevent the three ride height actuators from causing damage when going out of plane. Whenever the largest position difference between any of the three actuators exceeded 10 mm, all actuators would be stopped immediately and the operator would be notified for manual troubleshooting.

The force transducer was connected through a DC power supply to a laptop, where force and moment readings were acquired and calibrated automatically in a LabVIEW program. The measurements were then sent to the main PC through the local network, using the UDP protocol. A schematic of the control and acquisition system is displayed in Fig. 3.3, with the equipment and model depicted in Fig. 3.4.

The acquisition script, which was also built into the fitness function used by the optimisation algorithms, carried out the following steps:

1. Send the desired positions to the actuators;
2. Wait for the actuators to reach their targets while monitoring position feedback;
3. Wait for the flow to settle for a predefined amount of time;
4. Collect measurements over the required sampling time.

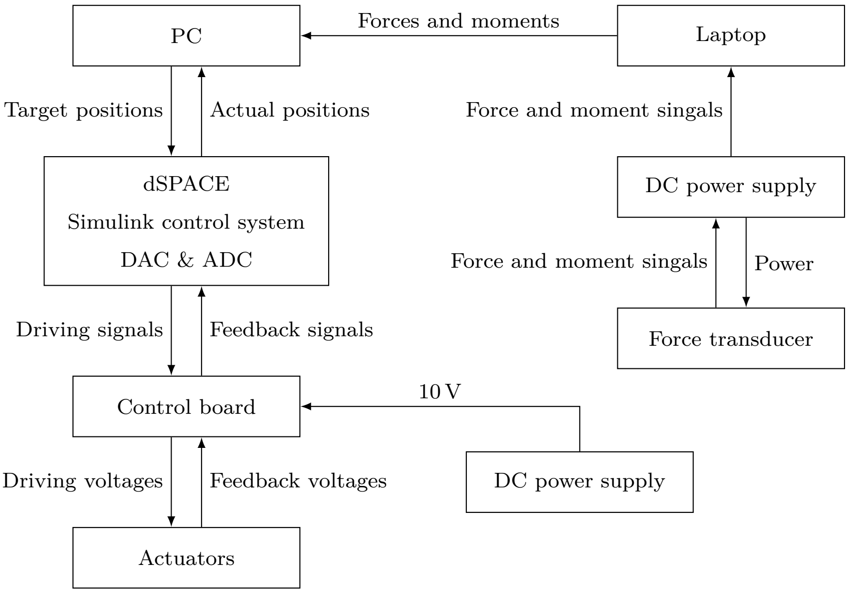


Figure 3.3: Schematic of the small-scale model's control and acquisition system.

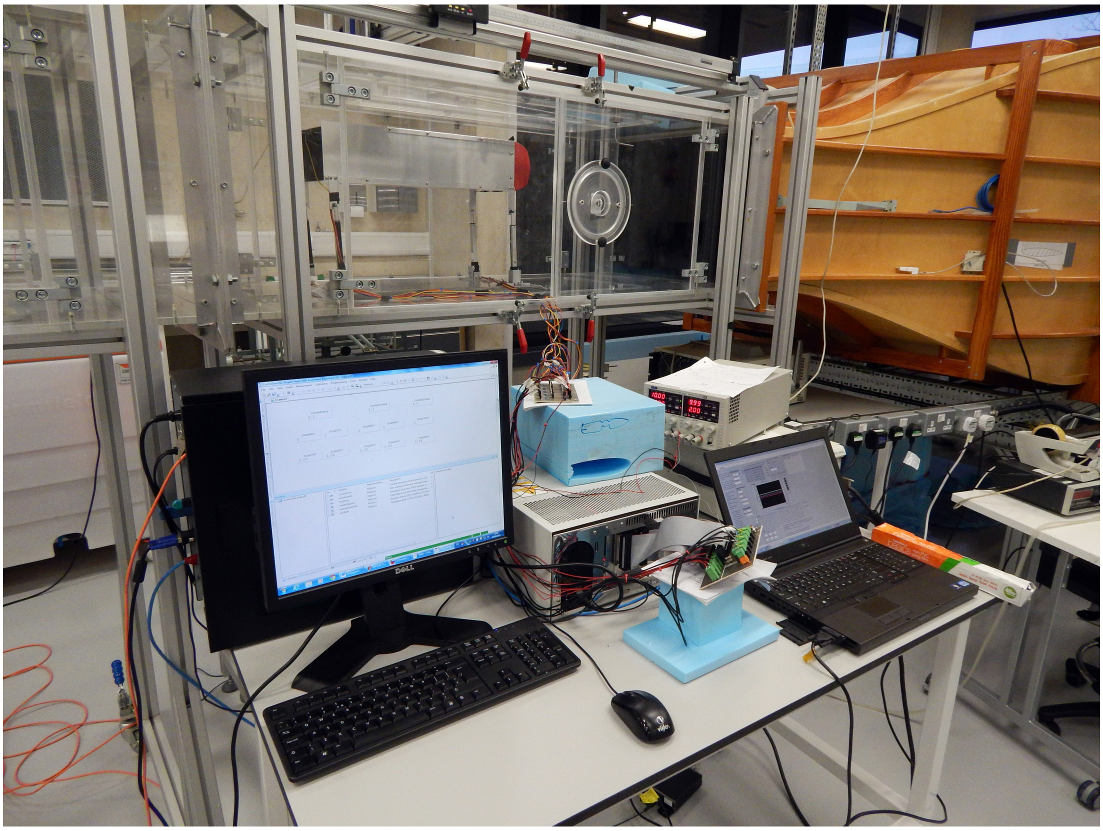


Figure 3.4: Acquisition and control hardware for the small-scale model.

3.1.5 Testing Procedure

Prior to testing in the wind tunnel, the model and actuation system were assembled, and the extreme positions for the actuators were determined. The minimum ride height was set at 2 mm to prevent the model from impacting on the roof of the test section, and the maximum of 92 mm was constrained by the actuator extension limits. The rake and diffuser actuators were limited by 0° rake and diffuser angles on one end, and by the roof of the model at the other. A gap of at least 2 mm was maintained between the trailing edge of the diffuser plate and the roof of the model at all times.

The testing freestream velocity was determined considering the sensing range of the force transducer. It was identified that the pitching moment sensing range of ±2 Nm was the limiting factor, due to the distance of the bluff body from the transducer. The bluff body was transformed into several configurations at a safe velocity and the one generating the largest pitching moment was identified (ride height of 7 mm, maximum rake angle and zero diffuser angle). Wind tunnel velocity was then increased until the sensing range limit was reached at 20 m/s (significantly below the safety limit of the transducer), and the final velocity was set at approximately $U_\infty = 16$ m/s to account for uncertainties. All tests using the small-scale system were carried out at this velocity, which corresponds to a Reynolds number of approximately $Re = 6 \times 10^5$, based on the full length of the bluff body ($L_M = 0.59\,\mathrm{m}$).

Tare forces and moments were acquired with wind off prior to every test, and the readings were subtracted from every subsequent measurement to eliminate static loads. The aerodynamic forces generated by the supporting structure were measured at the testing velocity with the bluff body and its internal actuators removed, but with the aluminium frame and the actuators controlling ride height still in place. The readings were taken at four equally-spaced extensions of the ride height actuators (corresponding to ride heights of 2 mm up to 92 mm, every 30 mm), to account for increased drag of longer actuators. The curves obtained for all 6 force components were approximately linear, so straight lines were fitted, and the resulting frame-only tares were subtracted from all subsequent measurements. R^2 values of the linear fits for the two loads most affected by actuator length, i.e. drag force and pitching moment, were 99.6% and 99.8% respectively, and between 78% and 95% for the remaining loads. However, very small variations were measured for the latter (up to 0.05 N and 0.0014 Nm for the forces and moments respectively), so the trends were significantly affected by measurement noise, but also less consequential for the processed measurements.

Preliminary tests revealed significant drift of force and moment measurements, therefore a datum configuration of the model was tested at regular intervals during performance map acquisitions. The datum was defined as the maximum ride height ($h_1 = 92\,\mathrm{mm}$) and zero rake and diffuser angles, as this configuration creates the most steady flow patterns, and is not susceptible to hysteresis with respect to any of the three degrees of freedom. The datum series of all 6 force components from each performance map were interpolated over time and smoothed, then normalised by the first datum measurement. An example datum series, its smoothed interpolation and the resultant drift correction curve are shown in Fig. 3.5. Then, at each data point in the performance maps, force and moment coefficients were normalised by the drift correction at the subsequent datum point. This procedure was applied to all performance maps, but not to real-time optimisation runs, as it would interrupt the continuity of optimisation. However, repeated data points during optimisation were reacquired to minimize the impact of drift on convergence.

In order to determine the sampling time required to obtain reliable average loads,

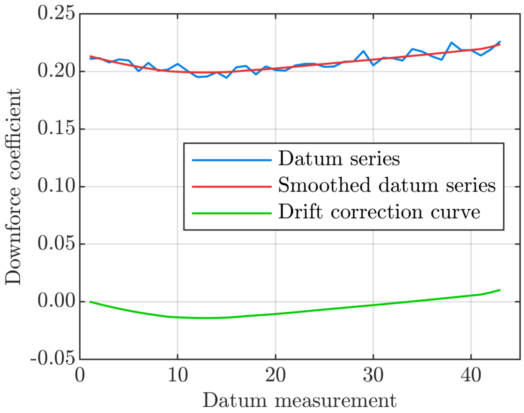


FIGURE 3.5: An example series of datum measurements, its smoothed interpolation, and the resultant drift correction curve.

continuous wind-on samples of up to 90 min were collected for configurations of the diffuser that did not exhibit any significant unsteady flow phenomena, such as flow separation from the diffuser surface. Inspection of running averages in a 20 min sample showed that a series of 2 s samples yields a smooth curve, which stays within 0.72% of the average of the entire sample, with a standard deviation of 0.00114, or 0.23% of the average. This sample length was deemed to provide sufficient precision and was used for most tests using the small-scale system. Configurations that promote unsteady flow phenomena were observed to induce small vibrations of the bluff body, and are therefore likely to generate increased noise in the readings. Although this could be alleviated by using a longer sampling time, the increased noise in regions of unsteady flow may act as a natural deterrent to convergence during optimisation.

Similar tests on prolonged continuous samples were carried out in order to determine the approximate settling time of the flow once shape transformation is completed. For this purpose, the model was actuated to several random configurations at intervals of approximately 10 s, and the system was deemed settled once a 1 s running average of downforce remained within a certain margin for a certain length of time. Manual trials using a range of margins and time lengths and direct comparisons to the continuous sample indicated that, if a margin of 0.05 N ($C_L \approx 0.015$) was satisfied for 0.5 s, the flow was very likely to be settled. A sample of downforce data, together with the 1 s running average, is shown in Fig. 3.6. The shaded areas of the plot are the periods when the test indicated that the flow was not settled. Importantly, no overshoot of loads after actuation was present in the majority of cases, only gradual convergence towards the settled value. The minimum time necessary to obtain the 1 s average and

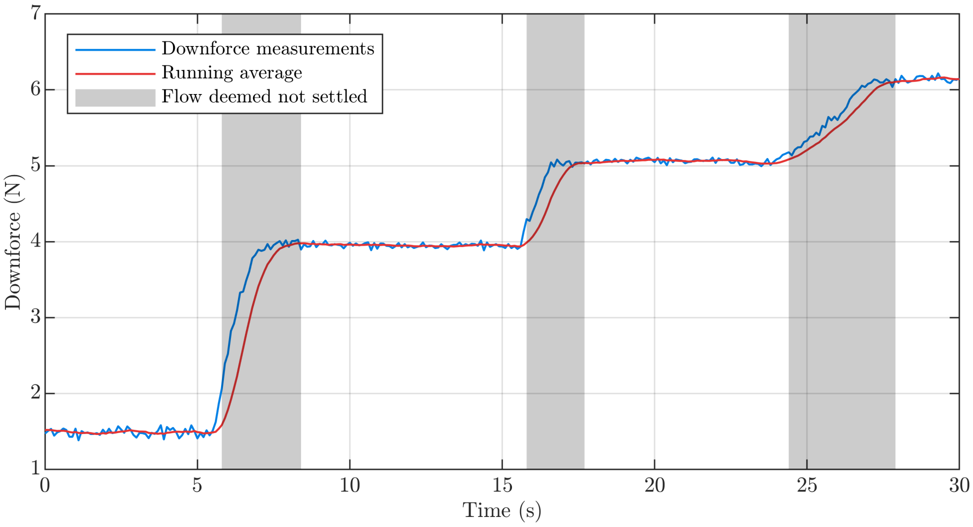


FIGURE 3.6: A continuous downforce sample averaged at 10 Hz and its 1 s running average. Shaded areas are periods when the flow was deemed to be not settled.

to monitor its variation for 0.5 s is 1.5 s, and preliminary optimisation runs showed that the vast majority of cases took between 1.5 s and 1.6 s to settle. In order to simplify the testing procedure, a fixed settling time of 1.5 s was selected as the default value for tests using the small-scale system.

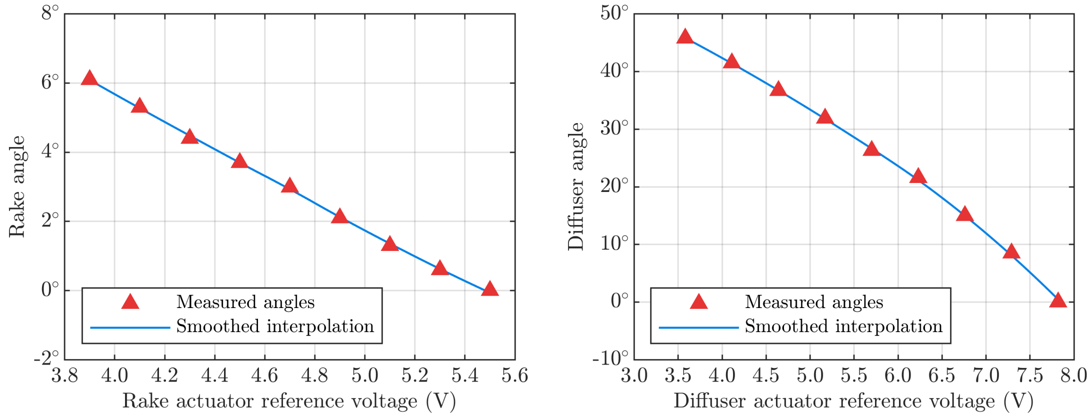
In order to assess the effect of modifying the settling and sampling times on measurement noise and hysteresis, and therefore on the convergence performance of optimisation algorithms, a more detailed investigation was also carried out. The results of this study, which involved both computational and experimental tests, are presented in Section 4.3.

3.1.6 Data Processing

The acquisition script, as outlined in Section 3.1.4, sent desired actuator positions in the form of integers, which corresponded to specific positions in the discretised, three-dimensional design space. These values were converted to voltage signals between 0 and 10 V, which corresponded to full retraction (0 mm) and full extension (100 mm) of the actuators respectively. Actuator feedback signals were on the same scale, and were compared to the target signals directly to determine the required direction of actuation.

Model ride height h_1 was linearly related to the extension of the ride height actuators, so only one data point was needed to obtain the calibration curve. The minimum ride height ($h_1 = 2\,\mathrm{mm}$) corresponded to actuator extension of 95 mm, or a reference voltage of 9.5 V, therefore the calibration curve took the form $h_1 = 97 - 10V_1$, where h_1 is ride height in mm and V_1 is ride height reference voltage in V.

The relations between reference voltages for rake (V_2) and diffuser (V_3) and the respective angles (γ and θ) were established by displacing both plates to series of positions and measuring the resulting angles with a digital protractor. The data points were first interpolated linearly, and then smoothed using linear least squares regression and a 2nd degree polynomial model using 90% of the total number of data points. This method was selected as it was qualitatively assessed to provide the best compromise between eliminating distortions caused by low measurement precision, and remaining faithful to the original trends. The final calibration curves, as shown in Fig. 3.7, were sampled and used as look-up tables during further processing.



(a) Rake angle.

(b) Diffuser angle.

FIGURE 3.7: Rake and diffuser angle calibration curves for the small-scale model.

All force and moment measurements were calibrated using the transducer's factory calibration matrix. After wind-off tares and frame-only aerodynamic loads were subtracted, the measurements were converted to their respective coefficients according to Eq. 3.1 and 3.2, non-dimensionalised by the bluff body's frontal area $S = 0.0216\,\mathrm{m}^2$, and its length $L_M = 0.59\,\mathrm{m}$. Dynamic pressure q_∞ was used directly, as this was the output of the manometer used to measure the freestream velocity.

$$C_F = \frac{F}{q_\infty \cdot S}, \tag{3.1}$$

$$C_M = \frac{M}{q_\infty \cdot S \cdot L_M}, \tag{3.2}$$

where C_F and C_M stand for force and moment coefficients, and F and M are forces and moments respectively.

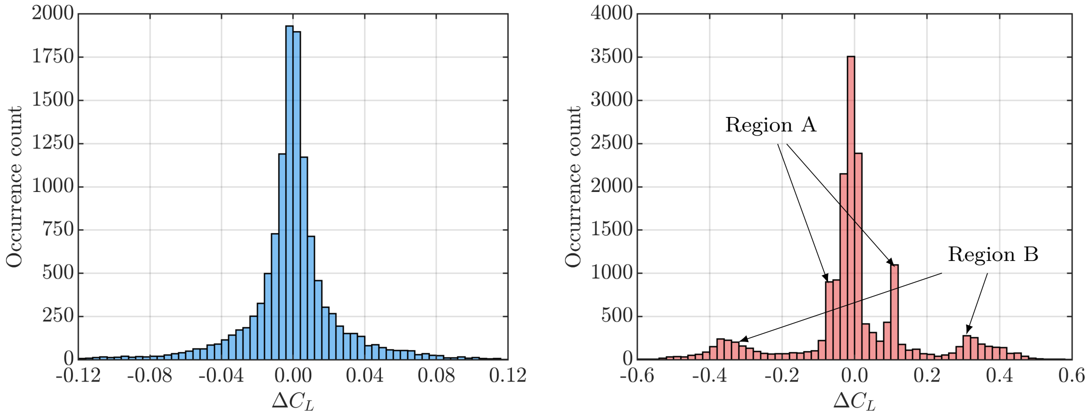
Ride height h_1 was non-dimensionalised by model height H, following the convention applied to similar geometries by Cooper et al. [99] and Jowsey and Passmore [114, 123].

However, h_1/d, where d is model half-width, was indicated on secondary axes where possible, to enable a more convenient comparison with other studies [102–105, 110, 111, 116].

3.1.7 Measurement Repeatability

In order to estimate long-term measurement repeatability, downforce readings for all duplicate design points within the collected data were compared. The mean downforce coefficient μ_{C_L} was calculated for each unique design point, then all individual readings were compared to their respective means. Histograms of deviations from the mean, $\Delta C_L = C_L - \mu_{C_L}$, are plotted in Fig. 3.8. The histogram in Fig. 3.8a presents a distribution of ΔC_L calculated using data from extensive sampling of the search space (see Fig. 4.1), as well as from the optimisation runs presented in Section 4.2. The histogram includes $12,395$ points, and the standard deviation is equal to $\sigma_{C_L} = 0.0351$. As the data used for this calculation was sampled over several weeks, σ_{C_L} indicates the long-term repeatability for downforce measurements.

The histogram in Fig. 3.8b shows a distribution of ΔC_L calculated using data from hysteresis maps, where quasi-static variations of diffuser angle were carried out in both directions, i.e. gradually decreasing and increasing the inclination of the diffuser (see Fig. 4.2). This histogram includes $16,978$ points, and the standard deviation is equal to $\sigma_{C_L} = 0.1713$. Five peaks may be observed, including the main peak at $\Delta C_L = 0$. Next, there is a pair of peaks at approximately $\Delta C_L = -0.06$ and $\Delta C_L = 0.10$, and



(a) Comparison using performance map and optimisation run data.

(b) Comparison using hysteresis map data.

Figure 3.8: Long-term repeatability of downforce measurements of the small-scale model based on repeated data points.

another pair at approximately $\Delta C_L = -0.36$ and $\Delta C_L = 0.32$. These four peaks correspond to the two hysteresis regions seen in Fig. 4.2, where the high-downforce and low-downforce regimes are above and below the mean downforce values respectively.

3.2 High-Fidelity Experimental System

The ground effect aerodynamics part of this project aimed to shed light on the performance and underlying flow physics of a diffuser-equipped bluff body with rake. This required improved geometry control, measurement precision and accuracy relative to the low-cost model and facility described in Section 3.1, which although suitable for its mainly comparative tests, did not provide the measurement quality required for a detailed aerodynamic analysis. Therefore, a new system was designed, as shown in Fig. 3.9, whose main goals were to recreate the conditions from previously-published studies as closely as possible, and to acquire more information about the flow around the model. This necessitated changes to model geometry and structural design, refined control and acquisition systems, and a wind tunnel facility equipped with a moving ground. Notably, the lessons learnt during extensive tests of the small-scale model

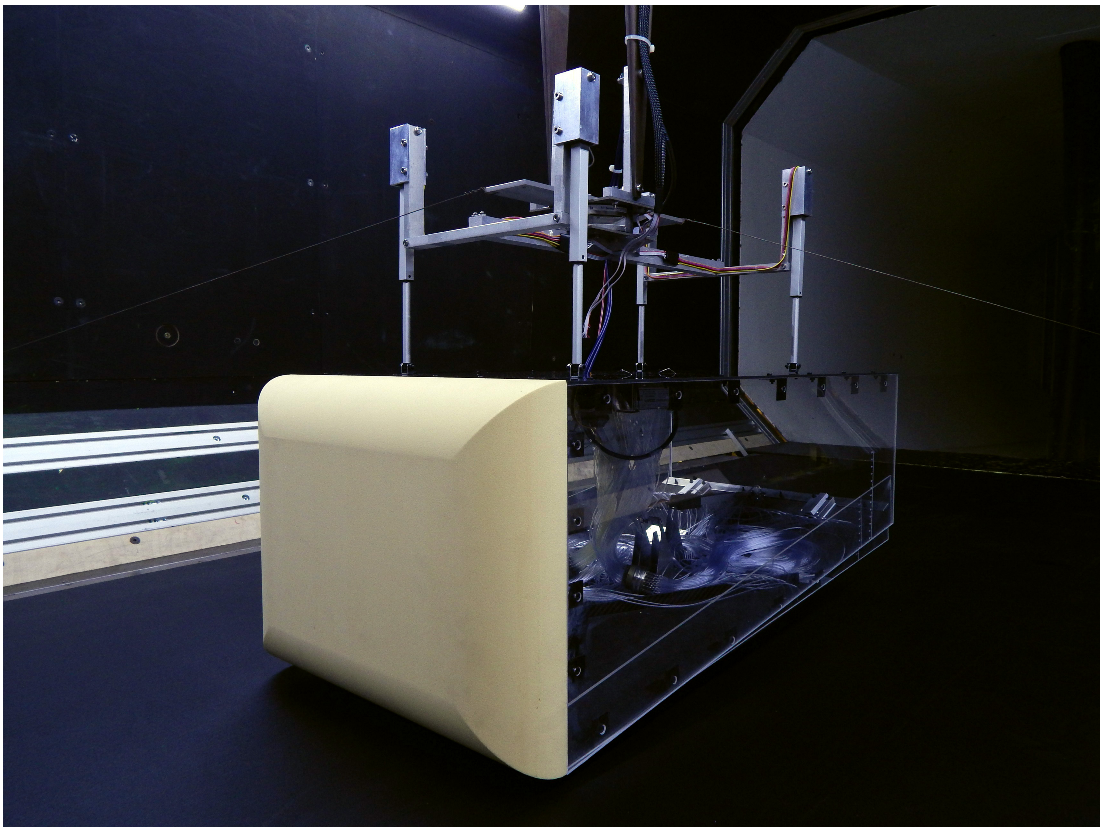


FIGURE 3.9: The large-scale diffuser model in the wind tunnel.

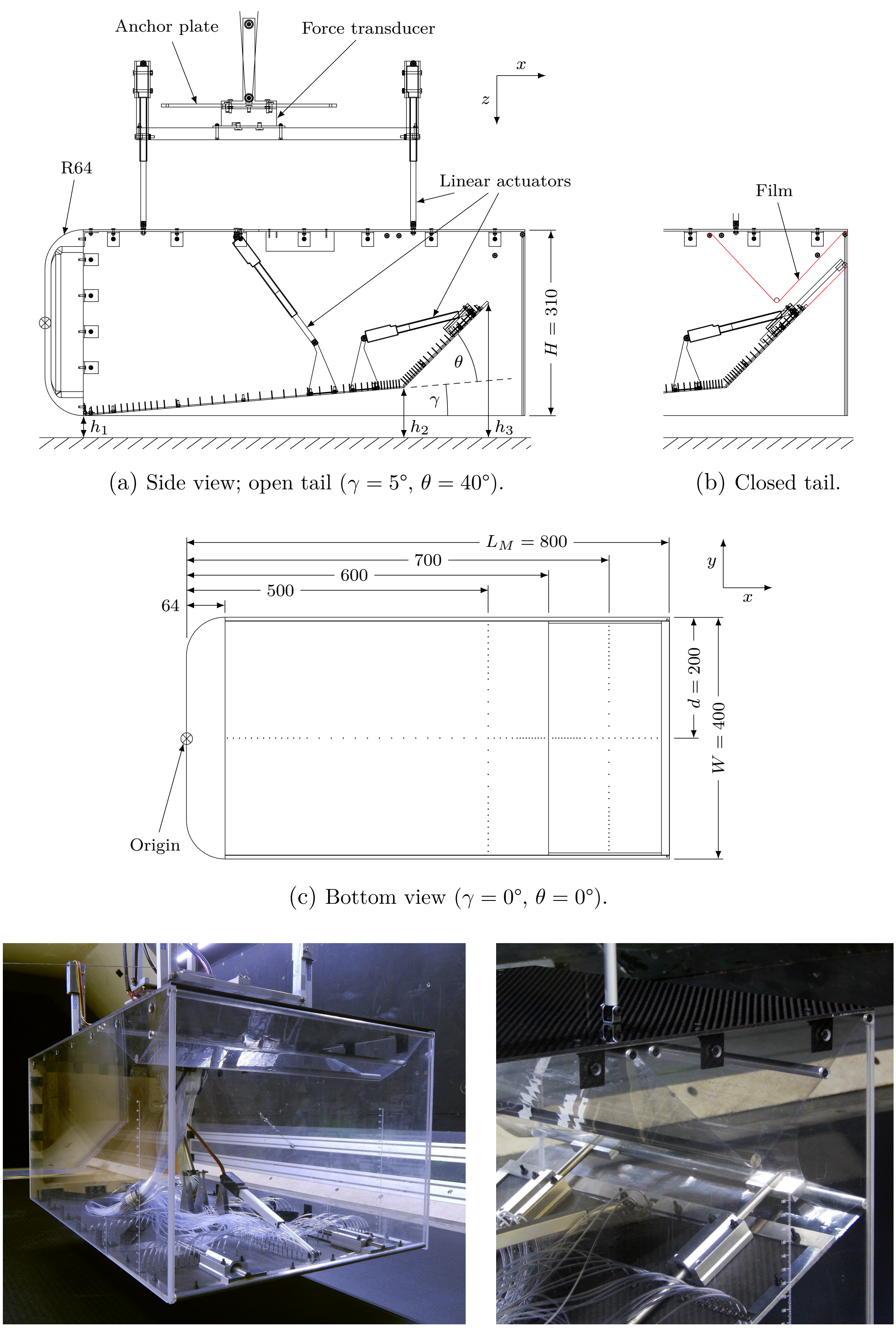
enabled numerous improvements to be made in the new design. The resulting system was used chiefly for aerodynamic measurements, with the results presented in Chapter 6, but also for experimental validation of population sorting, as described in detail in Section 5.7. The design and operation of this system are presented in detail in the following section, and it is hitherto referred to as the large-scale system.

3.2.1 Wind Tunnel Facility

Experiments using the large-scale system were carried out in a closed-circuit, low-speed wind tunnel at the University of Southampton, the same facility where a number of previous diffuser studies took place [102, 103, 106, 110, 111]. The test section has a 2.15 m by 1.57 m cross-section with chamfered corners. The size of the chamfers decreases in the direction of the flow to account for the growing boundary layers on tunnel walls, causing the cross-sectional area of the test section to increase from 3.246 m^2 to 3.273 m^2 across its length. The test section is also equipped with a moving ground assembly, which consists of a 2-stage boundary layer removal system, followed by a 1.442 m-wide and 3.350 m-long belt. In contrast to the small-scale experiments described in Section 3.1, all tests using this system were carried out with the moving ground run at freestream velocity, to ensure correct boundary conditions under the model. This had been shown to be essential, as an additional boundary layer on the ground surface has a significant impact on quantitative diffuser performance [99, 101–103, 106, 107, 116, 117]. The belt was tensioned and run over a perforated plate, through which suction was applied, in order to prevent the belt from raising due to low pressure between the belt and the model above. The freestream and moving ground velocities were controlled with a precision of ±0.05 m/s, and the freestream turbulence level was previously specified at 0.2% [102]. A more detailed description of the wind tunnel and the moving ground assembly was presented by Burgin et al. [126].

3.2.2 Model Design

The large-scale model was a cuboid with a width of $W = 400\,\mathrm{mm}$, a height of $H = 310\,\mathrm{mm}$, a length of $L_M = 800\,\mathrm{mm}$, and a 64 mm radius on all four sides of the nose, as shown in Fig. 3.10. These dimensions have similar ratios to the models used by Cooper et al. [99], Breslouer and George [113] and Desai et al. [108], and are identical to the model of Jowsey and Passmore [114]. The frontal areas of the model ($S = 0.124\,\mathrm{m}^2$) and the supporting structure and struts ($\sim 0.043\,\mathrm{m}^2$) resulted in a total blockage ratio



(a) Side view; open tail ($\gamma = 5°$, $\theta = 40°$).

(b) Closed tail.

(c) Bottom view ($\gamma = 0°$, $\theta = 0°$).

(d) The model in the wind tunnel.

(e) Film mechanism detail.

FIGURE 3.10: Schematics and pictures of the large-scale model. All dimensions in mm.

of $\sim 5.1\%$. The roof of the model was made of a 3 mm-thick carbon fibre plate, the nose was machined out of low-density tooling block with a 10 mm thickness, and the model's sidewalls were machined out of 6 mm-thick transparent acrylic.

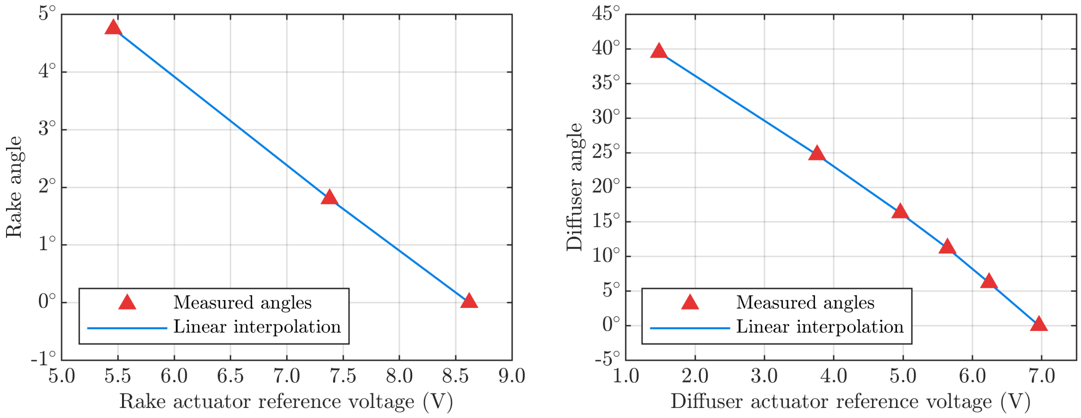
Similar to the small-scale model described in Section 3.1.2, mechanical actuation was used to modify the position and shape of the model, in order to maintain the high reliability and efficiency of the first system. The bluff body was suspended from a supporting frame by four Actuonix L16-P-63 linear actuators, which were used to control the ride height, h_1, defined as the height of the lowest point of the model above the ground (see Fig. 3.10a). Four actuators with increased power output were used in order to withstand the greater weight and downforce compared to the small-scale model, and their streamwise and spanwise locations were optimised using simple finite element analysis to minimise deflections of the model's roof under load. The range of ride heights was between 10 mm and 90 mm above the moving ground, or between $h_1/H = 0.032$ and 0.290 in the non-dimensional form. Lower ride heights were not used for safety reasons, and the range was constrained by the actuator extension limit. h_2 and h_3 in Fig. 3.10a denote the heights above the ground of the diffuser inlet and outlet respectively. The supporting frame was mounted to a force transducer, which was attached to wind tunnel struts. An additional steel plate was mounted between the force transducer and the struts, which was used to anchor the structure to wind tunnel walls, reducing vibrations of the model and allowing precise adjustments of its yaw angle.

The bottom of the model was comprised of two 2 mm-thick carbon fibre plates. The underfloor, 536 mm-long, was hinged to the nose, and its angle relative to the ground was modified by an Actuonix L16-P-150 linear actuator, mounted between the underfloor and the roof of the model. The diffuser plate, 200 mm-long, was hinged to the underfloor, their relative angle controlled by an identical actuator, mounted between the two plates. The underfloor was hinged to the nose with conventional steel butt hinges, as it was only deflected to a maximum rake angle of $\gamma = 5°$, so the gap between the surfaces did not exceed 0.5 mm, similar to the small-scale model. The diffuser plate and the underfloor had their coincident edges chamfered at 45°, and were joined with cross-braced aluminium tape, which ensured that no gap was present between the surfaces, even at the maximum diffuser angle of $\theta = 40°$. The maximum rake angle was chosen to cover the range typically used in race cars, and the maximum diffuser angle was constrained by the model's roof. Furthermore, in order to reduce the aerodynamic effect of mounting points, all bolts in the sidewalls, the underfloor and the diffuser plate were countersunk.

The model was installed with the roof parallel to the ground and the sidewalls parallel to the wind tunnel walls, with uncertainties on pitch, roll and yaw angles of ±0.2°, ±0.3° and ±0.2° respectively.

All individual actuators were calibrated using a digital caliper to ±0.2 mm, and the maximum permitted positional error in the control system was ±0.5 mm, yielding a total ride height uncertainty of ±0.7 mm. Positional precision of the actuators was also included in the pitch and roll angle uncertainties mentioned above.

The rake and diffuser angles were calibrated in situ with wind on, using laser-engraved scales on the inner surfaces of the sidewalls, and the uncertainties are ±0.1° and ±0.9° respectively. The calibration measurements and uncertainties were obtained from continuous video recordings of the scales next to the bottom edges of the underfloor and diffuser plates, as seen in Fig. 3.10d and 3.10e. The data points were then linearly interpolated to obtain calibration curves, which are shown in Fig. 3.11. As the precision of these measurements was greatly increased relative to the small-scale model, no smoothing was applied.



(a) Rake angle.

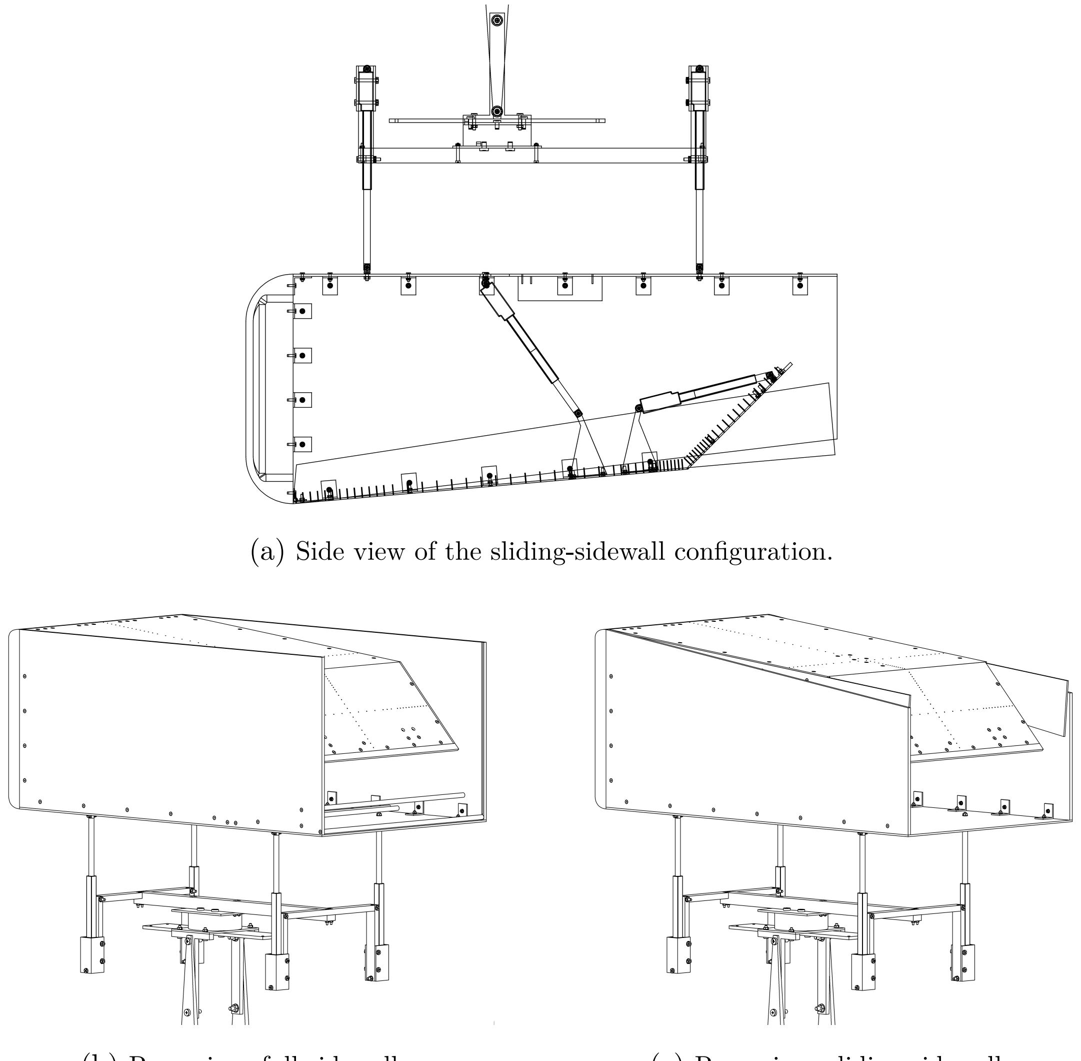
(b) Diffuser angle.

FIGURE 3.11: Rake and diffuser angle calibration curves for the large-scale model.

3.2.3 Model Configurations

The primary purpose of the large-scale model was investigation of the impact of rake on diffuser aerodynamics, where the presence of end plates on the sides of the underfloor is crucial. In order to gain understanding of the influence of rake both with and without underfloor end plates, two configurations of the model were devised. The first, as presented in the schematics and pictures in Fig. 3.10, consisted of fixed, rectangular sidewalls, which also served as end plates when the underfloor was deflected, or raked. The second configuration, similar to the small-scale model (Fig. 3.2), contained two pairs of sidewalls: fixed and sliding, the latter mounted to the underfloor. Detail of this configuration is shown in Fig. 3.12a, and a comparison between the two configurations is illustrated in Fig. 3.12b and 3.12c.

In both configurations, the ability to actuate the underfloor and diffuser plates in real-time meant that the trailing edge of the diffuser plate could not be connected to the trailing edge of the model's roof, as the location of the diffuser exit changed both horizontally and vertically. This resulted in an open cavity at the rear of the model, unlike



(a) Side view of the sliding-sidewall configuration.

(b) Rear view; full sidewalls.

(c) Rear view; sliding sidewalls.

Figure 3.12: Schematic of the sliding-sidewall configuration of the large-scale model, and comparison of the two sidewall configurations.

in previous studies on diffuser aerodynamics, where variable-length diffuser plates were connected to solid vertical plates at the rear. In order to quantify the effect of extending the diffuser plate and closing the tail cavity, a thin-film system was implemented, which covered the cavity with a transparent film, first extending the diffuser plate, and then connecting it vertically to the roof, as illustrated in Fig. 3.10b and 3.10e.

The film was gripped between the trailing edge of the carbon fibre diffuser plate and an aluminium extension bar, then wrapped around an aluminium rod coincident with the end of the model. This rod was attached to the diffuser plate through a pair of additional rods located within linear bearings, which ensured that the first section of the film was always parallel to the diffuser plate. Simultaneously, the rod was placed in grooves running along the trailing edges of the sidewalls. Next, the film was pulled upwards and wrapped around another rod, right next to the TE of the roof plate, before going inside the model to the final attachment point. A steel rod was hung between the two final rods, which provided tension to the film, as clearly seen in Fig. 3.10b, 3.10d and 3.10e.

The thin-film system could not be used with the sliding-sidewall configuration due to the tapered shape of the fixed sidewalls, but all tests on the full-sidewall configuration were carried out without and with the film, and a brief analysis of the differences in performance is presented in Appendix A. The primary discussion, however, is carried out on the open-cavity configuration, due to the film's poor repeatability; despite low friction and high tension, the film was pushed inwards by air pressure, curved at a varying degree depending on the configuration (see Fig. 3.10e), and in some cases flutter was observed. However, the system was useful to, firstly, establish quantitative correlation between open- and closed-tail diffuser studies, and secondly, to provide further insight into how modifying the rear end of the model affects the flow through the underbody.

3.2.4 Force and Pressure Measurements

Forces and moments exerted on the model were measured using an ATI Delta 6-component transducer, whose main specifications are shown in Table 3.2. The directions of positive drag force, side force and downforce are along the x, y and z axes respectively, whereas the directions of the rolling, pitching and yawing moments are around those respective axes according to the right-hand rule, and their origin is at the centre of the frontal plane of the nose, which is also the origin of the coordinate system.

Table 3.2: Specifications of the ATI Delta force transducer.

Load	Sensing Range	Resolution	Overload Range
F_x, F_y	±330 N	0.0625 N	±3700 N
F_z	±990 N	0.125 N	±10 000 N
M_x, M_y	±30 Nm	0.003 75 Nm	±280 Nm
M_z	±30 Nm	0.003 75 Nm	±400 Nm

The axes and the location of the origin are indicated in Fig. 3.10a and 3.10c. The force transducer was the only point of contact between the model and the wind tunnel, apart from actuator and pressure scanner cables and the reference pressure tube. To reduce their impact on measurements, they were tied to the model and to the wind tunnel struts as close to the transducer as possible.

The underfloor and diffuser plates were pressure-tapped along the centreline of the model, with a total of 66 taps along both surfaces. In addition, there were two spanwise series of taps, 28 across the underfloor and 34 across the diffuser plate, 100 mm upstream and downstream of the diffuser inlet respectively. The tap layout is shown in Fig. 3.10c. Metallic tubes with a 0.762 mm internal diameter were embedded in the plates, and were flush with the surfaces on the outside. They were connected via vinyl tubing to two 64-channel Scanivalve ZOC33 differential pressure scanners with a ±2500 Pa sensing range and an accuracy of 0.15% of the full-scale. The reference pressure channels were connected to the static pressure channel of a pitot-static tube in the freestream. Both pressure scanners were calibrated at two temperatures, 11.3 ± 0.2 °C and 21.7 ± 0.2 °C, across the entire sensing range, to within ±5 Pa.

At each data point, force and pressure measurements were collected sequentially: first, a vector containing force and moment measurements averaged over 0.1 s, followed by instantaneous readings of pressure from 64 consecutive pairs of channels (1 & 65, 2 & 66, ..., 64 & 128). This sequence took $\sim$ 0.2 s to acquire and was repeated 50 times, yielding a total acquisition time of approx. 10 s. All measurements were then time-averaged, resulting in a single set of values for all force components and pressure channels. This acquisition method, as opposed to acquiring long samples from one channel at a time, ensured that the resultant average values represent the entirety of the sampling period.

3.2.5 Closed-Loop Control System

The closed-loop control and acquisition system of the large-scale model was similar to the one described in Section 3.1.4, yet with a few notable differences; a simplified schematic is shown in Fig. 3.13, with pictures of the control and acquisition equipment in Fig. 3.14. The top-level acquisition script was run in MATLAB on a PC, and the actuators were controlled by the manufacturer's Linear Actuator Control boards, which converted target positions to voltage signals using an internal proportional-derivative controller. These were powered with a series of external power supplies and connected to the PC via Universal Serial Bus (USB), similar to the data acquisition (DAQ) device NI USB-6210, which was used to acquire force and moment signals from the externally-powered force transducer.

The pair of pressure scanners were controlled by a Simulink system built on a dSPACE DS1006 board, connected directly to the same PC as the actuators and the force transducer. The Simulink model converted pairs of channel numbers received from the MATLAB script to binary, and forwarded them to the scanners via a DAC board and a custom control board. This resulted in instantaneous switching of output channels in the scanners, and the pressure signals were sent back to the PC via the same pathway.

The top-level acquisition script in MATLAB carried out the following operations:

1. Send the target positions to the actuator control boards;

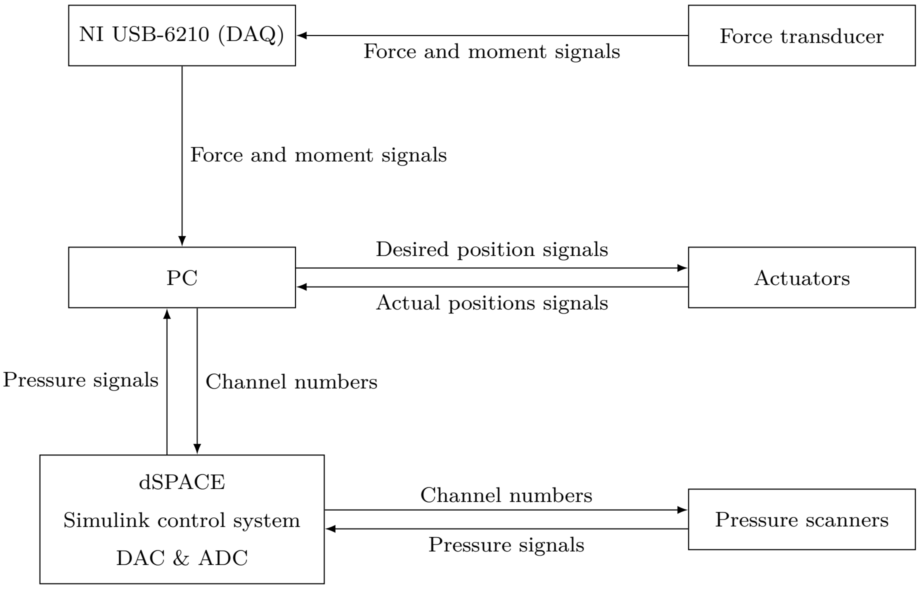
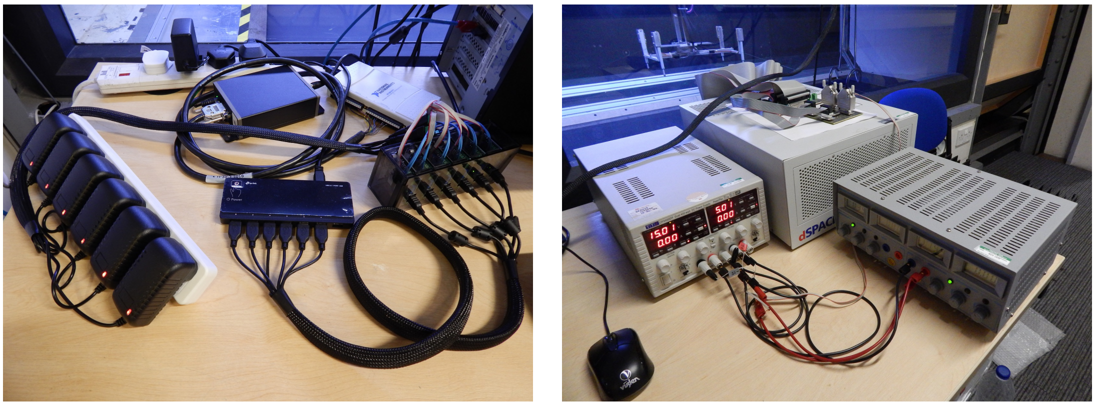


FIGURE 3.13: Schematic of the large-scale model's control and acquisition system. Power supplies and control boards are omitted for clarity.



(a) Actuator control and force acquisition equipment.

(b) Pressure acquisition equipment.

FIGURE 3.14: Acquisition and control hardware for the large-scale model.

2. Wait for the actuators to complete their motion;
3. Wait for a period of 2 s until the system settles;
4. Collect a sample of calibrated forces and moments;
5. Send a pair of channel numbers to the two pressure scanners;
6. Collect instantaneous pressure measurements;
7. Repeat steps 5 and 6 until all channels have been acquired;
8. Repeat steps 4 to 7 50 times.

3.2.6 Testing Procedure

All tests were carried out at a freestream velocity of $U_\infty = 20\,\mathrm{m/s}$, which corresponds to a length-based Reynolds number of $Re = 1.1 \times 10^6$, and was restricted by the holding limits of the actuators and the sensing range of the pressure scanners. Previous studies using similar geometries were carried out at $Re = 0.83 \times 10^6$ [99] and 2.2×10^6 [114], and most other diffuser studies used values between $Re = 1.8 \times 10^6$ and 2.7×10^6 [102–105, 111, 112].

Although Senior and Zhang [102] observed no effect on performance of varying Reynolds number between 1.3×10^6 and 2.7×10^6, Ehirim [116] noted that downforce loss when reducing ride height occurred marginally earlier at $Re = 1.4 \times 10^6$ compared to 1.8×10^6 and 2.3×10^6. Furthermore, Jowsey [123] carried out high-resolution sweeps of Reynolds number between 0.32×10^6 and 2.5×10^6 at a range of diffuser angles, and observed that whereas most diffusers experienced slow but gradual downforce increase with increasing Reynolds number, 22° and 25° diffusers switched from low-downforce to high-downforce regimes at $Re = 0.8 \times 10^6$ and 1.8×10^6 respectively. The above findings, obtained without fixed transition, are consistent with the boundary layer theory, which predicts a smaller boundary layer thickness at higher Reynolds numbers, which in the case of diffuser flow would decrease the effective diffuser area ratio, and hence reduce the adverse pressure gradient and delay stall phenomena.

Because of the relatively low Reynolds number used in this study, it is expected that those phenomena may occur at slightly higher ride heights or lower diffuser angles than in other studies. In order to reduce the potential sensitivity, transition on the underside of the body was fixed with a 0.35 mm-diameter wire, fixed to the underfloor with double-sided tape, 16 mm downstream of the leading edge of the underfloor, or 80 mm downstream of the front of the model ($x/L_M = 0.1$).

The purpose of this investigation was to establish the effect of quasi-static variations of ride height, rake and diffuser angles, therefore sweeps of all three degrees of freedom were carried out, each for several values of the remaining two DoFs, as outlined in Table 3.3. The sweeps were carried out at known configurations of interest, established through initial low-fidelity sampling and using previously published results. Each sweep was carried out in both directions in order to establish the extent of hysteresis in the system. Full force and pressure measurements were taken at every data point.

Furthermore, a datum configuration was tested at the beginning and at the end of each sweep, to track long-term repeatability of the force and moment measurements. The datum was defined as the maximum ride height ($h_1 = 90$ mm) and zero rake and diffuser angles, analogously to the small-scale experiments.

TABLE 3.3: Test matrix for the large-scale diffuser experiments.

DoF	Sweeps	Resolution	At
h_1	$90 \rightarrow 10 \rightarrow 90$ mm	4.2 mm	$\gamma = 0.0°, 1.8°, 4.8°$ $\theta = 0.0°, 6.2°, 11.2°, 16.3°, 24.7°$
γ	$0.0° \rightarrow 4.8° \rightarrow 0.0°$	0.7°	$h_1 = 10, 20, 40, 60$ mm $\theta = 0.0°, 6.2°, 11.2°, 16.3°, 24.7°$
θ	$0.0° \rightarrow 38.5° \rightarrow 0.0°$	3.0°	$h_1 = 10, 20, 40, 60$ mm $\gamma = 0.0°, 1.8°, 4.8°$

3.2.7 Data Processing

Plotting the datum measurements over time showed significant drift of all 6 force and moment components. It was assumed that this drift was due to vertical shifts of the voltage vs force response curves of the force transducer, i.e. if two load cases were tested at two points in time, the absolute values of the measured voltages would change, but the differences in voltages between the two cases would remain constant.

In order to correct this drift, voltages were reconstructed for every measurement by dividing the calibrated loads by the calibration matrix, and then corrected using the following formula:

$$\begin{aligned}\hat{V_i} = V_i &- (V_{i,\text{datum,prev.}} - V_{i,\text{datum,ref.}}) \\ &- (V_{i,\text{datum,next}} - V_{i,\text{datum,prev.}}) \cdot \frac{t_{\text{cur.}} - t_{\text{prev.}}}{t_{\text{next}} - t_{\text{prev.}}}\,,\end{aligned} \tag{3.3}$$

where V_i and $\hat{V_i}$ are the original and updated voltage measurements for the load i, $V_{i,\text{datum,ref.}}$ is the reference datum measurement, $V_{i,\text{datum,prev.}}$ and $V_{i,\text{datum,next}}$ are the last datum measurement before the current measurement and the first datum after current, $t_{\text{cur.}}$ is the current measurement timestamp, and $t_{\text{prev.}}$ and t_{next} are timestamps of the previous and next datums. The reference datum is different for every configuration (full sidewalls, full sidewalls with closed cavity, sliding sidewalls) due to the different loading conditions, and was arbitrarily chosen as the first measurement of the first ride height sweep for each configuration.

First, response curve shift between the reference datum and the previous datum, expressed as $(V_{i,\text{datum,prev.}} - V_{i,\text{datum,ref.}})$ in Eq. 3.3, was subtracted from the current measurement. Then, the shift between the previous and next datums was interpolated linearly over time, using time stamps of each measurement. This ensured that, within a single model configuration, all measurements were corrected to a single reference. However, the potential shift between that reference and the transducer's true calibration was not corrected, as the transducer was not calibrated in situ. Furthermore, potential shift between references for the three configurations were also not corrected, which increases the uncertainty when directly comparing measurements for different configurations.

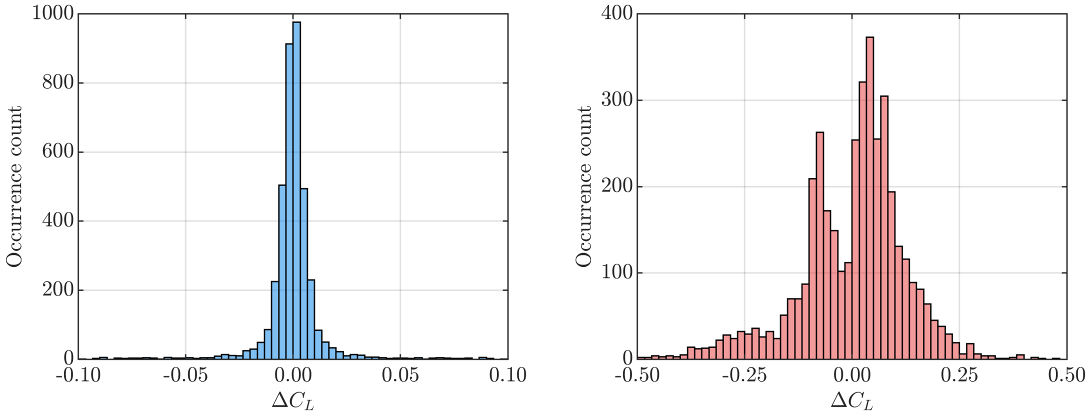
Wind-off tare measurements with the model installed were taken at a set of ride heights and rake and diffuser angles, in order to take into account the shifting centre of mass of the model. The data was then interpolated and subtracted from each wind-on measurement. Next, this procedure was repeated with the model removed, to establish the aerodynamic loads generated by the supporting frame, for the entire range of ride height actuator extensions. Both total and frame-only aerodynamic loads were then divided by freestream dynamic pressure q_∞, taken from a pitot-static tube at the time of each measurement, obtaining $C_F S$ and $C_M S L_M$ forms, where C_F and C_M are force and moment coefficients respectively. The frame-only loads were then subtracted from the total loads, and finally divided by S and SL_M to obtain aerodynamic force and moment coefficients of the model respectively. This procedure, as opposed to simply subtracting frame-only loads from total loads and dividing the result by $q_\infty S$ and $q_\infty S L_M$, was used to take into account the difference in atmospheric conditions (q_∞) between frame-only and full-model measurements. Tare measurements were not corrected for drift due to the different loading conditions in each type of measurement.

Pressure readings were calibrated using temperature taken at the time of each measurement, interpolating between the 11.3 °C and 21.7 °C calibration curves, and subsequently non-dimensionalised by q_∞. Analogously to the small-scale experiments, ride height h_1 was non-dimensionalised by model height H, and h_1/d was indicated on secondary axes where possible.

3.2.8 Measurement Repeatability

Similar to the small-scale model, measurement repeatability of the large-scale system was established through comparison of repeated data points. Data from all ride height, rake and diffuser sweeps was used, and histograms of deviations of downforce measurements from the mean, $\Delta C_L = C_L - \mu_{C_L}$, are shown in Fig. 3.15.

Figure 3.15a was plotted using means calculated for particular model configurations (full-sidewall, full-sidewall with closed tail cavity, sliding-sidewall), i.e. the same diffuser configuration (the same ride height, rake and diffuser angles) in different model configurations constituted different data points, and had separate mean values. However, the histogram also includes aerodynamic hysteresis, i.e. differences in performance depending on the direction of ride height, rake and diffuser angle sweeps. The standard deviation of the 3997 points is equal to $\sigma_{C_L} = 0.0354$, which suggests significantly better repeatability than for the small-scale system, owing partially to the improved rigidity and manufacturing precision of the model and supporting frame. However, the vast majority of the repeated points were obtained within minutes of each other within the same sweep, therefore this value only represents short-term repeatability.



(a) Comparison to mean of the same configuration.

(b) Comparison to mean of all configurations.

FIGURE 3.15: Repeatability of downforce measurements of the large-scale model based on repeated data points.

The histogram in Fig. 3.15b, on the other hand, was constructed from means irrespective of model configuration. The standard deviation is significantly higher at $\sigma_{C_L} = 0.1286$, which highlights the differences in downforce performance depending on the sidewall configuration. A small number of points were present outside of plot bounds in both figures (2.23% and 0.45% respectively), which were omitted for clarity. Those points correspond to configurations where significant hysteresis occurred, as discussed further in Sections 6.1 and 6.2.

Chapter 4

Hardware-in-the-Loop Aerodynamic Optimisation

This chapter presents an overview of the algorithm configurations and testing procedures used during hardware-in-the-loop optimisation, followed by a discussion of the results, which were obtained using the small-scale experimental system described in Section 3.1. The first main objective of the wind tunnel-based HIL tests was to test several distinctive optimisation methods in various configurations, and to inspect their effectiveness on a problem with inherent noise, hysteresis, and long-term measurement drift. Crucially, the aim was not to provide a comprehensive algorithm survey or to obtain an optimised geometry of the diffuser model, but rather to test a sample of well-established algorithms, and to identify which of their traits are desirable for problems of this type. The algorithms tested were pattern search, simulated annealing, particle swarm optimisation and two implementations of the genetic algorithm, and the results of the comparative tests are discussed in the second section of this chapter. Next, a brief study of the settling and sampling times is presented, which was aimed at reducing the overall operational time of HIL optimisation by shortening function evaluation, while maintaining a similar convergence rate.

4.1 Methodology

4.1.1 Algorithm Configurations

As previously outlined in detail in Section 2.1.3, four distinctive algorithms were used for the study: the genetic algorithm, simulated annealing and pattern search from

MATLAB's Global Optimization Toolbox [47], as well as particle swarm optimisation as implemented by Biswas [66]. Although hybrid methods could provide superior optimisation performance, their more intricate behaviour would make identification of advantageous traits more difficult. Instead, emphasis was placed on testing simple methods with distinctive features, in order to link particular aspects of their performance with their individual characteristics.

Each algorithm was applied to the real-time optimisation system in three configurations: the default, serving as a validated reference, and two alternatives, except pattern search, for which only one alternative was tested. The additional configurations were selected through simulated optimisation runs, which used interpolated performance maps of the small-scale diffuser model instead of real-time measurements. For each algorithm, the second configuration is the one that demonstrated the most effective convergence, and the third is a configuration that exhibited some distinctive convergence characteristics, e.g. a more global or local tendency. None of the tested pattern search configurations yielded a significant performance improvement over the default, so only one alternative was selected for HIL testing, with complete, rather than opportunistic polling. The default settings and detail on the functioning of the algorithms are explained in their respective documentation [47, 66], whereas the configurations used in this study are presented in Tables 4.1–4.4.

As multiple parameters were changed in each of the alternative configurations, it was not possible to establish quantitative relationships between the particular parameters and algorithm performance. However, detailed analysis enabled qualitative links to be made, which are outlined in Section 4.2.

Table 4.1: Configurations of the integer-encoded genetic algorithm.

Setting	Configuration 1 (default)	Configuration 2	Configuration 3
Population size	50	70	30
Fitness scaling function	Rank	Rank	Top
Elite size	3	0	3
Selection function	SUS	SUS	Tournament
Crossover function	Uniform	Heuristic	Heuristic
Crossover fraction	0.8	0.9	0.9
Mutation function	Adaptive feasible	Adaptive feasible	Uniform

Table 4.2: Configurations of the particle swarm optimisation algorithm.

Setting	Configuration 1 (default)	Configuration 2	Configuration 3
Swarm size	30	20	50
Cognitive factor	2	1.5	2.5
Social factor	2	2.5	1.5
Inertia weight	0.9 to 0.4	0.9 to 0.4	0.9 to 0.4

Table 4.3: Configurations of the simulated annealing algorithm.

Setting	Configuration 1 (default)	Configuration 2	Configuration 3
Annealing method	Fast	Fast	Boltzmann
Reannealing period	100	20	100
Initial temperature	100	150	100
Temperature function	Exponential	Boltzmann	Fast

Table 4.4: Configurations of the pattern search algorithm.

Setting	Configuration 1 (default)	Configuration 2
Initial mesh size	1	1
Mesh expansion factor	2	2
Mesh contraction factor	0.5	0.5
Polling method	GPS pos. basis 2N	MADS pos. basis 2N
Complete polling	off	on

4.1.2 Binary-Encoded Genetic Algorithm

In addition to the four algorithm implementations mentioned above, a second genetic algorithm was coded specifically for this study, in order to provide greater control over the algorithm's functioning, as well as for comparison between the two implementations. Unlike the MATLAB version, the custom algorithm was binary-encoded, i.e. all individuals were converted to binary strings for crossover and mutation, and converted back to integers before fitness evaluation. The discretisation of the diffuser design space, dictated by the spatial resolution of the actuators, required 21-bit strings in order to describe each configuration. The three degrees of freedom, viz. ride height, rake and diffuser angles, had 231, 41 and 107 discrete states, necessitating 8, 6 and 7 bits respectively. Furthermore, crossover and mutation were applied to the entire population with their respective probabilities, therefore some individuals could undergo both operations, as opposed to the integer-encoded implementation, where each individual could undergo only one of these operations. A large number of configurations of the binary-encoded GA were tested on sampled data and three configurations with clearly distinctive behaviour were selected, similar to the other algorithms. The settings used in the hardware-in-the-loop tests are presented in Table 4.5.

Table 4.5: Configurations of the binary-encoded genetic algorithm.

Setting	Configuration 1 (default)	Configuration 2	Configuration 3
Population size	50	30	70
Initialisation	Random	Random	Random
Fitness scaling function	Linear	Exponential	Linear
Elite size	3	3	3
Selection function	Roulette	Roulette	Tournament
Crossover function	Single-point	Two-point	Uniform
Crossover rate	0.8	0.8	0.8
Mutation function	Unbiased	Biased towards less-fit individuals	Unbiased
Mutation rate	0.05	0.08	0.1

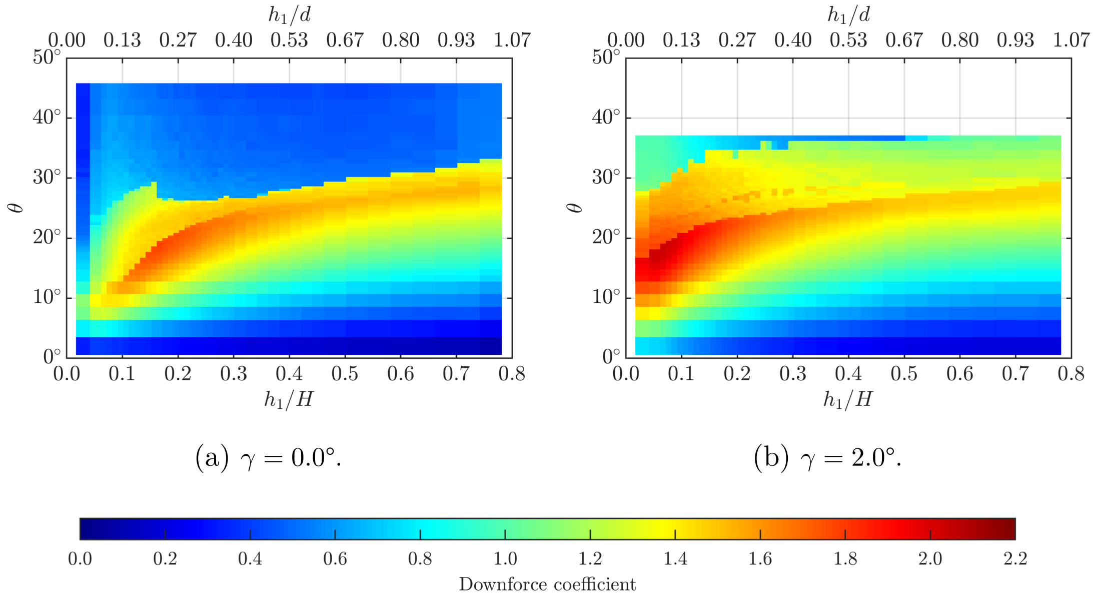
4.1.3 Testing Procedure

Every configuration from Tables 4.1 to 4.5 was tested three times, in order to obtain average convergence characteristics and to qualitatively compare their effectiveness, with the simulated annealing and pattern search algorithms initialised from the midpoint of the search space. All optimisation trials were terminated after an arbitrary limit of 60 min to assess the efficiency of the algorithms, with the exception of pattern search, which consistently converged before reaching this limit.

The fitness function, or the cost function, is the function that the optimisation algorithm attempts to maximise or minimise, depending on the goal. The measurements that could be obtained in the wind tunnel using the small-scale system were drag force F_x, side force F_y, and downforce F_z, as well as the three respective torques: rolling moment M_x, pitching moment M_y, and yawing moment M_z. As all 6 loads were acquired in real-time at a high frequency, the optimisation goal could be for any combination of them, for example maximising downforce (F_z), minimising drag (F_x), maximising aerodynamic efficiency (F_z/F_x), minimising lateral loads (F_y, M_x, or M_z), etc. However, the focus of the hardware-in-the-loop optimisation study was investigation of the characteristics of this type of optimisation in response to various factors, rather than attaining a particular optimal geometry. Therefore, for the purpose of simplicity, maximising downforce (F_z) was selected as the objective for all optimisation runs.

In order to contextualise the discussion of these tests, the chosen objective function is plotted on two slices of the three-dimensional search space in Fig. 4.1. The plots were obtained by gradually increasing the diffuser angle at a range of ride heights and rake angles, and illustrate the trends in downforce experienced by the small-scale diffuser model. Large parts of the search space comprise monotonic variations, which constitute a simple optimisation problem. However, several discontinuities are present with respect to diffuser angle, as well as minor local maxima, both of which add complexity to the problem. Furthermore, despite a sampling time of 2 s, experimental noise is also present in the measurements, most clearly visible in the low-gradient areas at high diffuser angles in Fig. 4.1a and 4.1b.

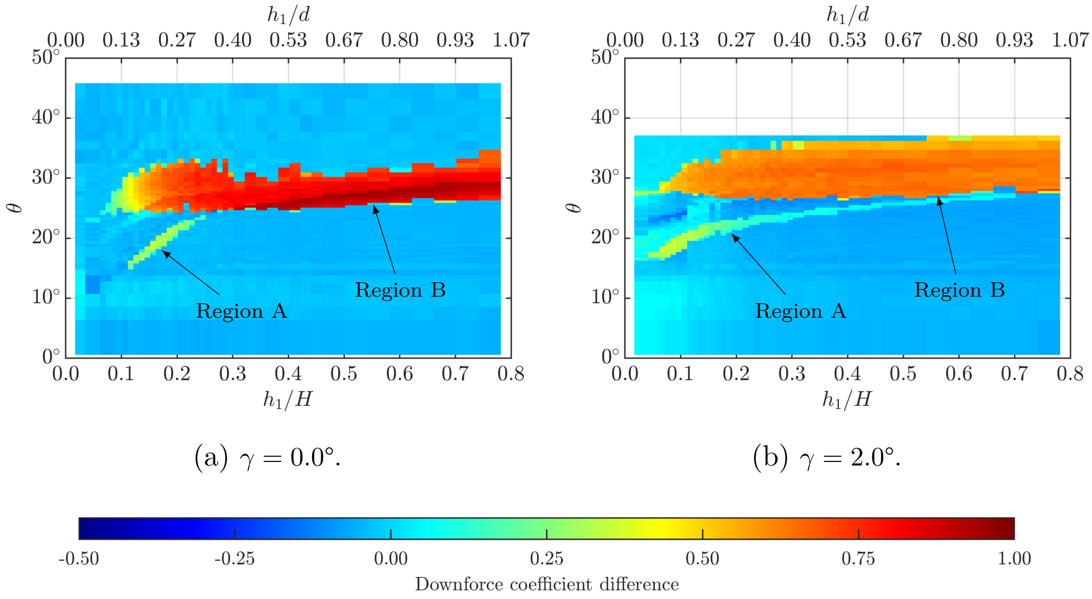
Another characteristic of this problem is aerodynamic hysteresis, where the forces generated by the body depend on the configuration or orientation of the body prior to its current state. In this case, gradually decreasing the diffuser angle results in different downforce values than increasing it, even if the measurement is taken at the same diffuser angle. This phenomenon, which is a known feature of automotive diffuser aerodynamics, is illustrated in Fig. 4.2, where the difference in downforce depending on the direction of diffuser angle variations is plotted for the same two slices of the



(a) $\gamma = 0.0°$.

(b) $\gamma = 2.0°$.

FIGURE 4.1: Surface plots of downforce coefficient of the small-scale diffuser model, sampled with increasing diffuser angle for each ride height. The plots correspond to two slices of the three-dimensional search space.



(a) $\gamma = 0.0°$.

(b) $\gamma = 2.0°$.

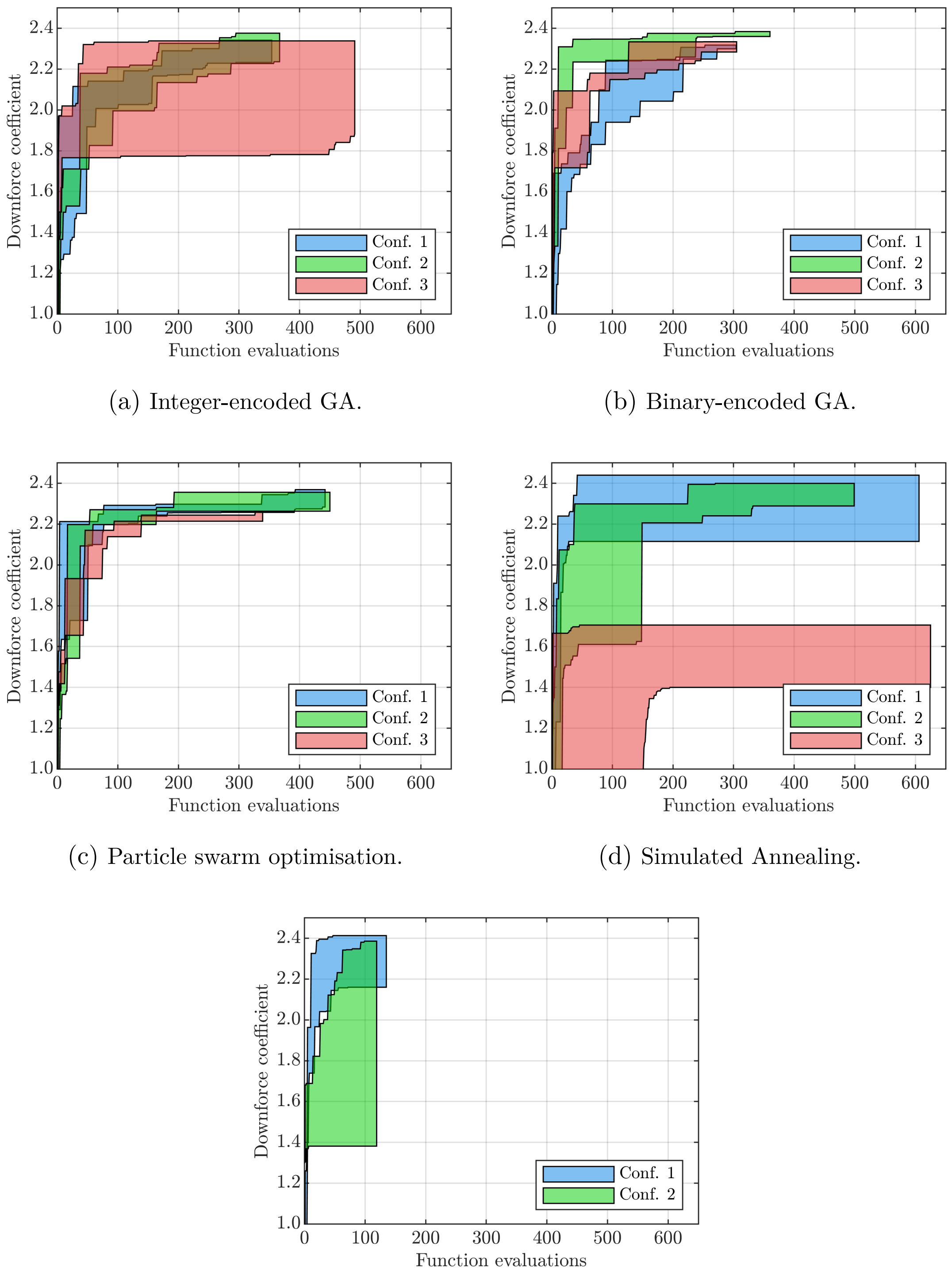
FIGURE 4.2: Surface plots of hysteresis of downforce coefficient $C_L|_{\theta\nearrow} - C_L|_{\theta\searrow}$ of the small-scale diffuser model, showing the difference in downforce depending on the direction of diffuser angle variations.

search space as in Fig. 4.1. The downforce coefficient difference is defined as C_L when diffuser angle is being increased ($C_L\big|_{\theta\nearrow}$) minus C_L when it is being reduced ($C_L\big|_{\theta\searrow}$). The two distinct trends seen in both figures, referred to as regions A and B, are caused by shifts of the discrete changes in downforce from Fig. 4.1, and are discussed in detail in Chapter 6.

In the case of hardware-in-the-loop optimisation, aerodynamic hysteresis might result in different fitness values from repeated evaluations of a particular point in the search space, depending on the location of the preceding point. Furthermore, whereas Fig. 4.2 illustrates hysteresis with respect to diffuser angle, similar trends are expected to occur with respect to all three degrees of freedom. This significantly increases the complexity of the problem, posing an additional challenge for the optimisation algorithms.

4.2 Comparative Optimisation Tests

Figure 4.3 illustrates the convergence of the HIL optimisation runs carried out for this part of the study. For each algorithm, tests using all configurations from Tables 4.1 to 4.5 are shown, and each configuration's translucent plot is defined by the maximum and minimum fitness (C_L) reached by a given function evaluation among its 3 trials. Apart from showing the pace of progression of the algorithms towards higher fitness,



(a) Integer-encoded GA.

(b) Binary-encoded GA.

(c) Particle swarm optimisation.

(d) Simulated Annealing.

(e) Pattern search.

FIGURE 4.3: Algorithm convergence during the 60 min optimisation runs. 3 trials per algorithm configuration.

these plots also indicate the consistency of each optimisation method and configuration, with wide plots suggesting the algorithm's performance is susceptible to chance and vice versa. The maximum fitness values $C_{L_{\max}}$ attained by each algorithm, as well as the corresponding model configurations, are summarised in Table 4.6.

TABLE 4.6: Maximum-downforce configurations attained by each algorithm.

Algorithm	$C_{L_{\max}}$	h_1/H	γ	θ
Integer-encoded GA	2.3764	0.080	4.96°	24.63°
Binary-encoded GA	2.3844	0.073	4.96°	24.63°
Particle swarm optimisation	2.3692	0.070	5.28°	23.81°
Pattern search	2.4180	0.073	4.96°	25.03°
Simulated annealing	2.4400	0.080	4.96°	25.03°

Figure 4.3a shows the progression of the integer-encoded genetic algorithm. The first two configurations exhibit similar performance; although configuration 2 reached a slightly higher downforce value in at least one run, it was also slightly less consistent, reaching lower values after certain numbers of evaluations. Configuration 3 is the least consistent, and failed to locate the global maximum in at least one run, despite the greater number of function evaluations carried out within the 60 min time limit. The large number of evaluations and inconsistent convergence are indications of premature convergence, or too rapid a shift of the population towards a small region of the search space. This was likely caused by the tournament selection, which tends to favour a small proportion of fittest individuals in every generation.

Next, Fig. 4.3b displays the convergence of the binary-encoded GA. Whereas its performance in late stages of optimisation was similar for all configurations, the shift towards high fitness in the first few generations strongly depended on the settings. Configuration 2 consistently found the high fitness region fastest, but it retained sufficiently exploratory characteristics to avoid premature convergence, and located the highest fitness values in all runs. The remaining configurations converged slower, but also consistently located the global maximum. The fast convergence of configuration 2 was likely caused by the exponential fitness scaling, which favoured high-performance individuals, while mutation biased towards less-fit individuals ensured sufficient exploration.

Particle swarm optimisation is the subject of Fig. 4.3c, and exhibits marginal differences between its configurations. All settings provided fast and consistent convergence, and allowed approximately 450 function evaluations within the 60 min time limit—

significantly more than the well-performing configurations of both genetic algorithms.

The progression of simulated annealing is shown in Fig. 4.3d, with all the configurations exhibiting significantly different behaviour. Although configuration 1 attained the overall highest value of downforce, it did not do so consistently. Configuration 2 was slightly more consistent, likely thanks to the significantly shorter reannealing period, which allowed it to broaden its search more times during the permitted 60 minutes. Finally, configuration 3 failed to locate the global maximum in all 3 runs. This configuration used the fast temperature function, which reduces temperature much quicker than the other methods (see Fig. 4.4), leading to premature tightening of the search.

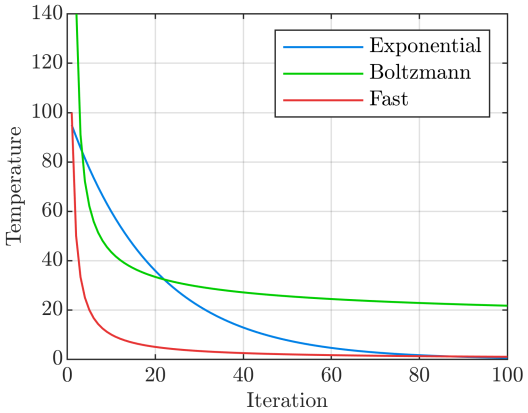


FIGURE 4.4: Comparison of the simulated annealing temperature functions used in the hardware-in-the-loop optimisation runs.

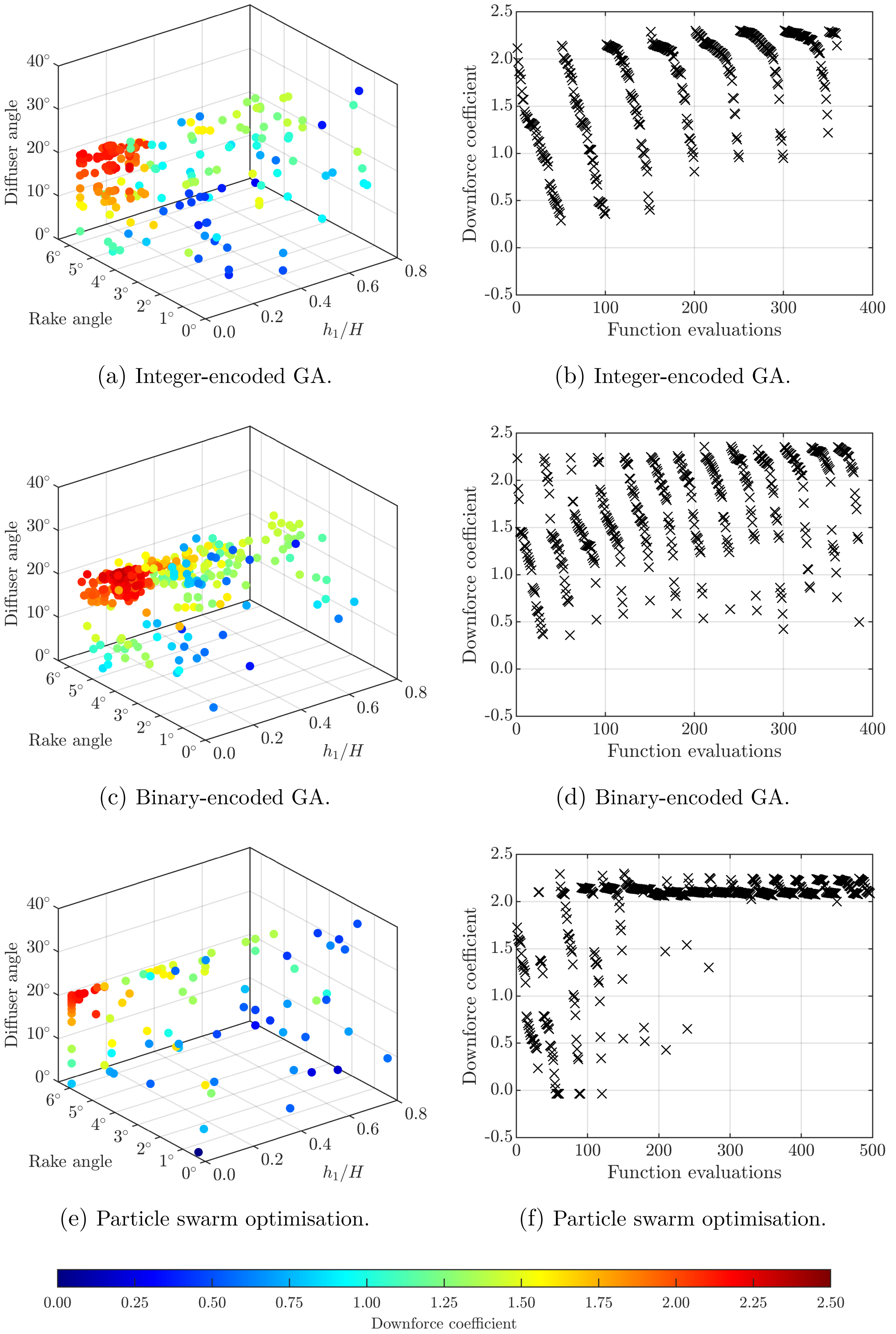
Finally, the plots in Fig. 4.3e illustrate the convergence of pattern search, with significantly different characteristics depending on algorithm settings. Both configurations converged before the time limit was reached, but did not consistently attain the global maximum. Configuration 1, which utilised opportunistic polling, can be seen to progress very fast in the early stages, but stagnated after about 50 evaluations. When complete polling was used, the rate of convergence was significantly slower, and the algorithm got stuck in a region of very low downforce in one instance. This trend suggests that opportunistic polling is more efficient, without making the algorithm more prone to premature convergence. However, a larger number of runs would be needed to quantify the difference in performance.

In order to better understand the characteristics of the algorithms, and to identify the causes of the trends outlined above, selected runs that attained the global optimum, one for each of the 5 algorithms, are illustrated in greater detail in Fig. 4.5 and 4.6. All presented runs used the default configuration (configuration 1) of their respective algorithms, apart from the binary-encoded algorithm, where configuration 2 was selected. The left-hand side plots in both figures present scatter plots of the diffuser configurations tested by the algorithms, displayed in the three-dimensional search space and coloured by fitness. The right-hand side plots display progression plots, where fitness of consecutive diffuser configurations is plotted against function evaluation, additionally sorted within individual generations in the case of population-based algorithms.

4.2.1 Population-Based Algorithms

The scatter and progression plots in Fig. 4.5 illustrate the performance of the genetic and particle swarm optimisation algorithms. The integer-encoded GA can be seen to converge towards the region of maximum downforce effectively, placing relatively few points in the low-performance regions, but carrying out enough exploration to avoid falling into a local maximum. However, its behaviour is somewhat systematic, with many points appearing in lines, even within the high-performance region, which does not appear to be explored thoroughly. The progression plot in Fig. 4.5b confirms that the algorithm shifted most of the population to the high-downforce region quickly, while still maintaining genetic diversity, as evidenced by the vertical spread of each generation. The scatter plot of the binary-encoded GA in Fig. 4.5c shows a more concentrated distribution of points, with a large, coherent structure in the high-performance region. Figure 4.5d shows that, with a smaller population size than the integer-encoded GA, the binary-encoded algorithm demonstrates gradual and seemingly more effective convergence onto the high-performance region, while also maintaining significant diversity.

The different encoding of the two implementations fundamentally changes the way the algorithms operate. The integer-encoded GA only exchanges values of particular variables during crossover, and occasionally mutates some of the variables, which results in numerous duplicates or similar points (e.g. differing by only one variable and hence forming a line). As a result, on average only 61.4% of the diffuser configurations tested by configuration 1 of the algorithm were unique. The binary-encoded GA, on the other hand, constantly breaks down variables by splitting the binary strings for crossover, resulting in a more random yet concentrated distribution of data points, as well as fewer duplicates, with an average of 81.9% unique points across the 3 runs using configuration 2. However, these discrepancies are emphasised due to the low number of DoFs; usually at least one of the three variables is broken down in each crossover operation in the binary-encoded algorithm, and this ratio would be much smaller for



(a) Integer-encoded GA.

(b) Integer-encoded GA.

(c) Binary-encoded GA.

(d) Binary-encoded GA.

(e) Particle swarm optimisation.

(f) Particle swarm optimisation.

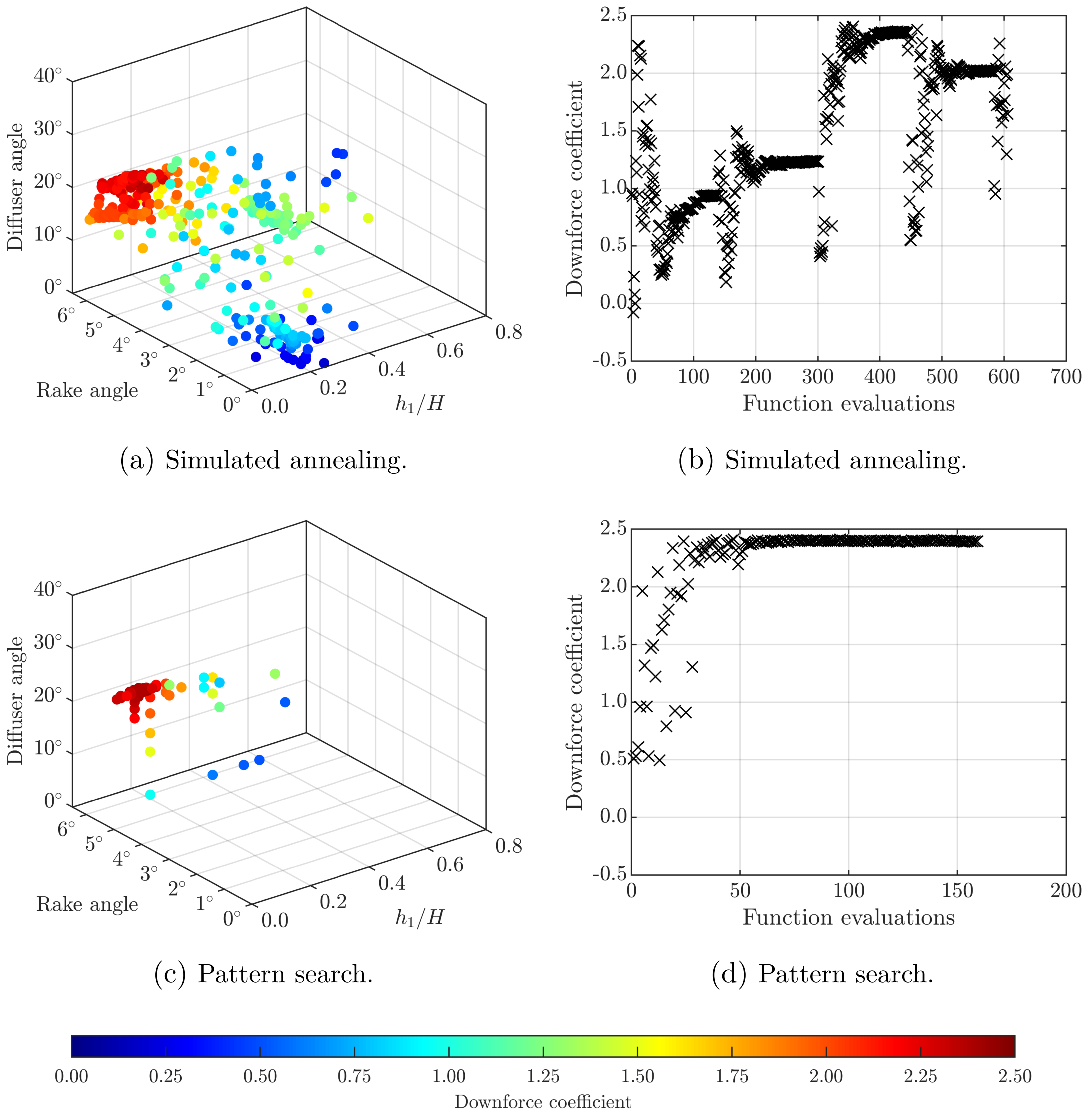
FIGURE 4.5: Example scatter plots (left-hand side) and progression plots sorted within individual generations (right-hand side) of the population-based algorithms, obtained from the HIL optimisation runs.

problems with more DoFs. Notably, neither implementation was significantly affected by noise or hysteresis due to their population-based nature, locating the global optimum in all runs except configuration 3 of the integer-encoded GA.

The behaviour of the particle swarm optimisation algorithm, seen in Fig. 4.5e and 4.5f, is significantly different from that of the genetic algorithms. The progression plot reveals that, when the algorithm found a high-performance point in the second generation, it immediately started shifting the swarm in that direction, which is desirable. However, it also shows that the entire swarm was already concentrated in a very small area by the sixth generation. The scatter plot shows that most points rapidly moved to the edges of the search space near the high-fitness region, indicating excessive inertia weight, which is a numerical factor that accelerates the particles through the domain. As the values used for this parameter were in line with those suggested in literature [77, 78], it is hypothesised that the algorithm implementation used does not take the size of the search space into account, leading to excessive particle velocity. Therefore, a scaling function for the inertia weight parameter is recommended for future applications.

4.2.2 Non-Population-Based Algorithms

The convergence of simulated annealing is shown in Fig. 4.6a and 4.6b. As seen in the middle and bottom areas of Fig. 4.6a, the algorithm spent a large proportion of time probing regions of poor performance. This is confirmed by the progression plot in Fig. 4.6b, where the algorithm is seen to repeatedly converge towards low downforce values. This can be explained by the gradual transition from exploratory to exploitative search, as the simulated temperature decreases. The algorithm might get trapped in a low-performance region during this transition, and not be able to escape until reannealing occurs. As a result, unnecessary exploitation of low-quality regions might take place. Reannealing can be observed in Fig. 4.6b, where it appears as chaotic segments with large C_L differences between consecutive points. As temperature is gradually reduced, the permitted perturbation size also decreases, leading to smaller fitness variations. The large number of function evaluations (over 600, compared to less than 500 for all the other algorithms) indicates a very small average actuation distance, suggesting that temperature was reduced too quickly for the reannealing period used, not enabling sufficient exploration of the design space. Furthermore, the algorithm moved to a region of lower fitness after finding the global maximum, which suggests that the maximum was found by chance.



(a) Simulated annealing.

(b) Simulated annealing.

(c) Pattern search.

(d) Pattern search.

FIGURE 4.6: Example scatter plots (left-hand side) and progression plots (right-hand side) of the non-population-based algorithms, obtained from the HIL optimisation runs.

Testing on sampled data, confirmed by wind tunnel-based optimisation runs using configuration 2 (see Fig. 4.3d), revealed that configurations with very short reannealing periods and high temperatures statistically yield the best convergence performance. Shortening the reannealing period results in the creation of a larger number of clusters of points in the search space, whereas increasing temperature makes those clusters more dispersed. It was found that the most effective algorithm configuration resulted in a nearly uniform distribution of points around the search space, which located the optimum by chance, much like random sampling. This suggests that the simulated annealing algorithm may not be effective for experimental problems, where noise and hysteresis are both significant issues. However, it is unknown whether its effectiveness would be greater for systems with more degrees of freedom.

Similar to simulated annealing, the pattern search algorithm was initialised from the mid-point of the search space, and Fig. 4.6c illustrates its systematic convergence onto the global maximum. The strength of pattern search is its adaptive mesh size, which allows it to make very small adjustments and to converge on the optimum with high precision. However, one of the other two trials carried out using the default algorithm configuration failed to reach the maximum C_L region. This is apparent in Fig. 4.3e, where the bottom edge of the plot for configuration 1 does not exceed $C_L = 2.2$. This was likely caused by experimental noise or local maxima, which the standard variant of pattern search is not designed to cope with. The algorithm was also found to be sensitive to starting point combined with noise when tested on sampled data, which is a further hindrance for its application to experimental systems with unknown characteristics.

4.3 Sampling Time Study

This section presents the study on the impact of varying settling and sampling times on convergence during hardware-in-the-loop aerodynamic optimisation. The settling and sampling times that were used in the experimental optimisation runs discussed in the preceding section were conservatively set at 1.5 s and 2 s respectively, which on average constituted approximately 35% of the total optimisation time. However, the hitherto presented results indicate that the population-based algorithms, namely the genetic and PSO algorithms, were insensitive to noise in most configurations. Therefore, it might be possible to reduce the settling and sampling times without impeding the algorithms' ability to converge on the global optimum, while significantly reducing the time needed for each function evaluation, and hence accelerating convergence with respect to wall-clock time. This trade-off between quicker convergence and faster function evaluation was examined in detail through testing on sampled experimental data with simulated settling and sampling times, and the sections below present the results of this campaign.

4.3.1 Noise Sampling

The first phase of the study involved simulated optimisation runs using the algorithms and configurations listed in Section 4.1. Interpolated downforce maps of the small-scale diffuser model obtained in the wind tunnel were used in place of real-time wind tunnel measurements, as the latter were prohibitively time-consuming. Therefore, a method to determine the noise of wind tunnel readings was necessary to simulate the impact of different settling and sampling times on convergence.

This was achieved by sampling the three-dimensional design space of the diffuser model with a low-resolution grid, and interpolating the results. The grid was $5 \times 5 \times 5$ points in size, with the outer points at the edges of the design space, and the remaining points equally-spaced for every degree of freedom, i.e. the ride height, rake and diffuser angles. In order to obtain the standard deviation of downforce readings, 50 samples were collected in the wind tunnel at each point on the grid, resulting in a total of 6250 samples. Each sample was 10 s long, with a sampling frequency of 1000 Hz. Crucially, all samples were shuffled prior to data collection, so that the effect of aerodynamic hysteresis was taken into account, resembling real-time optimisation measurements as closely as possible. After all the samples were collected, they were grouped together according to their location in the design space.

When particular settling and sampling times were required, their values were passed to a script that trimmed each of the 6250 samples to the desired length. For example, given a settling time of $t_{\text{settling}} = 1.5\,\text{s}$ and a sampling time of $t_{\text{sampling}} = 2\,\text{s}$, only the parts between 1.5 s and 3.5 s were retained, and the remainder of the data was discarded. This corresponds to the experimental optimisation scenario, where sampling begins t_{settling} after actuation has ended, and ends after $t_{\text{settling}} + t_{\text{sampling}}$.

After the desired sample fragments were isolated, they were averaged. This resulted in 50 measurements of downforce, obtained experimentally using the desired settling and sampling times. The standard deviation of the 50 measurements was computed at each point on the $5 \times 5 \times 5$ grid, and the resulting three-dimensional matrix was linearly interpolated for each point in the design space. The poor resolution of the sampling grid means that the interpolated matrix of standard deviation did not accurately represent short-term downforce fluctuations at each point in the design space. However, this method enabled realistic simulation of experimental noise and its variations across the design space, which was the requirement for this study.

4.3.2 Simulated Optimisation

Simulated optimisation runs were carried out using a total of 73 combinations of settling and sampling times, ranging from $t_{\text{settling}} = 0\,\text{s}$ and $t_{\text{sampling}} = 0.01\,\text{s}$, to 5 s each, the latter making use of the entire 10 s-long samples. The full range of values used in the tests is shown in Table 4.7. For each combination, 100 trials were carried out using every configuration of the 5 optimisation algorithms from Tables 4.1 to 4.5, in order to obtain reliable average results. Importantly, simulated annealing and pattern search were initialised from random locations in the search space in every trial, in contrast to the comparative HIL trials discussed in Section 4.2. Random sampling was also tested for comparison purposes, yielding a total of 116,800 simulated optimisation runs.

Table 4.7: Combinations of settling and sampling times used in the simulated optimisation runs.

	t_{settling}									
t_{sampling}	0 s	0.01 s	0.05 s	0.1 s	0.2 s	0.5 s	1 s	1.5 s (baseline)	2 s	5 s
0.01 s	●	●	●	●	●	●	●		●	●
0.05 s	●	●	●	●	●	●	●		●	●
0.1 s	●	●	●	●	●	●	●		●	●
0.2 s	●	●	●	●	●	●	●		●	●
0.5 s	●	●	●	●	●	●	●		●	●
1 s	●	●	●	●	●	●	●		●	●
2 s (baseline)	●	●	●	●	●	●	●	●	●	●
5 s	●	●	●	●	●	●	●		●	●

When trials using given t_{settling} and t_{sampling} were started, the interpolated matrix of standard deviation of downforce was obtained using the procedure explained in Section 4.3.1. During the optimisation runs, the fitness function sourced mean downforce values from the interpolated downforce maps of the small-scale diffuser (see Fig 4.1). The mean and the standard deviation were used to create a normal probability distribution function of downforce, subsequently converted into a cumulative distribution function (CDF). The value of downforce that was passed onto the optimisation algorithm as fitness was determined by generating a random number between 0 and 1, and finding the corresponding value of downforce on the cumulative distribution function. An example is shown in Fig. 4.7, where a CDF is plotted for a mean downforce of $\mu_{F_z} = 6\,\text{N}$ with a standard deviation of $\sigma_{F_z} = 0.5\,\text{N}$.

During all optimisation trials, corresponding actuator positions, original mean downforce and generated noise-induced downforce were saved at each function evaluation. Actuation distance between configurations $n-1$ and n was calculated for every function evaluation starting from $n = 2$, and then converted to actuation time. This was done using data from experimental validation tests (see Section 4.3.4), where actuation distance and measured actuation time were compared to determine the speed of every actuator and additional overheads. The resultant actuation time, $t_{\text{actuation}}$, was added

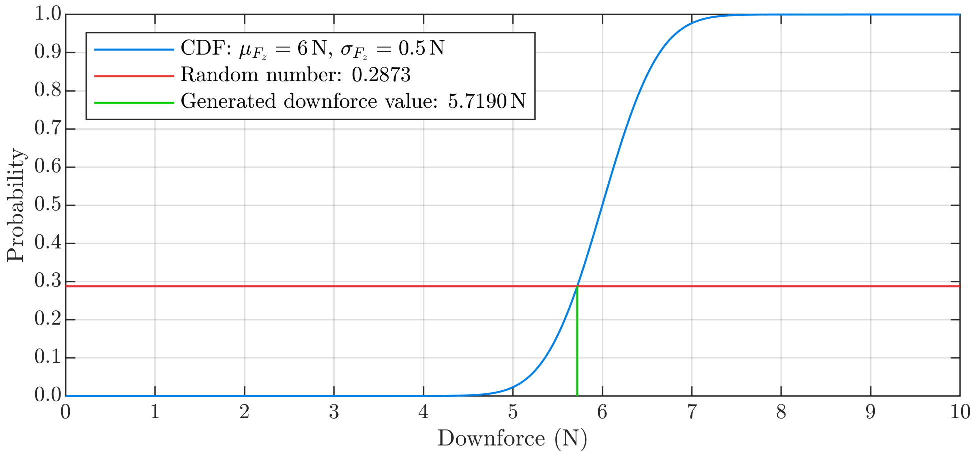


FIGURE 4.7: An example of a cumulative distribution function, calculated using a mean downforce of $\mu_{F_z} = 6\,\mathrm{N}$ and a standard deviation of $\sigma_{F_z} = 0.5\,\mathrm{N}$. The resultant randomly-generated downforce value is equal to $F_z = 5.7190\,\mathrm{N}$.

to the settling and sampling times, to obtain the total time that would be required for the particular function evaluation if carried out in the wind tunnel. This time was added to the optimisation time from the previous function evaluation $(n-1)$, to obtain simulated optimisation time t_{op} at the current evaluation (n):

$$t_{\mathrm{op}}\Big|_{n} = t_{\mathrm{op}}\Big|_{n-1} + t_{\mathrm{actuation}} + t_{\mathrm{settling}} + t_{\mathrm{sampling}}\,. \tag{4.1}$$

The optimisation runs were terminated only when t_{op} exceeded 60 min and at least 1000 function evaluations had been carried out. The 60 min time limit corresponds to the limit applied during the hardware-in-the-loop optimisation runs (see Section 4.2), whereas the second condition was imposed to provide sufficient comparison of convergence speed between configurations, as the configuration with the highest t_{settling} and t_{sampling} carried out fewer than 200 evaluations before 60 min elapsed. Two additional columns of data were saved from each optimisation run: maximum fitness and optimisation time, where the n^{th} element in these columns represented the maximum fitness after n function evaluations and the optimisation time from the start until function evaluation n respectively.

Importantly, short settling and sampling times resulted in higher standard deviation of downforce, and therefore higher noise. This increased noise magnitude meant that optimisation runs using short settling and sampling times occasionally acquired values of downforce higher than the mean, misrepresenting the actual convergence performance of the algorithms. In order to remove this bias from the analysis, the downforce values used in data visualisation were the original mean values, rather than the noise-induced values that were passed to the optimisation algorithms as fitness.

In order to determine the average performance of each algorithm using each settling and sampling time combination, the column of maximum fitness at each function evaluation (with downforce values without noise, as mentioned above) was linearly interpolated onto a uniform time grid, which had points every 0.1 s between 0 min and 60 min, using the calculated columns of optimisation time t_{op}. Optimisation runs that converged before 60 min elapsed had their maximum fitness extrapolated to the time limit using the last available value. After all trials were interpolated, the values of maximum fitness at each point on the time grid were averaged. This resulted in two columns for each combination of optimisation algorithm, its configuration, and settling and sampling times: the time grid, and corresponding mean maximum fitness.

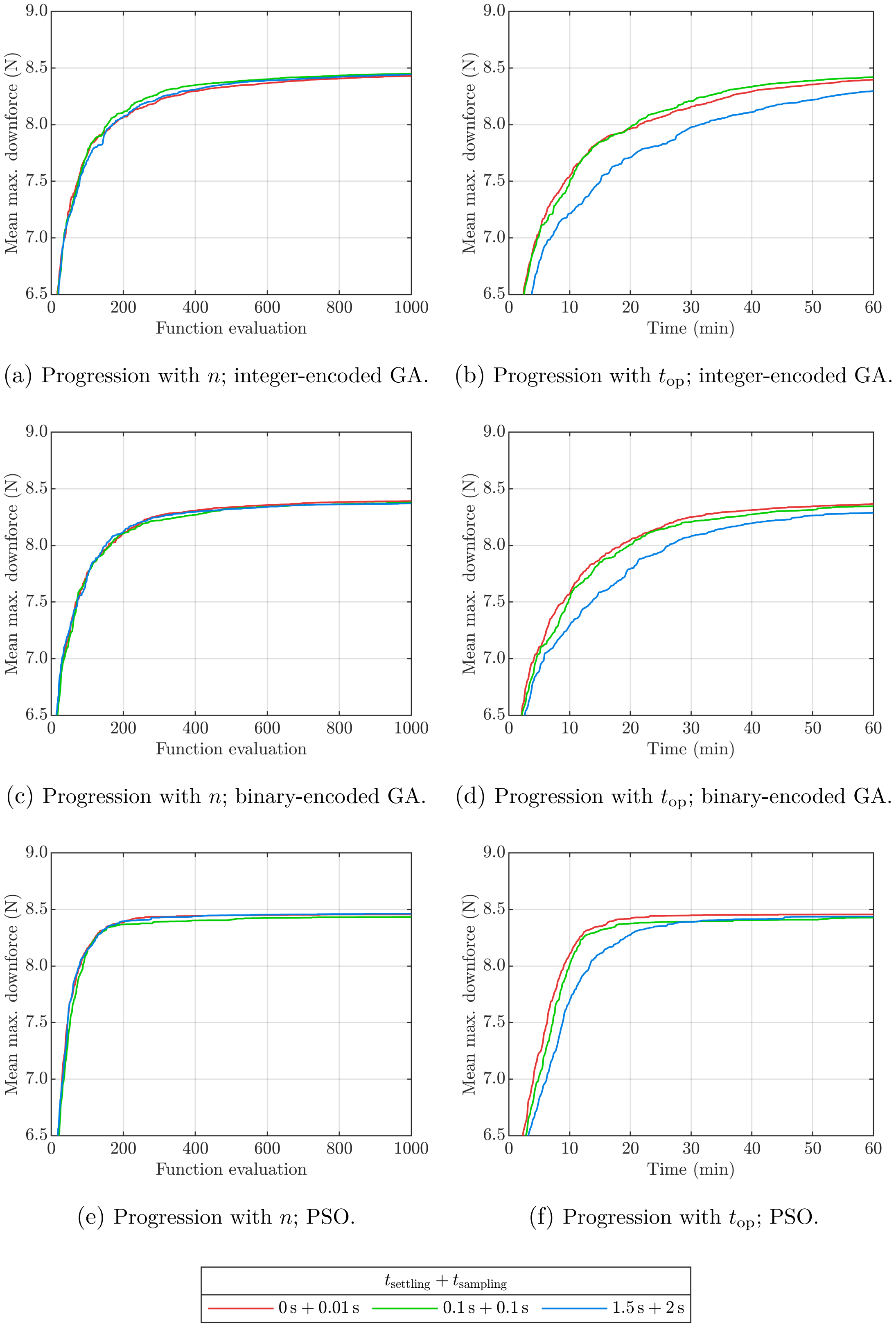
4.3.3 Simulation Results

The progression of mean maximum fitness for different settling and sampling times is compared in two ways. Firstly, with respect to function evaluation, which reveals whether settling and sampling times had any effect on the convergence speed of the algorithms, and secondly, with respect to optimisation time, which also takes into account the changing function evaluation length. Only the best-performing configuration of each algorithm was used for visualisation (configuration 1 for pattern search, and configuration 2 for the other algorithms—see Tables 4.1 to 4.5), and three distinctive settling and sampling time combinations:

- Baseline: $t_{\text{settling}} = 1.5\,\text{s}$, $t_{\text{sampling}} = 2\,\text{s}$;
- Reduced: $t_{\text{settling}} = 0.1\,\text{s}$, $t_{\text{sampling}} = 0.1\,\text{s}$;
- And minimal: $t_{\text{settling}} = 0\,\text{s}$, $t_{\text{sampling}} = 0.01\,\text{s}$.

Figures 4.8 and 4.9 present comparisons of progression of mean maximum downforce using the three combinations, for population-based and non-population-based algorithms respectively. The plots on the left- and ride-hand sides in both figures are with respect to function evaluation and optimisation time respectively. Random sampling serves as a reference, as measurement noise does not affect its progression. This is confirmed by Fig. 4.9e, which exhibits negligible differences in maximum downforce progression between the three combinations.

Crucially, the genetic algorithms (Fig. 4.8a and 4.8c) and particle swarm optimisation



(a) Progression with n; integer-encoded GA.

(b) Progression with t_{op}; integer-encoded GA.

(c) Progression with n; binary-encoded GA.

(d) Progression with t_{op}; binary-encoded GA.

(e) Progression with n; PSO.

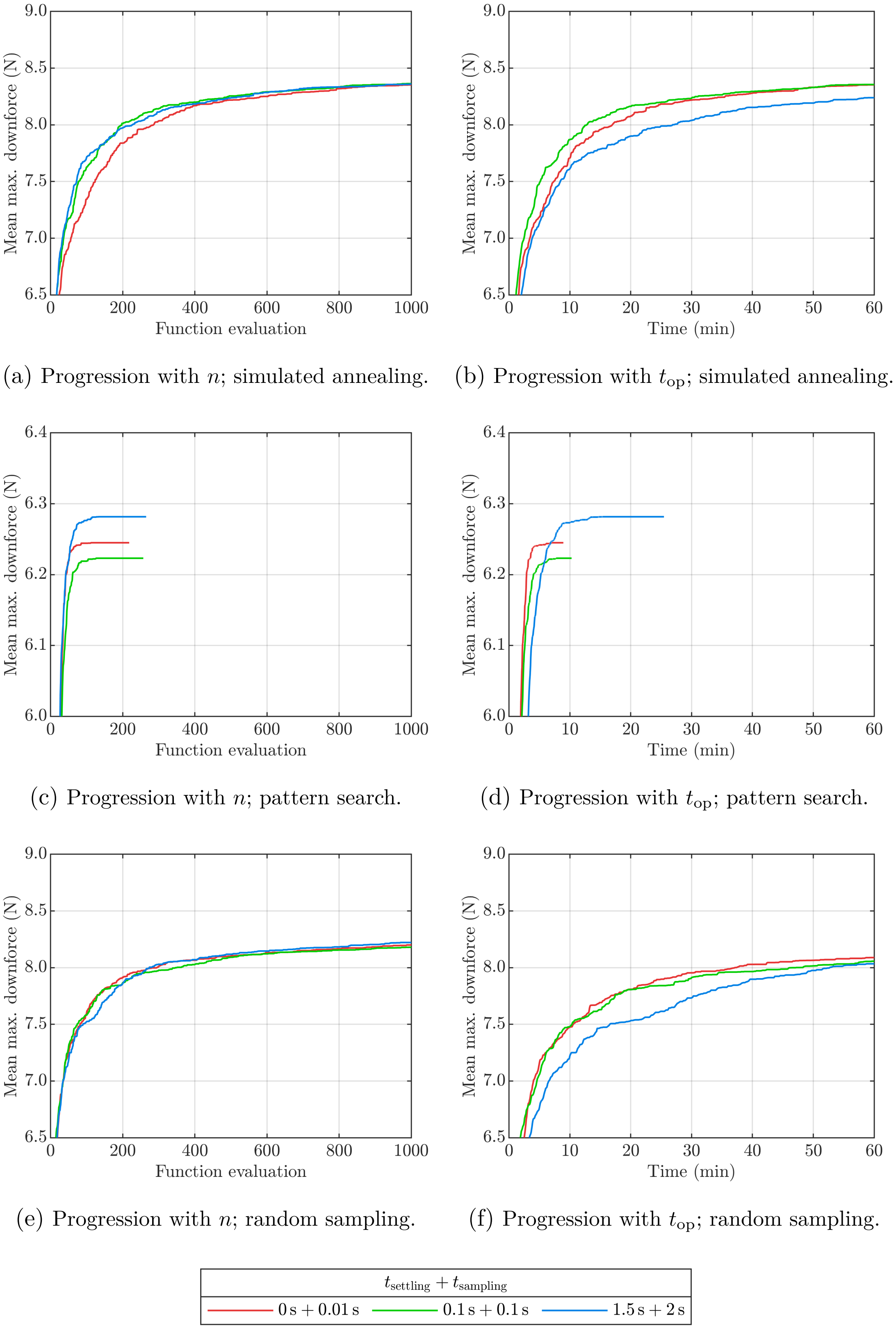
(f) Progression with t_{op}; PSO.

FIGURE 4.8: Mean maximum fitness progression of the population-based algorithms, plotted against function evaluation and optimisation time.

(Fig. 4.8e) appear to be unaffected by reduced settling and sampling times, as their mean downforce grows at the same rate with respect to function evaluation for all three combinations. This result confirms the resistance to noise of the population-based algorithms, and affirms their applicability to hardware-in-the-loop optimisation systems. The right-hand side plots (Fig. 4.8b, 4.8d and 4.8f respectively) illustrate how reducing t_{settling} and t_{sampling} improved the overall efficiency of the optimisation process, allowing the algorithms to reach higher downforce values in the same amount of time. Average downforce gains of the reduced combination relative to the baseline after 60 min were 1.5%, 0.7% and −0.1% for the integer- and binary-encoded GAs and PSO respectively. However, after 20 minutes, these gains were 3.5%, 2.7% and 1.2% respectively, due to less advanced convergence. These improvements are in line with random sampling, which experienced gains of 3.6% and 0.3% after 20 min and 60 min respectively. From a different perspective, the reduced settling and sampling times allowed the algorithms to attain downforce of 8 N 33.9%, 25.1% and 25.1% quicker respectively. The differences between the reduced and minimal combinations were significantly smaller, due to the negligible change in function evaluation length.

Figure 4.9 illustrates the convergence of simulated annealing and pattern search. The progression of mean maximum downforce for SA was not visibly affected by the reduced settling and sampling times, but the minimal combination significantly deteriorated performance in the early phase of optimisation, as seen in Fig. 4.9a. This is consistent with the findings presented in Section 4.2, where the SA algorithm was shown to be sensitive to noise. However, in spite of this difference, the minimal combination of t_{settling} and t_{sampling} still provides a significant overall efficiency improvement over the baseline case, as seen in Fig. 4.9b. Furthermore, no consistent difference between the reduced and minimal combinations was observed beyond 400 function evaluations.

Pattern search, displayed in Fig. 4.9c and 4.9d, was not significantly affected by increased noise, similar to the GAs and PSO. However, it rarely attained the global optimum even with the highest settling and sampling times, with the average maximum downforce equal to only about 75% of that attained by the other algorithms, and significantly lower than that obtained through random sampling in the same amount of time. This suggests that the noise and local maxima present in the system are sufficient to prevent reliable convergence even in the most conservative settling and sampling time configuration, confirming that pattern search is not a viable optimisation method for a hardware-in-the-loop system.



(a) Progression with n; simulated annealing.

(b) Progression with t_{op}; simulated annealing.

(c) Progression with n; pattern search.

(d) Progression with t_{op}; pattern search.

(e) Progression with n; random sampling.

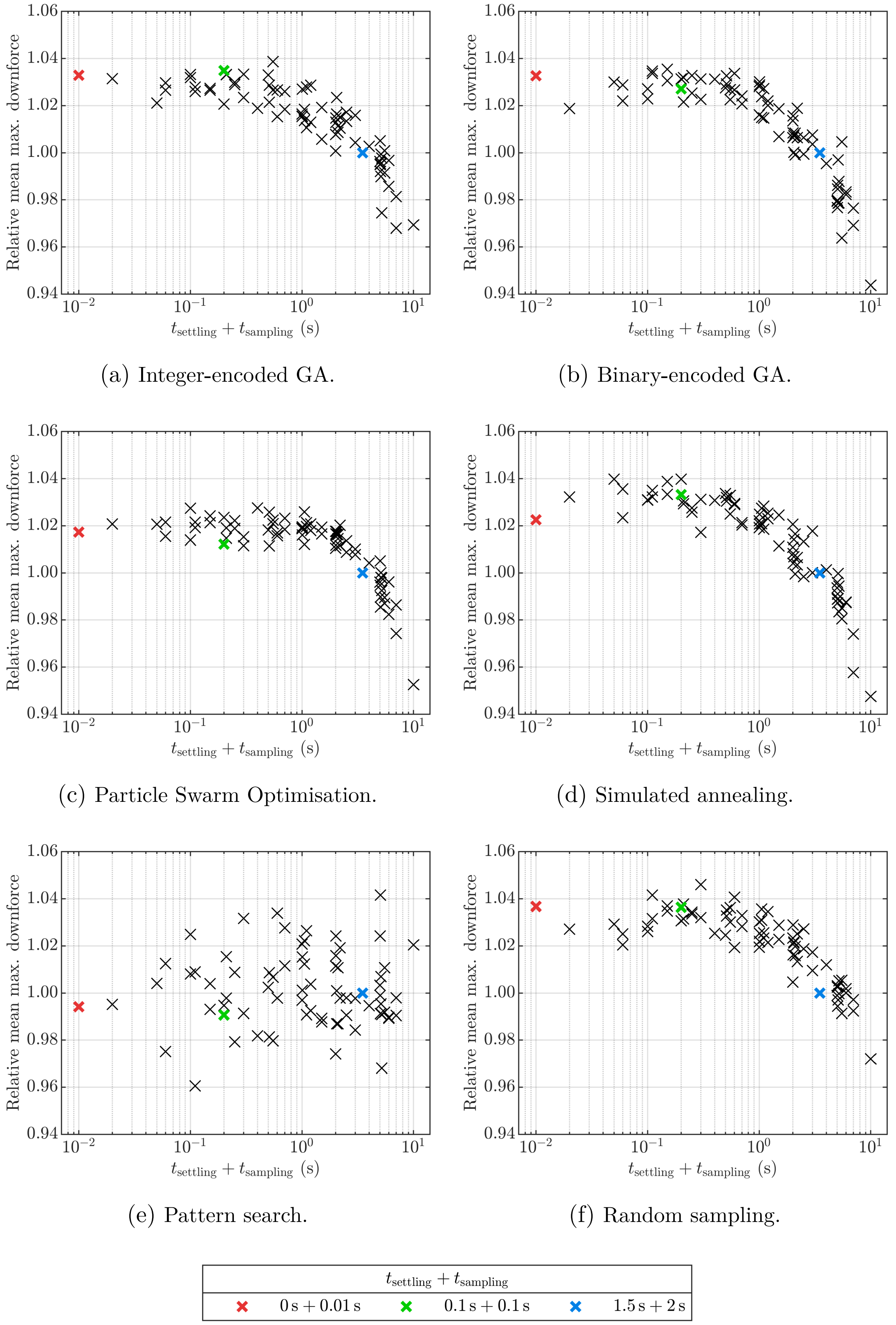
(f) Progression with t_{op}; random sampling.

FIGURE 4.9: Mean maximum fitness progression of the non-population-based algorithms, plotted against function evaluation and optimisation time. Y-axis scale for pattern search is adjusted for clarity.

In order to better visualise the potential benefit of reducing settling and sampling times on all tested combinations, mean maximum downforce relative to the baseline ($t_{\text{settling}} = 1.5\,\text{s}$ and $t_{\text{sampling}} = 2\,\text{s}$) after 20 min was averaged from all 100 trials for each case, and plotted as a function of $t_{\text{settling}} + t_{\text{sampling}}$ in Fig. 4.10. The 20 min point was chosen, rather than the time limit of 60 min, as none of the algorithms are converged at this stage, highlighting gains in the early phase of optimisation. Every cross on the plots corresponds to a unique combination of settling and sampling times, and the coloured crosses correspond to the baseline, reduced and minimal combinations discussed above. Only data for configuration 1 of pattern search and configuration 2 of the remaining algorithms is shown, similar to Fig. 4.8 and 4.9, in addition to random sampling, which again serves as a reference.

As seen in Fig. 4.10a to 4.10d, the genetic, PSO and SA algorithms exhibit similar trends in response to changing settling and sampling times, with relative downforce growing from 0.95–0.97 at $t_{\text{settling}} + t_{\text{sampling}} > 6\,\text{s}$, to 1.02–1.03 at $t_{\text{settling}} + t_{\text{sampling}} \leqslant 1\,\text{s}$. At lower values the trend is approximately flat, with no significant improvement between 1 s and 0.01 s. The thickness of the bands of points in all figures corresponds to the random variations in the state of convergence at the given point, present despite the 100-trial averaging. Notably, simulated annealing was equally consistent as the population-based algorithms, and the deterioration of performance from $0.1 + 0.1\,\text{s}$ to $0 + 0.01\,\text{s}$ appears to be a random fluctuation. The slightly lower improvement experienced by PSO is due to the algorithm's faster convergence—PSO consistently converged on the global optimum within the time limit even with long settling and sampling times, as seen in Fig. 4.8f, therefore the effect of increasing the number of function evaluations is less significant than for algorithms with slower convergence.

Pattern search, shown in Fig. 4.10e, demonstrates a distinctive trend compared to the other algorithms. The vertical spread of the points is much greater, which is caused by the reduced convergence consistency. As pattern search, similar to simulated annealing, was initialised from random locations in the search space, in fortuitous cases it converged onto the global maximum rapidly, whereas in others it was hindered by noise or local maxima. This contrasts with the steadily-converging population-based algorithms, and with configuration 2 of simulated annealing, where the very short reannealing period yielded strong exploratory behaviour. The lack of a clear relationship between settling and sampling times and relative performance has two causes. Firstly, the algorithm consistently converged before the 20 min time limit, as



(a) Integer-encoded GA.

(b) Binary-encoded GA.

(c) Particle Swarm Optimisation.

(d) Simulated annealing.

(e) Pattern search.

(f) Random sampling.

FIGURE 4.10: Scatter plots of mean maximum fitness after 20 min relative to the baseline case, for all the tested algorithms, plotted as a function of $t_{\text{settling}} + t_{\text{sampling}}$. Each cross corresponds to a different combination of t_{settling} and t_{sampling}.

seen as Fig. 4.9d, so any differences in convergence speed could only be observed at an earlier stage of optimisation. Secondly, the average impact of reduced settling and sampling times on convergence is very small, as discussed above.

Finally, random sampling exhibits similar characteristics to the first four algorithms, as seen in Fig. 4.10f, but with slightly increased values along the entire band of points. This is likely caused by random chance, with the baseline point (highlighted with a blue cross) lying at the bottom edge of the band, and not in the middle, as in the remaining plots, therefore shifting the remainder of the band upwards.

4.3.4 Experimental Validation

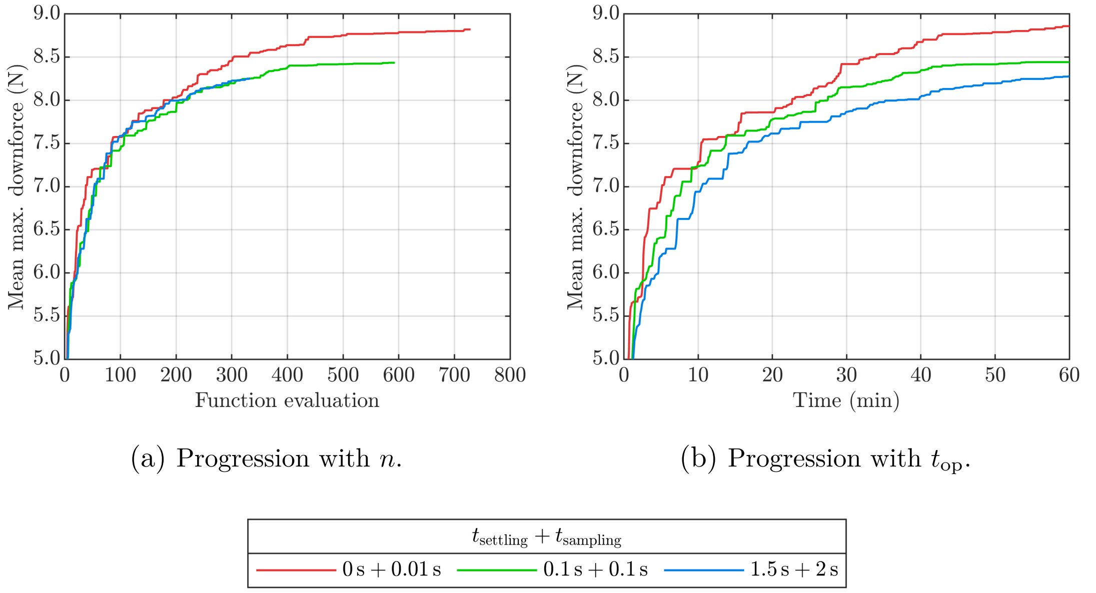
In order to validate the results presented in the preceding section, experimental tests were performed using the small-scale diffuser model, similar to the tests presented in Section 4.2. Due to the small variations in convergence performance with changing settling and sampling times, 10 repeated trials of every configuration were performed to obtain meaningful average performance. The same combinations of settling and sampling times were used as in Fig. 4.8 and 4.9, i.e. baseline, reduced and minimal. Configuration 2 of the integer-encoded GA was chosen for the validation tests due to its high performance, broad applicability to hardware-in-the-loop optimisation, and public availability. All optimisation runs were terminated after 60 min, following the previously established convention.

Similar to the simulation results (see Section 4.3.3), mean maximum downforce progression was averaged for each combination of t_{settling} and t_{sampling}, and plotted against function evaluation and optimisation time in Fig. 4.11.

Figure 4.11a shows that reducing the settling and sampling times permitted the GA to carry out significantly more function evaluations within the 60 min time limit:

- Between 333 and 404 with $t_{\text{settling}} = 1.5\,\text{s}$ and $t_{\text{sampling}} = 2\,\text{s}$ (baseline);
- Between 593 and 795 with $t_{\text{settling}} = 0.1\,\text{s}$ and $t_{\text{sampling}} = 0.1\,\text{s}$ (reduced);
- And between 729 and 846 with $t_{\text{settling}} = 0\,\text{s}$ and $t_{\text{sampling}} = 0.01\,\text{s}$ (minimal).

Although the baseline and reduced configurations exhibited similar convergence pace with respect to function evaluation, the reduced times enabled the algorithm to converge on a higher downforce value, as seen in Fig. 4.11b. However, the minimal configuration ($t_{\text{settling}} = 0\,\text{s}$ and $t_{\text{sampling}} = 0.01\,\text{s}$) significantly outperformed the other two, not only



(a) Progression with n.

(b) Progression with t_{op}.

FIGURE 4.11: Progression of mean maximum downforce against function evaluation and optimisation time, averaged from 10 experimental trials using configuration 2 of the integer-encoded genetic algorithm.

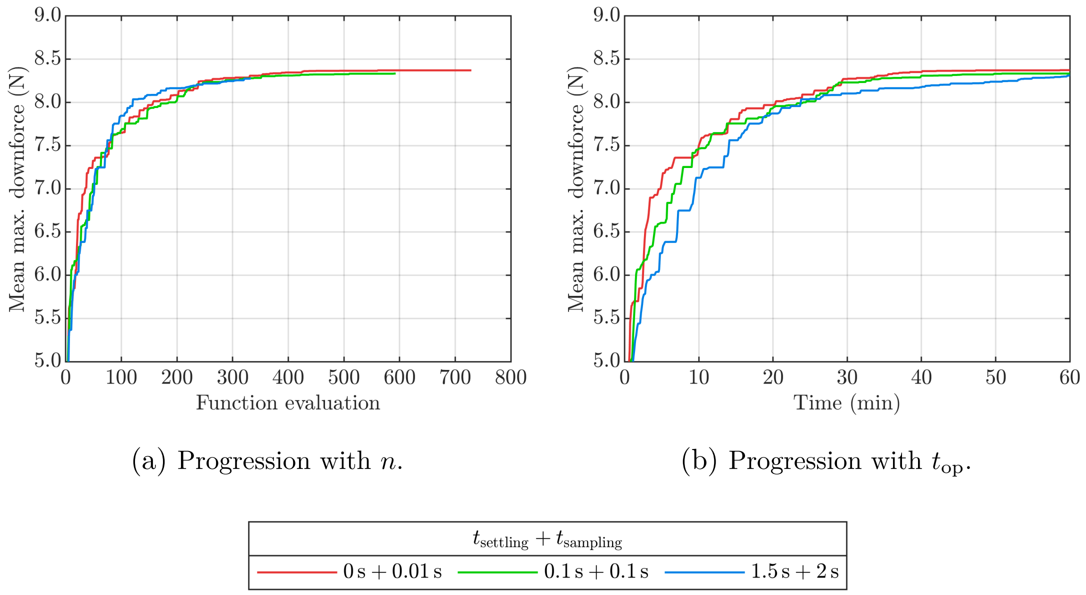
in the final downforce value, but even in early stages of optimisation. This is likely a direct consequence of the short sampling time, which increases the chance that a large part of a measurement sample is located on a positive peak in the continuous stream of fluctuating readings. As a result, some samples register higher downforce than the average value at that point, driving the maximum downforce progression upwards.

In order to remove the impact of short sampling times on the downforce readings, the measured values in all validation runs were replaced by values from the corresponding position in the interpolated downforce matrix previously used for the simulations presented in Section 4.3.3. This standardised way of displaying results enables a more fair comparison between the settling and sampling time combinations, and is analogous to displaying downforce values without noise in Section 4.3.3. This method introduces an error due to long-term measurement drift and the lack of hysteresis, which may result in a slight shift of the global maximum in the search space. However, as the same data matrix is used consistently for all validation runs, the impact of the error on the comparison between settling and sampling time combinations is minimal.

Figure 4.12 shows a comparison of the progression of mean maximum downforce, against function evaluation and optimisation time, using the standardised downforce values. It can be immediately seen from Fig. 4.12a that all three configurations of settling and sampling times exhibit very similar convergence speed. The baseline configuration rose slightly faster between the 100th and 200th function evaluations, but no significant differences are present after the 220th evaluation. This confirms that the convergence speed of the genetic algorithm is not hindered by the increased noise due to reduced settling and sampling times, as previously suggested by the simulation results.

The most realistic comparison of convergence efficiency is shown in Fig. 4.12b, which displays a comparison of progression of standardised downforce with respect to optimisation time. The plot suggests that decreasing the settling and sampling times from baseline to reduced, and from reduced to minimal, results in a gradual improvement in optimisation efficiency, with the minimal configuration achieving the highest average value of downforce throughout the entire run, higher than the baseline by up to 2% in the second half of the runs. However, in order to better understand the improvement in optimisation efficiency, we may swap the axes, and instead of comparing what downforce levels can be achieved in the same amount of time, compare how long it takes to achieve the same levels of downforce.

Figure 4.13a shows the plot from Fig. 4.12b with swapped axes, representing the time needed to reach particular mean maximum downforce values. It can be seen that the shorter the settling and sampling times, the less time is needed to reach the majority of downforce values. In order to quantify the efficiency gain, Fig. 4.13b shows the time needed to reach particular downforce values relative to the baseline settling



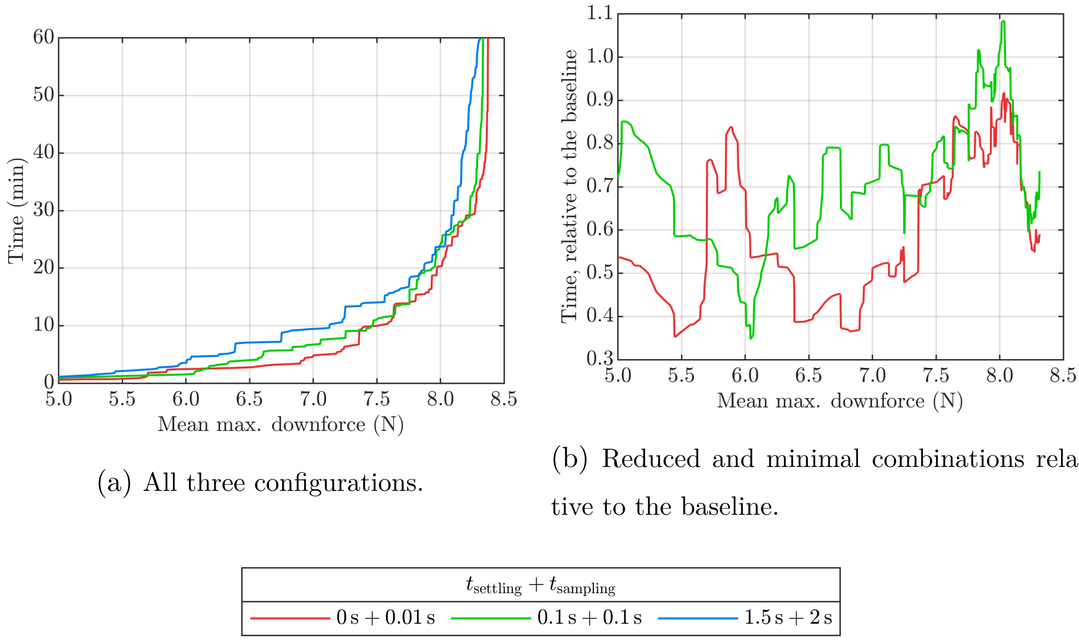
(a) Progression with n.

(b) Progression with t_{op}.

FIGURE 4.12: Progression of mean maximum downforce against function evaluation and optimisation time, averaged from 10 experimental trials using configuration 2 of the integer-encoded genetic algorithm, plotted using standardised downforce values.

and sampling times, for both the reduced and minimal combinations. The two curves confirm that shortening function evaluations enables the GA to attain the same performance quicker. Both combinations can reduce the required optimisation time by up to 65%, or by around 30% to 40% on average, but crucially, reducing the times down to $t_{\text{settling}} = 0\,\text{s}$ and $t_{\text{sampling}} = 0.01\,\text{s}$ provided the greatest benefits. The minimal configuration was also able to attain the highest mean maximum downforce, confirming both the positive impact of reducing the function evaluation length, and the GA's resistance to noise.

However, the plots presented so far only illustrate average performance, but give no indication of the consistency of convergence when settling and sampling times are decreased. This crucial aspect of optimisation performance was inspected through plots of the standard deviation (SD) of downforce at a given function evaluation and optimisation time, as illustrated in Fig. 4.14. Figure 4.14a shows that the standard deviation of maximum downforce decreases rapidly as the algorithm converges, but in the first 200 evaluations it is lowest for the baseline combination, suggesting that the increased noise indeed has some negative effect on consistency. However, the reduced ($t_{\text{settling}} = 0.1\,\text{s}$, $t_{\text{sampling}} = 0.1\,\text{s}$) and minimal ($t_{\text{settling}} = 0\,\text{s}$, $t_{\text{sampling}} = 0.01\,\text{s}$) combinations allow the algorithm to converge further, and SD of downforce decreases down to 0.3% and 0.8% of the mean respectively, compared to 1.9% for the baseline.

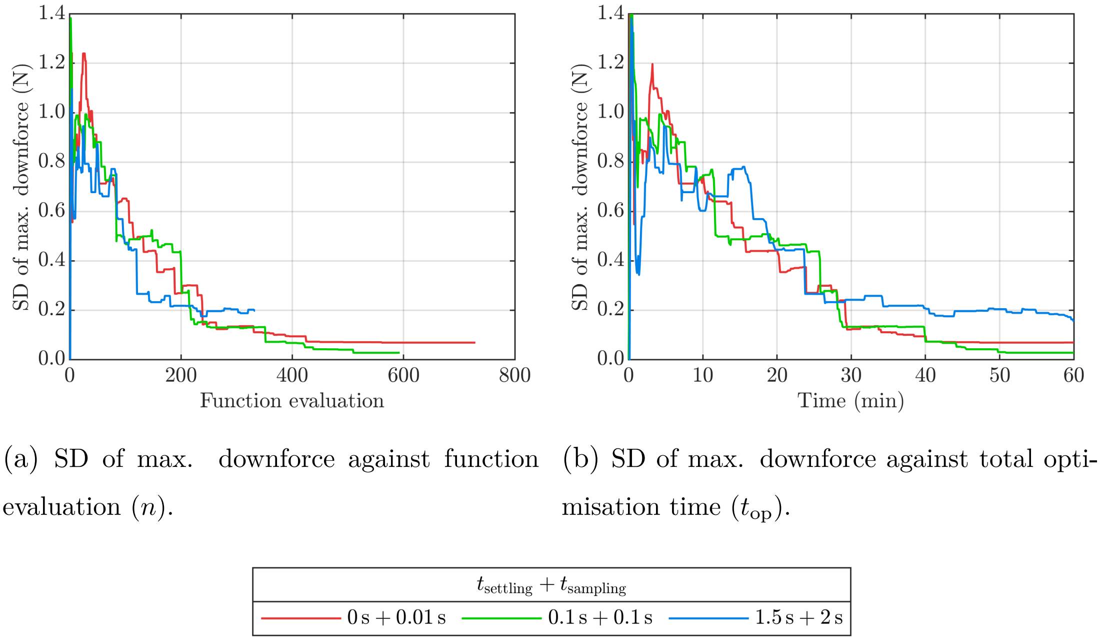


(a) All three configurations.

(b) Reduced and minimal combinations relative to the baseline.

FIGURE 4.13: Time needed to reach particular mean maximum downforce values, for all settling and sampling time configurations, and relative to the baseline configuration.

Figure 4.14b displays a plot of downforce standard deviation against optimisation time, and it can be seen that although similar values are achieved by the three combinations in the first 30 min, the reduced and minimal settling and sampling times enable the GA to attain much more consistent downforce levels in the second half of the runs. Therefore, despite the increased experimental measurement noise, shorter settling and sampling times enable the optimisation algorithm to converge faster, to higher absolute values of fitness, and more consistently, in the same amount of time.



(a) SD of max. downforce against function evaluation (n).

(b) SD of max. downforce against total optimisation time (t_{op}).

Figure 4.14: Standard deviation of maximum downforce with respect to function evaluation and optimisation time.

4.4 Summary

4.4.1 Comparative Optimisation Tests

The current chapter presented the results of two studies related to hardware-in-the-loop aerodynamic optimisation. The first involved a series of wind tunnel-based optimisation runs of the small-scale diffuser model, using a total of 14 configurations of 4 distinctive optimisation methods. Its aim was to establish how different methods and their implementations can cope with an inherently noisy experimental problem.

The results showed that population-based algorithms, in this case genetic algorithms and particle swarm optimisation, converged on the global optimum efficiently and reliably, despite the noise and hysteresis present in the system. This was due to a favourable blend of exploratory and exploitative characteristics, which allowed them to simultaneously explore the search space enough to avoid getting trapped in a local maximum, and to shift the majority of the population to the high-performance region and exploit it with high precision. However, tests using different configurations of the algorithms revealed that tuning their parameters may unsettle this balance sufficiently to prevent reliable convergence, and therefore care should be taken when setting up an algorithm for a particular problem. Notably, maintaining sufficient genetic diversity is paramount in the case of genetic algorithms, as excessive selection pressure may lead to premature convergence. Furthermore, binary encoding exhibited improved performance relative to integer encoding, although theory suggests that this effect would be smaller in problems with more degrees of freedom. In the case of particle swarm optimisation, a balance between cognitive and social search should be sought, and the magnitude of inertia weight must be tuned to the size of the search space.

Simulated annealing and pattern search were both far less effective than the population-based methods, unable to overcome the noise and local extrema in the system, regardless of their configurations. The SA algorithm was found to perform best if high temperature and a short reannealing period were used. However, this was not due to improved convergence characteristics, but due to a more uniform distribution of measurement points, which made the algorithm less likely to overlook the vicinity of the global maximum. Other configurations had a tendency to overexploit low-performance regions, and occasionally found the optimum by chance, rather than systematically. Pattern search, on the other hand, was sensitive to starting position, and was unable to avoid getting trapped in local maxima, preventing it from being considered as a viable method for hardware-in-the-loop aerodynamic optimisation.

Importantly, a more advanced hybrid algorithm could alleviate some of the flaws of the algorithms tested in this study. For example, the pattern search could be complemented by frequent but brief broadening of the search, or several instances of pattern search could be performed starting from promising locations identified in an initial low-fidelity global search, e.g. through a few generations of a genetic algorithm. However, the aim of this research was not to determine or develop an optimal optimisation method, but rather to identify what algorithm characteristics were or were not effective for a particular type of problem. This was achieved using a range of simple yet extensively-researched algorithms, as their behaviour could be readily linked to their configurations, and therefore hybrid optimisation methods were not explored.

4.4.2 Sampling Time Study

The second part of this chapter detailed the investigation on how reducing the settling and sampling times in wind tunnel-based optimisation affects convergence speed and overall optimisation efficiency. Simulated optimisation runs were carried out on interpolated downforce data of the small-scale diffuser model, and were supported with experimental noise measurements acquired specifically for this purpose.

A range of settling and sampling times were used, both longer and shorter than the baseline used in the comparative tests, and 100 trials were carried out using all 14 previously tested algorithm configurations, as well as with random sampling. The results showed that convergence of the genetic algorithms and particle swarm optimisation was not significantly affected by the changing parameters, but reducing the time required for one function evaluation enabled the algorithms to test more configurations in the same amount of time, and therefore converge on the global optimum quicker. Eliminating the settling time, or the delay between actuation completion and the start of sampling, and reducing the sampling time from 2 s to 0.01 s, provided an improvement in maximum fitness of up to 4%, depending on the configuration. Simulated annealing was the only algorithm negatively affected by the shortest sampling time, but similar improvements as for the population-based algorithms were observed with sampling times in the order of 0.1 s. Experimental validation of the simulations confirmed these findings, as no negative impact on convergence of the integer-encoded genetic algorithm due to reduced settling and sampling times was observed. Shortening the function evaluation allowed the algorithm to more than double the number of evaluations within the time limit, enabling more effective, but also more consistent convergence.

In conclusion, optimisation of experimental problems with inherent measurement noise may benefit from minimised settling and sampling times, in contrast to the traditionally conservative values used in aerodynamic testing, where data smoothness is highly valued. Depending on the response rate of the system, a settling period might be redundant, as population-based methods are insensitive to hysteresis and occasional measurement disturbances. The optimal sampling time will depend on the relative magnitude of measurement noise, but sample lengths in the order of 0.1 s, or even 0.01 s, might be sufficient for effective convergence.

Chapter 5

Population Sorting for Operational Time Reduction

This chapter presents the results of the population sorting study, where combinatorial optimisation was utilised to reduce operational time in systems with multiple degrees of freedom that need to transform through a series of configurations in any order, e.g. a robotic arm or an actuated geometry during hardware-in-the-loop aerodynamic optimisation. The first aim of this study was to improve the total efficiency of the HIL aerodynamic optimisation process by reducing operational overheads, specifically geometry actuation time, and therefore to accelerate convergence with respect to wall-clock time, without any detriment to the quality of the outcome. Actuation constituted between 35% and 60% of total operational time in optimisation trials using the small-scale model, and this proportion would be even higher in systems with slower actuation, so any reduction in actuation time would translate to a significant overall efficiency improvement. The second objective was to generalise the results to any multidimensional sorting problem to which the Chebyshev cost metric applies.

The first section presents a formulation of the problem, and the methods used to simulate sorting performance under controlled conditions. Then, the impact of a range of factors on sorting effectiveness is presented, followed by the development of a methodology for minimising total operational time by choosing a particular sorting method and sorting time limit. Finally, the methodology is validated experimentally, using the large-scale wind tunnel system described in Section 3.2.

5.1 Methodology

5.1.1 Problem Definition

The specification of the problem is as follows: an L-sized permutation of integer design points $[p_1, p_2, \ldots, p_L]$ is generated, randomly or otherwise, using a set of constraints. Each design point p_i exists in a finite design space with N dimensions, each dimension bound by the constraints $[1; B_n]$, where B_n is the effective length of the n-th dimension, defined as follows:

$$B_n = \frac{B_0 \cdot R_n}{R_1 \cdot R_2 \cdot \ldots \cdot R_N}\,, \tag{5.1}$$

where B_0 is the baseline dimension length, and R are randomly generated numbers from 1 to the maximum aspect ratio AR. Using this formula enables the investigation of the effect of varying the design space aspect ratio, while maintaining a constant design space size, defined as the total number of points within the design space and equal to B_0^N. This procedure results in a design space that is orthogonal, as all constraints are constant-variable hyperplanes, but not necessarily equilateral, as dimensions might have different lengths if $AR > 1$. Design space skew, or deviation from an orthogonal shape, is achieved by applying a non-orthogonal constraint hyperplane during initialisation, as explained further in Section 5.5.3.

After a permutation is initialised using the parameters outlined above, it is given to a sorting algorithm, which is tasked with rearranging (or sorting) it to minimise the overall cost C, which is defined as the total Chebyshev distance between consecutive points, as in Eq. 5.2:

$$C = \sum_{i=1}^{L-1} \max\left(\left|p_{i+1}^1 - p_i^1\right|, \left|p_{i+1}^2 - p_i^2\right|, \ldots, \left|p_{i+1}^N - p_i^N\right|\right), \tag{5.2}$$

where p_i^k is the k-th dimension coordinate of point p_i.

5.1.2 Testing Procedure

To obtain an average measure of performance, each sorting algorithm was run 100 times for every test case. In each trial, the algorithms received permutations of design points randomly generated using appropriate constraints for every dimension. The total Chebyshev distance C of those permutations was calculated prior to sorting. The algorithms were then allowed a fixed amount of time to optimise the order of design points to minimise C, and were terminated after this time elapsed. The baseline value for the time limit was set to an arbitrary value of $t_{\max} = 1\,\mathrm{s}$, but the effect of changing it was investigated as well.

Importantly, all sorting algorithms used pre-calculated cost matrices, i.e. $L \times L$ matrices of Chebyshev distance for all pairs of points in the permutation, as this required fewer calculations than obtaining the values on-demand.

When all 100 iterations using a given algorithm were completed, the average total cost before sorting, denoted as C, was compared to the average total cost after sorting $\mathfrak{C}$, to quantify the percentage improvement I according to the following formula:

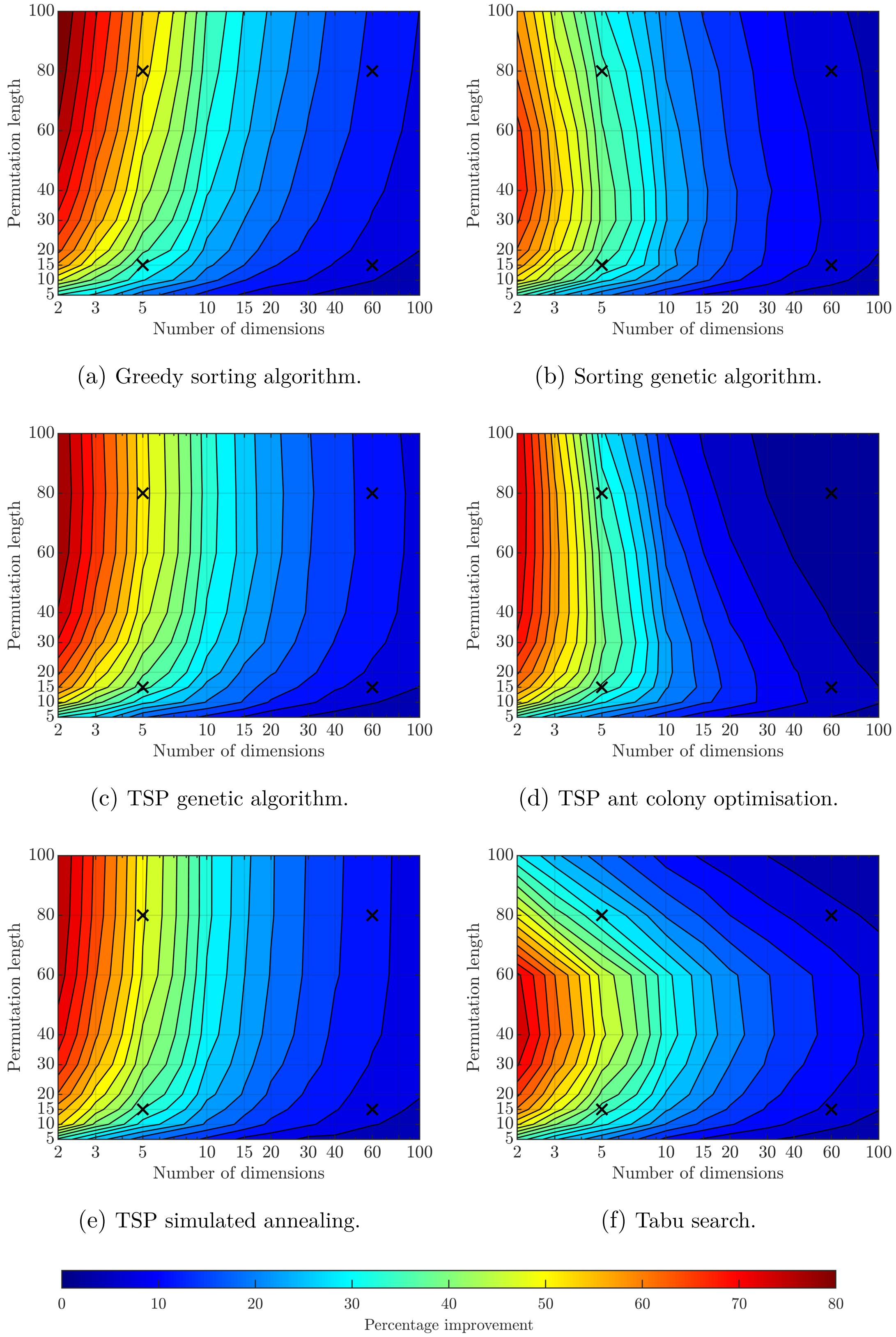
$$I = 100 \cdot \frac{C - \mathfrak{C}}{C}\,. \tag{5.3}$$

It is important to note that most types of optimisation carried out using population-based algorithms begin with populations that are generated without any physical input, either randomly or using a more advanced method. Therefore, it is possible to perform realistic simulations of sorting of such populations using exclusively numerical methods, without implementing hardware-in-the-loop. However, gradual convergence of the population of solutions occurs over the course of an optimisation run, as the algorithms transition from exploratory to exploitative behaviour and tend to converge on the global maximum. This reduces the average spread of points in the design space, which might affect the potential effectiveness of sorting such a population. It is not possible to accurately reproduce this convergence during simulated tests, therefore the results presented in Chapter 5 are most applicable to initial, unconverged populations.

5.2 Effect of Permutation Length and Dimensionality

The series of plots in Fig. 5.1 show the average percentage improvement I achieved by each of the 6 sorting algorithms introduced in Section 2.2.2, using randomly generated permutations of the given length and number of dimensions, for orthogonal and equilateral ($AR = 1$) design spaces with a baseline dimension length of $B_0 = 100$. The x-axes of the plots are logarithmic, in order to better display the trends occurring with low numbers of dimensions.

The most notable trend that can be observed in all the plots in Fig. 5.1 is that improvement I decreases significantly with increasing number of dimensions N. Crucially, N does not affect the size of the sorting problem and hence the computational cost of



(a) Greedy sorting algorithm.

(b) Sorting genetic algorithm.

(c) TSP genetic algorithm.

(d) TSP ant colony optimisation.

(e) TSP simulated annealing.

(f) Tabu search.

FIGURE 5.1: Contour plots of percentage improvement in cost I, for equilateral, orthogonal design spaces, with $t_{\max} = 1\,\mathrm{s}$.

solving it, but only affects the cost function. Therefore, the reduction in sorting effectiveness with increasing number of dimensions is not due to degrading effectiveness of the algorithms, but due to the decreasing maximum possible improvement I_{max} that can be achieved through sorting.

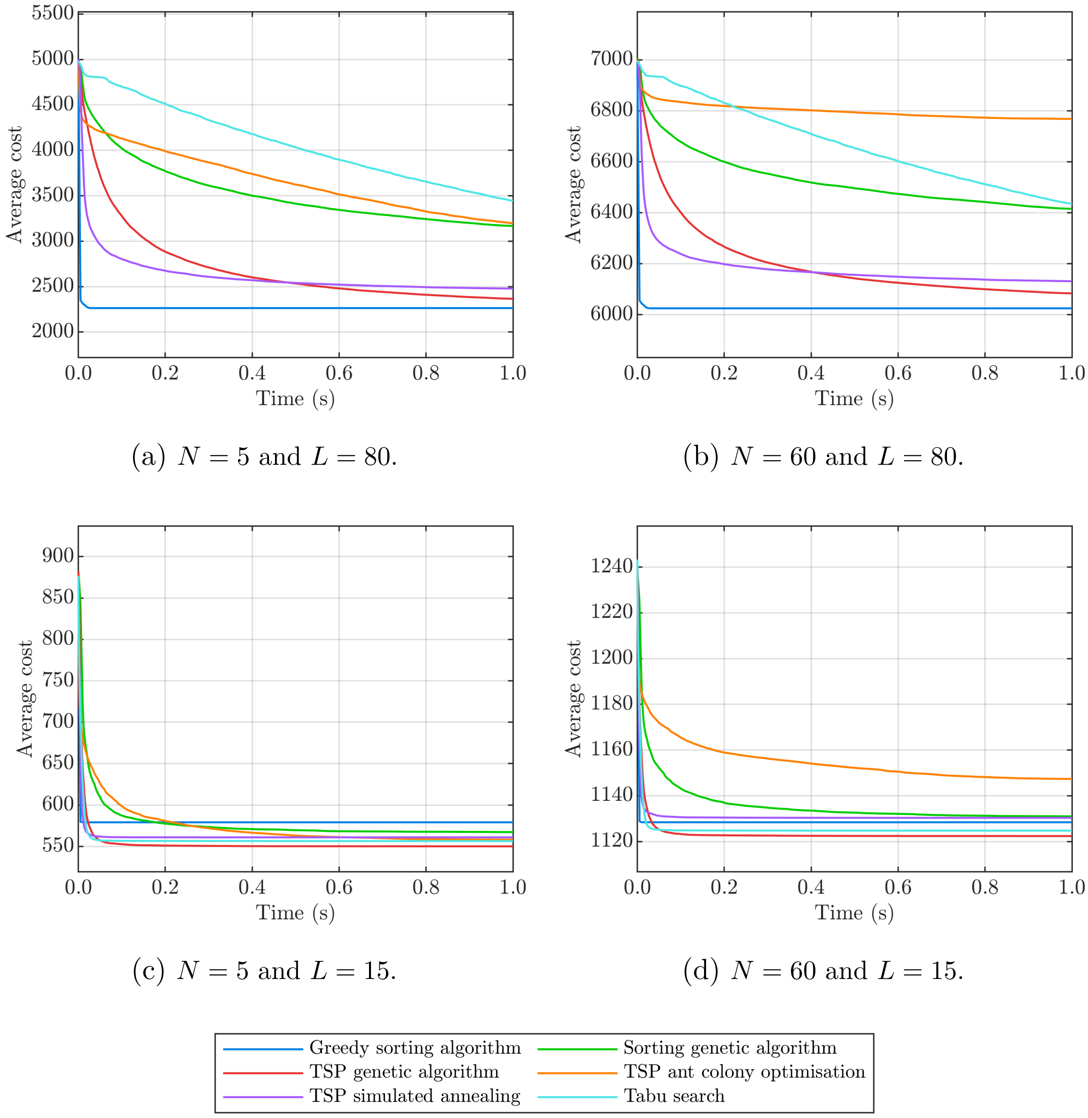
The reason for this can be understood by considering a pair of design points randomly located in a design space with N dimensions, and is linked to the way cost is calculated in these tests. As shown in Eq. 5.2, the Chebyshev distance between a pair of points is taken as the longest distance separating the two points in any single dimension. The larger the number of dimensions, the higher the probability that the two points are separated by a long distance in at least one dimension. As this is true for any given pair of points, there is less potential for improvement due to sorting of the entire population. This was confirmed by examining all possible permutations for permutation lengths up to 10. The results showed that I_{max} at $L = 10$ reduces from almost 70% at $N = 2$ to less than 15% at $N = 100$.

The factor that directly determines the size of the sorting problem is permutation length L. As the sorting algorithms were given the same amount of time regardless of the value of L in order to compare their sensitivity to problem size, it is expected that their ability to exploit the sorting potential would decrease with growing permutation length. However, Fig. 5.1 shows that this is the case for only 3 of the 6 algorithms, and to a varying extent. This indicates that the sensitivity of the sorting algorithms to problem size differs. Indeed, the algorithm that appears to be most sensitive to L is tabu search (seen in Fig. 5.1f), whose number of operations per iteration scales approximately with L^2, as it carries out three types of operations, i.e. swap, reversion and insertion, for all possible pairs of points within the permutation (see Section 2.2.2.6).

Figure 5.1a shows that, at constant N, the improvement I increases with increasing L for the greedy algorithm, although at a decreasing rate. The TSP GA and TSP SA algorithms (seen in Fig. 5.1c and 5.1e) display similar trends, but improvement I stalls at $L = 60$ and above. However, these results do not indicate that the algorithms are insensitive to L, as improvement I is not directly proportional to the maximum possible improvement I_{max}. Indeed, testing on all possible permutations showed that I_{max} increases with L in the range $L \in [3, 10]$, changing between 25% and 70% at $N = 2$, and between 20% and 50% at $N = 5$. It is not feasible to determine I_{max} for $L \geqslant 15$, as the total number of permutations is equal to $L!$, but it is hypothesised that it would keep increasing at a decreasing rate.

In order to better understand the sensitivity of the various algorithms to permutation length L and number of dimensions N, the progression of cost C (defined in Eq. 5.2) with time, averaged across all 100 trials, was plotted for all algorithms at 4 combinations of L and N, which are marked with black crosses on the contour plots in Fig. 5.1. The layout of the plots in Fig. 5.2 corresponds to the relative positions of the crosses in Fig. 5.1 for clarity, and the y-axes were scaled so that the first and final points of the greedy algorithm's plot are in the same position in all sub-figures.

It can be seen that, at $N = 5$ and $L = 15$ (Fig. 5.2c), all algorithms converged before the permitted time of 1 s elapsed. At the corresponding points in Fig. 5.1, the gradient of improvement with respect to permutation length, $\frac{\mathrm{d}I}{\mathrm{d}L}$, is positive for all 6 algorithms.



(a) $N = 5$ and $L = 80$.

(b) $N = 60$ and $L = 80$.

(c) $N = 5$ and $L = 15$.

(d) $N = 60$ and $L = 15$.

FIGURE 5.2: Plots of cost C against time, averaged across 100 trials, with $t_{\max} = 1$ s.

This gradient indicates the algorithms' sensitivity to problem size, i.e. whether further increasing permutation length has a detrimental effect on sorting effectiveness. Moving to $N = 60$ (Fig. 5.2d and the bottom-right points in Fig. 5.1), it can be seen that TSP ACO is the only algorithm that had not converged before the time limit was reached, and it is the only algorithm with a negative gradient $\frac{\mathrm{d}I}{\mathrm{d}L}$ at $N = 60$ and $L = 15$ in Fig. 5.1.

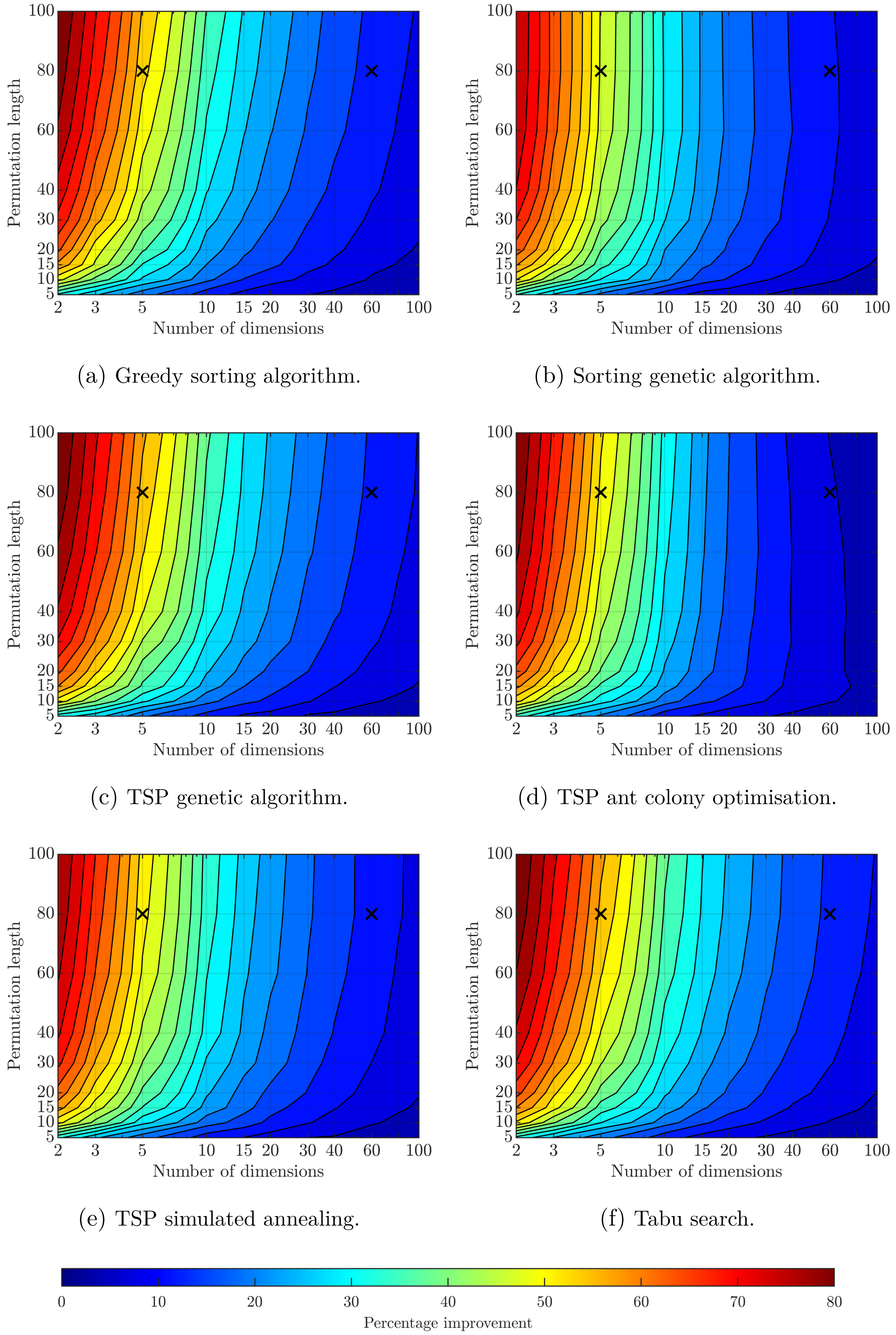
Increasing L to 80 has a much greater effect on all algorithms, as seen in Fig. 5.2a and 5.2b. The greedy sorting algorithm was the only one to converge before the time limit at both $N = 5$ and $N = 60$, and it has a positive gradient $\frac{\mathrm{d}I}{\mathrm{d}L}$ at both corresponding points in Fig. 5.1a. The TSP GA and TSP SA algorithms were both close to convergence when the time limit was reached, as seen in Fig. 5.2a and 5.2b, and Fig. 5.1c and 5.1e show that the gradient $\frac{\mathrm{d}I}{\mathrm{d}L}$ is close to 0.

The remaining algorithms, viz. sorting GA, TSP ACO and tabu search, did not reach convergence at the time limit in Fig. 5.2a and 5.2b, and all 3 have negative values of the gradient $\frac{\mathrm{d}I}{\mathrm{d}L}$ in Fig. 5.1b, 5.1d and 5.1f. Tabu search, which has the steepest gradient of cost versus time at $t = 1\,\mathrm{s}$ in Fig. 5.2a and 5.2b, also has the lowest values of the gradient $\frac{\mathrm{d}I}{\mathrm{d}L}$, as seen in Fig. 5.1f.

5.3 Effect of Increasing the Sorting Time Limit

The relationship between the degree of convergence of cost (Fig. 5.2) and the gradient of improvement I with respect to L (Fig. 5.1) indicates that the sensitivity of algorithms to permutation length L at the time limit of 1 s is not determined by their effectiveness, but by their efficiency. In this context, effectiveness is understood as an algorithm's ability to exploit the optimisation potential of a problem, or to maximise performance, whereas efficiency is the ability to do that quickly, or at a low computational cost. In order to obtain a clear picture of the effectiveness of the sorting algorithms, unhindered by time constraints, the above tests were repeated with a sorting time limit of 10 s. Contour plots of improvement I with respect to L and N at $t_{\max} = 10\,\mathrm{s}$ are shown in Fig. 5.3, and the corresponding plots of progression of cost are shown in Fig. 5.4.

The contour plots show that, as the sorting time limit was increased from 1 s to 10 s, all algorithms became significantly less sensitive to permutation length, with only the sorting GA and the TSP ACO algorithms having regions of negative gradient $\frac{\mathrm{d}I}{\mathrm{d}L}$ at high values of L and N. This is confirmed by Fig. 5.4b, where they are the only unconverged



(a) Greedy sorting algorithm.

(b) Sorting genetic algorithm.

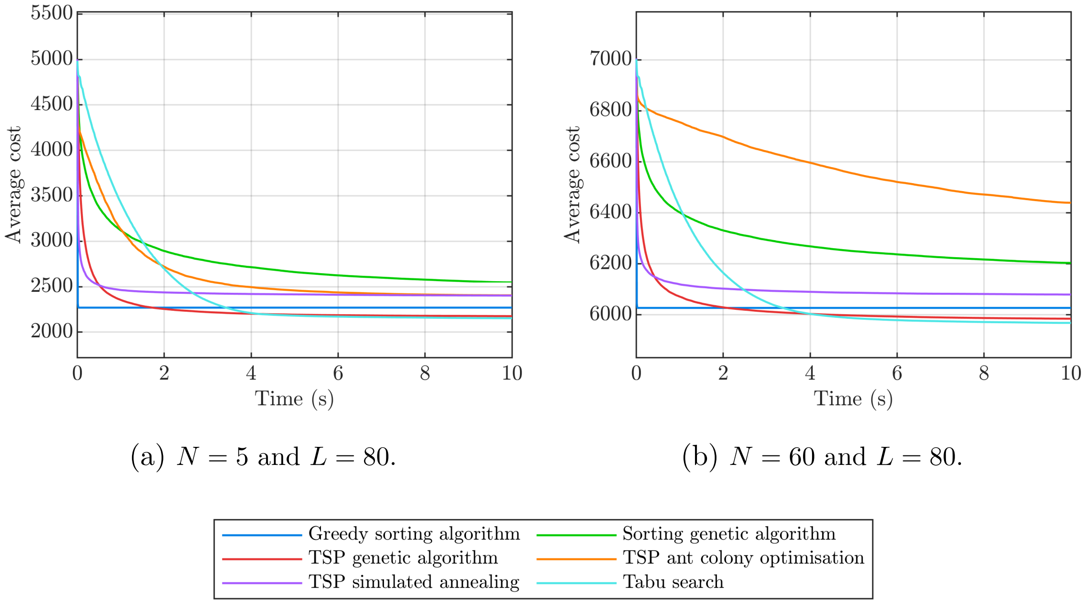
(c) TSP genetic algorithm.

(d) TSP ant colony optimisation.

(e) TSP simulated annealing.

(f) Tabu search.

Figure 5.3: Contour plots of percentage improvement in cost I, for equilateral, orthogonal design spaces, with $t_{\max} = 10\,\mathrm{s}$.



(a) $N = 5$ and $L = 80$.

(b) $N = 60$ and $L = 80$.

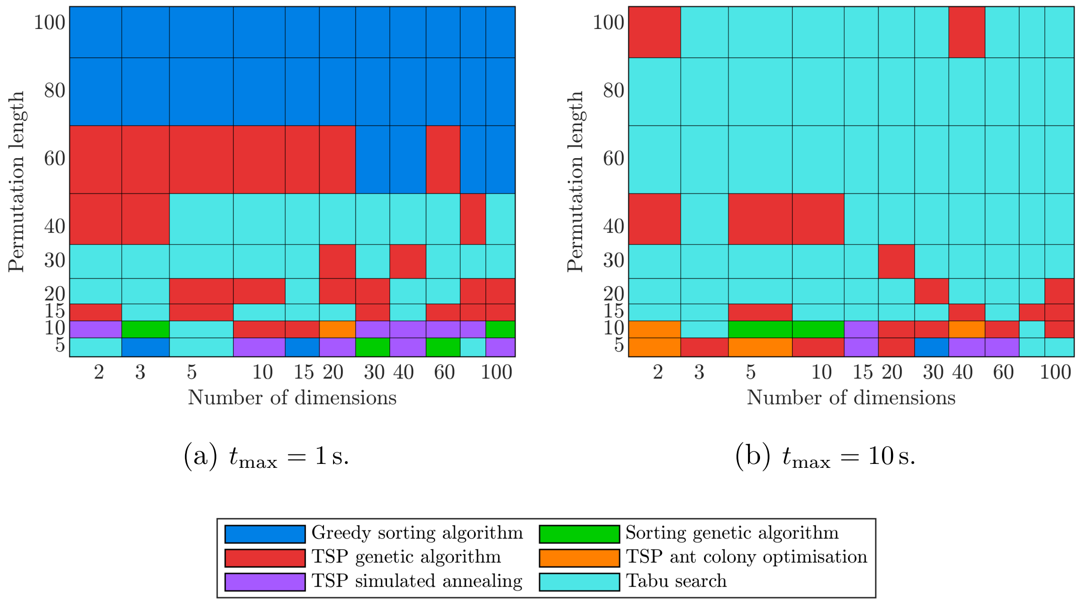
FIGURE 5.4: Plots of cost C against time, averaged across 100 trials, with $t_{\text{max}} = 10\,\text{s}$.

algorithms at $t = 10\,\text{s}$.

These two algorithms are the most computationally costly of the group, which means that they carry out the lowest numbers of look-ups to the pre-calculated cost matrices per second. The sorting GA and TSP ACO algorithms carried out on average approximately 4 and 0.14 million look-ups per second respectively, compared to 5, 30 and 8 million look-ups per second by the TSP GA, TSP SA and tabu search algorithms respectively. Whereas the computational cost of the algorithms partly justifies their deficit in efficiency, it is also likely that the operations modifying permutations during sorting have a significant impact on sorting effectiveness. It is also unclear why the ant colony optimisation algorithm is more sensitive to the number of dimensions N than all the other tested algorithms.

Figure 5.5 shows the best-performing algorithm at every combination of L and N, with time limits of $1\,\text{s}$ and $10\,\text{s}$, using the same colours as in Fig. 5.2 and 5.4.

At $t_{\text{max}} = 1\,\text{s}$ (Fig. 5.5a), greedy sorting is the most effective algorithm at $L \geqslant 80$, and the TSP GA has similar effectiveness to tabu search at $15 \geqslant L \geqslant 40$. At permutation lengths $L \leqslant 10$ the trends appear to be random, with all 6 algorithms being the most effective for at least one value of N. This is because, at low permutation lengths, the number of all possible permutations is small enough that all the algorithms are able to examine a substantial proportion of them. As a result, $I \approx I_{\text{max}}$, with the irregular variations caused by noise due to the random initialisation.



(a) $t_{\max} = 1\,\mathrm{s}$.

(b) $t_{\max} = 10\,\mathrm{s}$.

FIGURE 5.5: Diagram showing the best sorting algorithm for each design case.

At $t_{\max} = 10\,\mathrm{s}$ (Fig. 5.5b), the tabu search algorithm is the most effective at $L \geqslant 15$ and for almost all values of N. However, the TSP GA eclipses the performance of tabu search in approximately 15% of the cases, which suggests that their effectiveness is similar. These results, supported by the progression plots from Fig. 5.4, indicate that, whereas the tabu search algorithm converges more slowly than the other algorithms in the initial phase of sorting, it tends to converge on lower values of cost C. At $L \leqslant 10$ all algorithms demonstrate similar effectiveness, resembling the trend from Fig. 5.5a.

5.4 Selection of Sorting Method and Sorting Time Limit

The results from Section 5.3 show that different sorting methods require different amounts of time to converge, or to exploit the sorting potential of a problem. In any situation where total permutation cost C corresponds to time, as does Chebyshev distance in physical systems, the time taken to optimise the order of design points will also contribute to the operational time. Therefore, it is desirable not only to select an adequate optimisation algorithm to sort the points, but also to select a time limit that will minimise the overall operational time. This section presents a methodology that allows selecting both simultaneously, using a low-cost, computational analysis.

The total operational time t_{op} for a single permutation of design points can be expressed as:

$$t_{\mathrm{op}} = t_{\mathrm{sorting}} + t_{\mathrm{actuation}} + t_{\mathrm{overhead}}\,, \tag{5.4}$$

where t_{sorting} is the sorting time (limited to t_{max}), $t_{\text{actuation}}$ is the total actuation time of the system for the given permutation of design points, and t_{overhead} includes all operational overheads, such as measurement collection or data processing. The overheads also include actuator acceleration and deceleration, which may depend on actuation distance. However, as no clear relation had been identified, it was assumed that t_{overhead} is independent of the order of design points, and therefore may be omitted in the selection of sorting method and sorting time limit, resulting in:

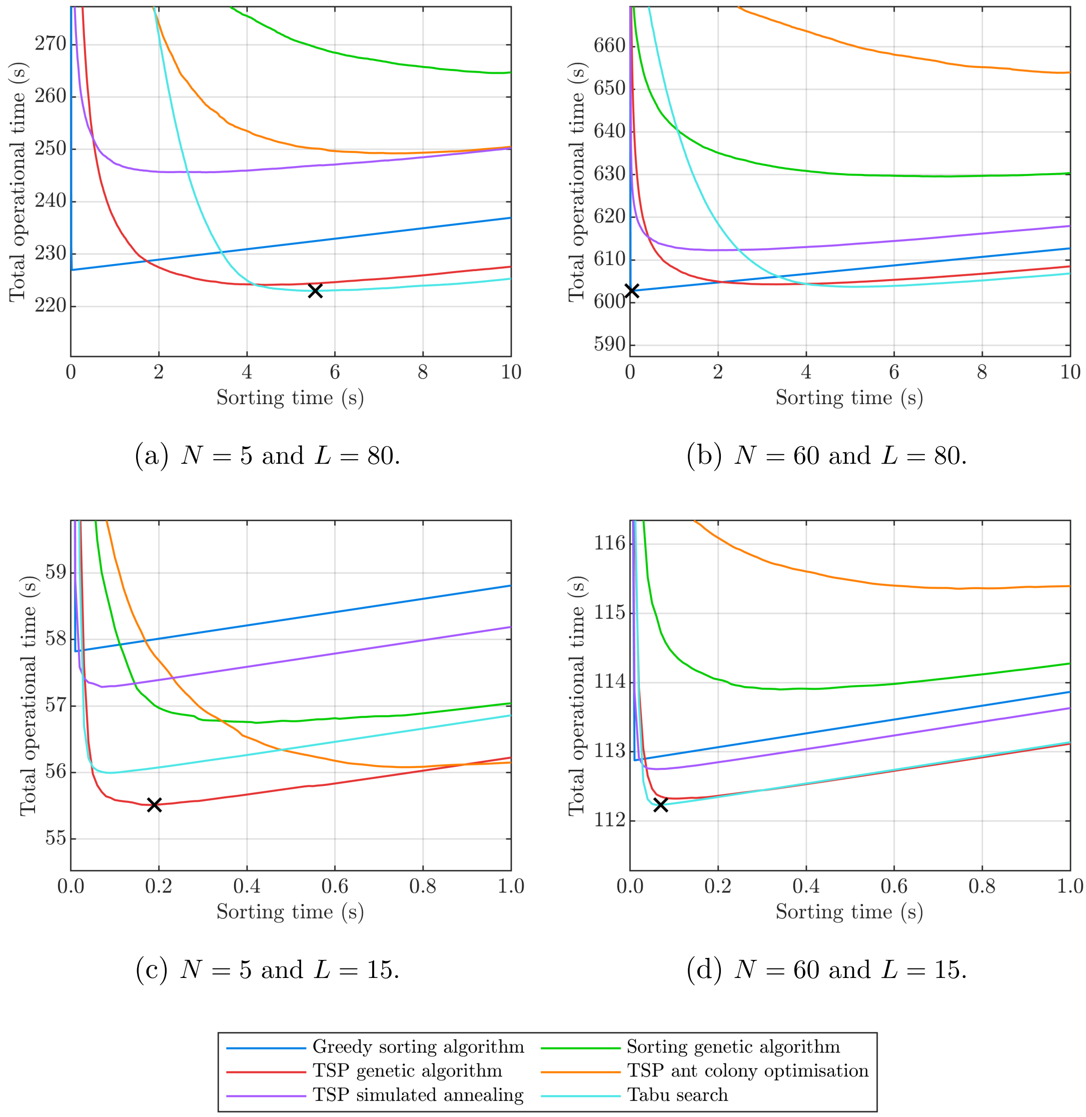
$$t_{\text{op}} \approx t_{\text{sorting}} + t_{\text{actuation}} \,. \tag{5.5}$$

The actuation time can be approximated as the quotient of the total Chebyshev distance C of the entire permutation, and nominal actuator speed v_{act}, so Eq. 5.5 becomes:

$$t_{\text{op}} \approx t_{\text{sorting}} + \frac{C}{v_{\text{act}}} \,, \tag{5.6}$$

where C, expressed in distance units, is a function of t_{sorting}, and v_{act} is actuator speed in distance units per second. This may be optimised directly, by dividing the progression plots of C from Fig. 5.2 and 5.4 by actuator speed v_{act}, and adding t_{sorting} (a linear function $y = x$), to achieve plots of $\frac{C}{v_{\text{act}}} + t_{\text{sorting}}$, as in Eq. 5.6. The resulting curves for $L = 15$ and 80, $N = 5$ and 60, and an example actuator speed of $v_{\text{act}} = 10\,\text{s}^{-1}$, are shown in Fig. 5.6 (the x-axis ranges were adjusted for improved clarity). The crosses mark the lowest achievable total operational time t_{op} in each case, indicating which sorting method should be used, and with what sorting time limit. Depending on the values of L and N, three different algorithms provided the lowest operational time, viz. the greedy algorithm, the TSP GA and the tabu search algorithm, and the optimal sorting time limit varied between 0.04 s and 5.5 s.

Figure 5.7 presents diagrams showing the optimal sorting algorithm chosen according to minimum t_{op}, at example actuator velocities of $v_{\text{act}} = 10\,\text{s}^{-1}$ and $50\,\text{s}^{-1}$. As the baseline size of the orthogonal and equilateral design space was 100 distance units, it would take the actuation system 10 s and 2 s respectively to transform between the furthest design points. It can be seen that the diagrams are significantly different from those obtained using cost improvement I (Fig. 5.5). Tabu search and the greedy algorithm are still optimal in most cases, but the two figures show that actuator speed v_{act} has a significant impact. Notably, the greedy sorting algorithm is the preferred sorting method at permutation lengths $L \geqslant 60$ at high actuator speed due to its rapid convergence.



(a) $N = 5$ and $L = 80$.

(b) $N = 60$ and $L = 80$.

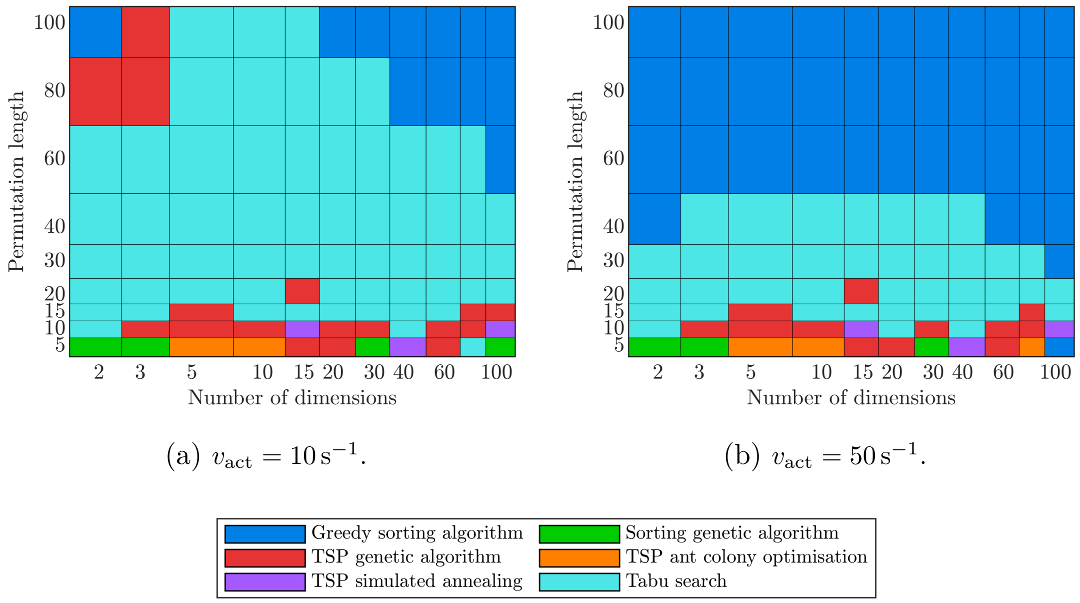
(c) $N = 5$ and $L = 15$.

(d) $N = 60$ and $L = 15$.

FIGURE 5.6: Plots of total operational time t_{op} against sorting time t_{sorting}. The crosses mark the lowest achievable operational time, which indicates the optimal sorting method and sorting time limit.

Analogous plots to those shown in Fig. 5.6 may be generated for any combination of permutation length L, number of dimensions N, design space shape, and actuator speed v_{act}, in order to determine the optimal sorting method and sorting time limit for a particular application.

If actuator ranges or speeds are not the same for all dimensions, the constraints applied when generating permutations, as explained in Section 5.1.2, can be scaled, so that their lengths correspond to maximum actuation time for each dimension. For example, in a 3-dimensional system where maximum actuation times for individual dimensions are 1 s, 2 s and 5 s, the constraints would be scaled to $[1, 101]$, $[1, 201]$ and $[1, 501]$ respectively,



(a) $v_{\text{act}} = 10\,\text{s}^{-1}$.

(b) $v_{\text{act}} = 50\,\text{s}^{-1}$.

FIGURE 5.7: Diagrams showing the optimal sorting algorithm for each design case at $v_{\text{act}} = 10\,\text{s}^{-1}$ and $v_{\text{act}} = 50\,\text{s}^{-1}$.

with actuator speed $v_{\text{act}} = 100\,\text{s}^{-1}$.

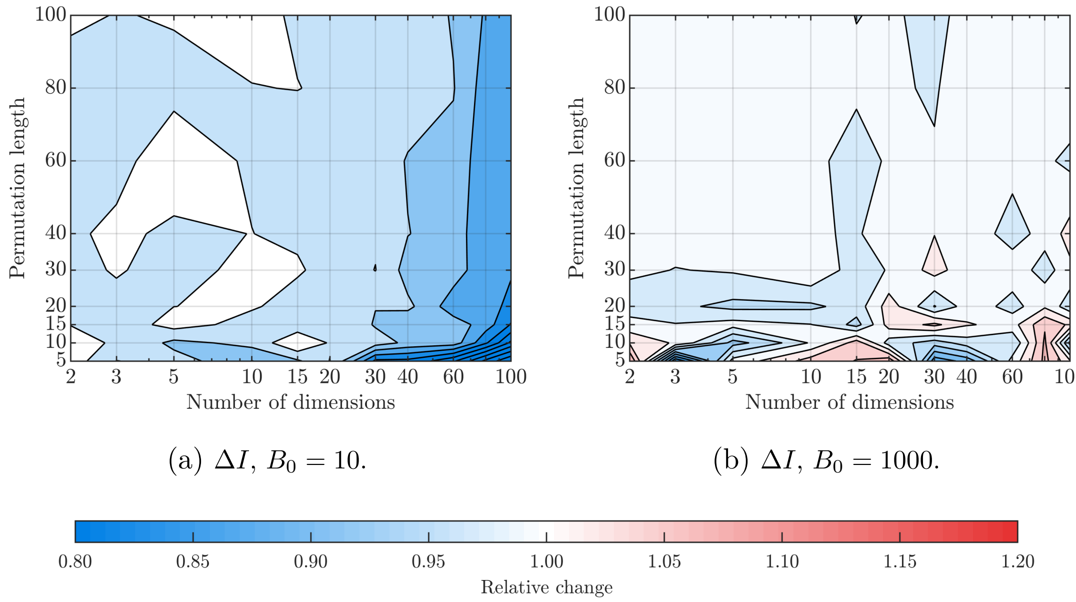
This methodology allows the user to minimise the operational time of any multidimensional system that needs to transform through a series of configurations, as long as the order of configurations is inconsequential.

5.5 Effect of Design Space Size and Shape

5.5.1 Design Space Size

The effect of design space size on sorting effectiveness was examined through a modified version of the baseline simulations described in Section 5.2. Design space size is determined by the baseline dimension length B_0, and the magnitude of B_0 was decreased and increased by one order of magnitude relative to the initial value of $B_0 = 100$, i.e. to $B_0 = 10$ and $B_0 = 1000$. For orthogonal and equilateral design spaces, the cost C is directly proportional to B_0. Therefore, at $B_0 = 10$, the cost before sorting (C) and after sorting ($\mathfrak{C}$) decreased approximately by a factor of 10, and at $B_0 = 1000$ it increased by the same factor, for all algorithms and at all values of L and N. In order to compare the relative sorting effectiveness, change in improvement ΔI was plotted, defined as:

$$\Delta I = \frac{\hat{I}}{I}\,, \tag{5.7}$$



(a) ΔI, $B_0 = 10$.

(b) ΔI, $B_0 = 1000$.

FIGURE 5.8: Contour plots showing the change in improvement ΔI, for dimension lengths of $B_0 = 10$ and $B_0 = 1000$.

where I and $\hat{I}$ are improvement in the baseline ($B_0 = 100$) and modified ($B_0 = 10$ and $B_0 = 1000$) cases respectively. As all algorithms demonstrated similar trends, only the plots for the greedy sorting algorithm are shown in Fig. 5.8.

Figure 5.8a shows that sorting was less effective by approximately 5% for most values of L and N, and by up to 20% at $L = 5$ and $N \geqslant 30$. It is not clear what the cause of this trend is, but problems with over 30 dimensions are unlikely to be optimised with a population of 10 or fewer individuals, so the performance deterioration is of little practical significance. Increasing the design space size to $B_0 = 1000$ had no noticeable effect on sorting effectiveness, as seen in Fig. 5.8b.

5.5.2 Design Space Aspect Ratio

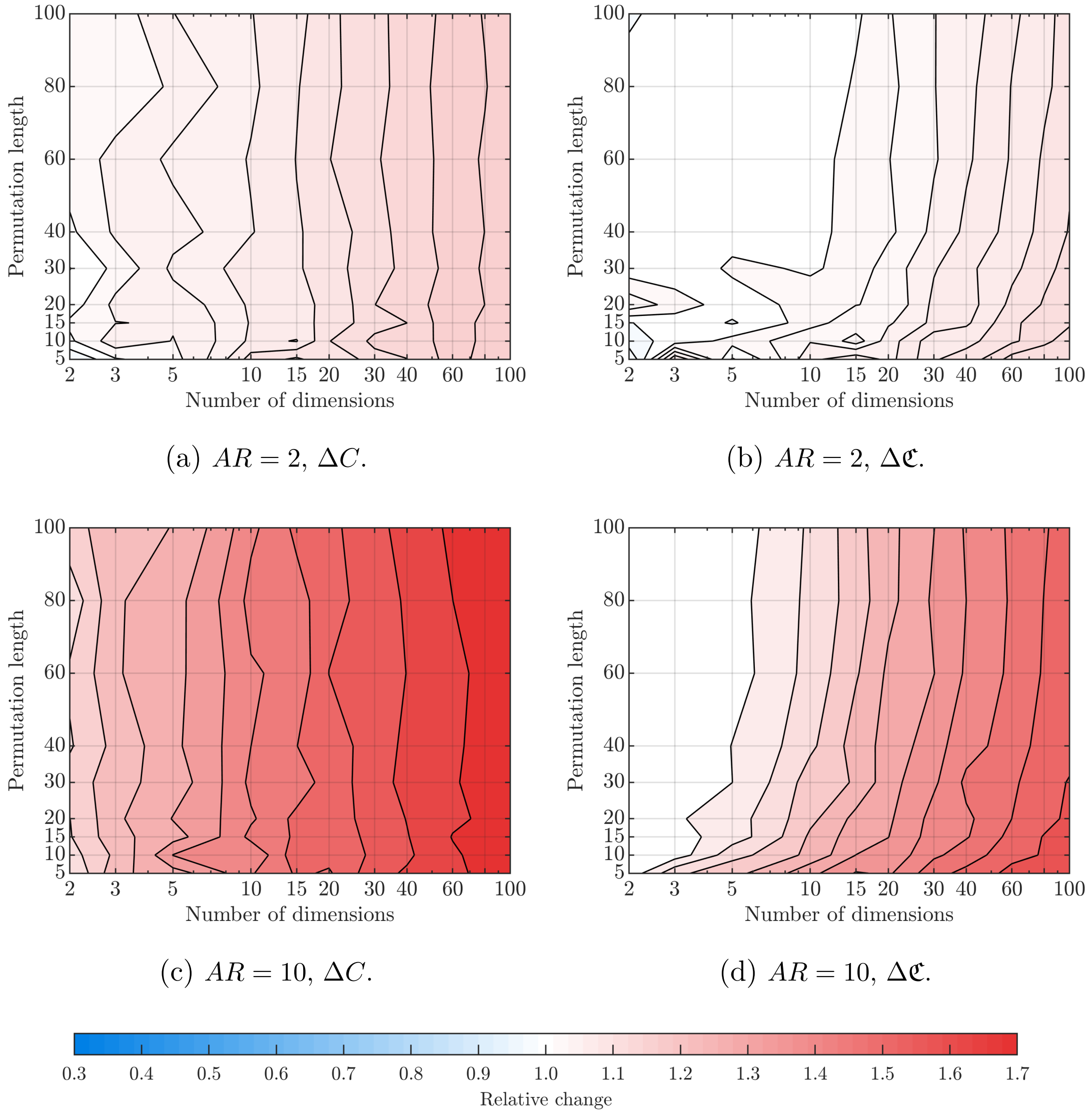
The effect of varying the aspect ratio of the design space was determined by increasing the value of AR to 2 and 10, from the baseline value of 1. The trends exhibited by all algorithms were again similar, therefore only greedy sorting is shown. Figure 5.9 presents contour plots of ΔC and $\Delta \mathfrak{C}$, as defined in. Eq. 5.8 and 5.9, at $AR = 2$ and $AR = 10$:

$$\Delta C = \frac{\hat{C}}{C}, \tag{5.8}$$

$$\Delta \mathfrak{C} = \frac{\hat{\mathfrak{C}}}{\mathfrak{C}}, \tag{5.9}$$

where C and $\mathfrak{C}$ represent cost in the baseline case ($AR = 1$), and $\hat{C}$ and $\hat{\mathfrak{C}}$ are cost in the modified cases ($AR = 2$ and $AR = 10$), before and after sorting respectively.

Figures 5.9a and 5.9c show that, at both $AR = 2$ and $AR = 10$, the average unsorted cost is greater than for an equilateral design space, for all values of L and N. Moreover, the magnitude of this difference grows with increasing N, and reaches approximately 20% at $AR = 2$ and 70% at $AR = 10$. This occurs despite the constant design space size, because the Chebyshev distance between a pair of points is determined by the longest distance in any dimension, as previously explained in Section 5.2. As the maximum aspect ratio increases and the design space is scaled to maintain a constant size, some dimensions become longer than the baseline B_0, and some shorter. The effect of the



(a) $AR = 2$, ΔC.

(b) $AR = 2$, $\Delta\mathfrak{C}$.

(c) $AR = 10$, ΔC.

(d) $AR = 10$, $\Delta\mathfrak{C}$.

FIGURE 5.9: Contour plots showing the change in cost ΔC and $\Delta\mathfrak{C}$, compared to the baseline case, for different design space aspect ratios.

shorter ones becomes smaller as AR increases, so the cost is mainly determined by the longer ones, whose maximum length is directly linked to AR. Moreover, as the number of dimensions N increases, more dimensions have a length longer than the baseline, and therefore it is more likely that any two given points are separated by a large distance in one of the long dimensions. As a result, ΔC increases with both AR and N.

The effect of $AR > 1$ on cost after sorting ($\Delta\mathfrak{C}$) is similar, but less pronounced than before sorting, as shown in Fig. 5.9b and 5.9d, which indicates that the effectiveness of sorting is increased. This may be caused by the fact that, as AR increases, the longer dimensions have a more significant effect on cost than the shorter dimensions, so the effective number of dimensions is reduced. As outlined in Section 5.2, increasing the number of dimensions N has a negative effect on maximum possible improvement $I_{\max}$, therefore a reduced effective dimensionality might cause an opposite trend.

A parameter useful for describing a given design space would be the effective design space size, taking into consideration the baseline dimension length, as well as the aspect ratios of all dimensions. Obtaining this parameter would enable more precise estimation of sorting effectiveness in advance of tests, leading to more accurate operational planning. A derivation of this parameter is recommended for future work.

5.5.3 Design Space Skew

Design space size and aspect ratio define the lengths of dimensions, but a design space remains orthogonal, i.e. a hyperrectangle, regardless of the values of these parameters. In other words, the values of constraints for each degree of freedom are independent of the positions in the remaining DoFs, e.g. one actuator's extension limits are independent of the extension of all other actuators. In order to investigate the effect that design space skew, or non-orthogonality, might have on population sorting effectiveness and average Chebyshev distance, a constraint hyperplane was implemented into the baseline case with an equilateral design space ($AR = 1$), a baseline dimension length of $B_0 = 100$, and random initialisation.

The equation of the constraint hyperplane is defined as:

$$\sum_{j=1}^{N} c_j \cdot x_j = 0\,, \tag{5.10}$$

where c_j are coefficients, and x_j are design variables, corresponding to dimensions $j \in [1, N]$ of the design space. The coefficients were assigned random integers in the range $[-c_{\max}, c_{\max}]$, where $c_{\max}$ is the maximum permitted coefficient value.

In order to maintain a constant design space size despite the randomly generated coefficients, the constraint hyperplane was shifted, so that it always passed through the mid-point of the design space. This shift is represented by a constant k. The design space mid-point was defined as the point coincident with the mid-points of all dimensions (that is, the points equidistant from the minimum and maximum values of the dimensions). The final inequality constraint, which was imposed during the initialisation phase, is as follows:

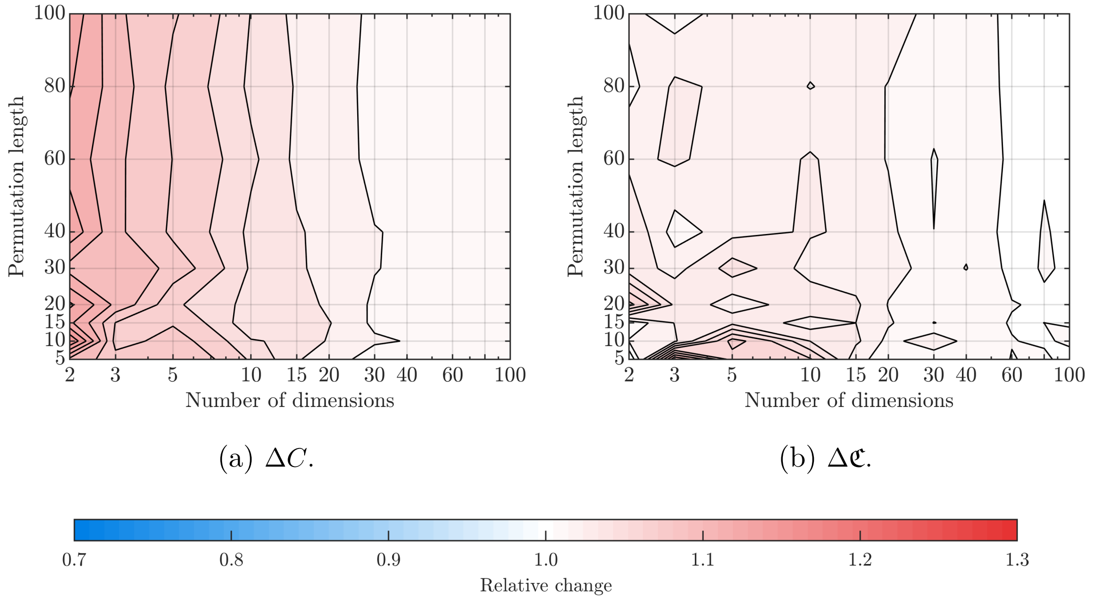
$$\sum_{j=1}^{N} c_j \cdot x_j > k\,. \tag{5.11}$$

As applying this constraint cut the design space size in half, the baseline dimension length B_0 was scaled up according to Eq. 5.12, ensuring that the size was equal to the baseline hypercubic case with $B_0 = 100$:

$$B_0 = 100 \cdot \sqrt[N]{2}\,. \tag{5.12}$$

Simulations were carried out with $c_{\mathrm{max}} = 2$ and $c_{\mathrm{max}} = 5$. However, as the trends in ΔC and $\Delta\mathfrak{C}$ were very similar, only the results obtained using $c_{\mathrm{max}} = 2$ are shown. Figure 5.10 shows contour plots of ΔC and $\Delta\mathfrak{C}$ obtained using the greedy sorting algorithm, which exhibited performance representative of all algorithms.

Figure 5.10a displays the change in cost before sorting ΔC. It can be seen that, at low values of N, the average cost is increased by up to 10%, regardless of the permutation



(a) ΔC.

(b) $\Delta\mathfrak{C}$.

Figure 5.10: Contour plots showing the change in cost ΔC and $\Delta\mathfrak{C}$, compared to the baseline case, for a skewed design space with $c_{\mathrm{max}} = 2$.

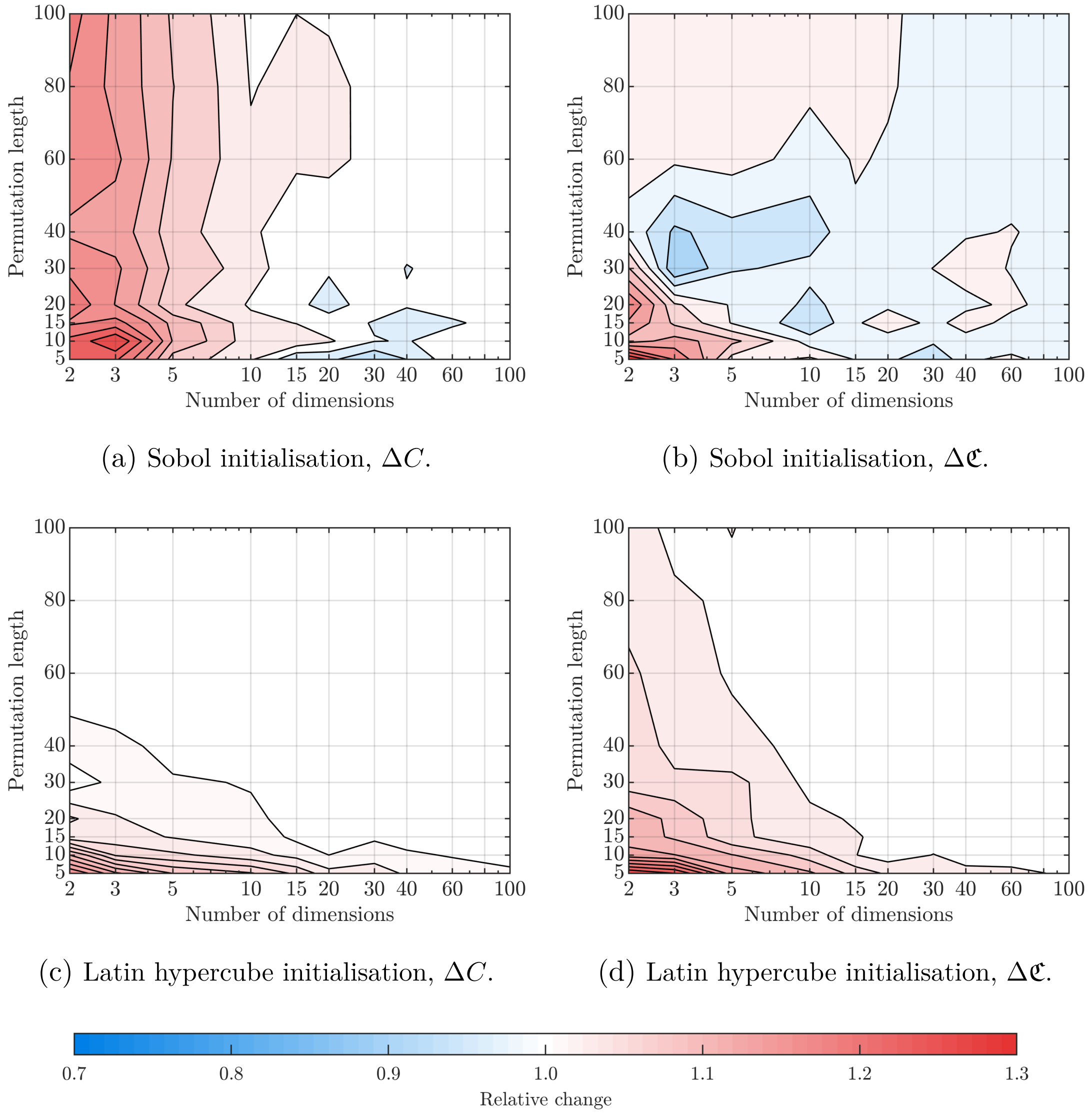
length. The magnitude of this increase decreases with increasing number of dimensions, until it becomes negligible at $N \geqslant 40$. This result can be explained analogously to the trends observed with varying aspect ratio, as outlined in Section 5.5.2. At low numbers of dimensions, applying the constraint hyperplane increases the maximum possible dimension length by a significant proportion (by 41% at $N = 2$ and by 26% at $N = 3$, see Eq. 5.12), so there is a high probability that a given pair of points will be separated by more than the original dimension length in at least one dimension. As the number of dimensions increases, this percentage increase tends to 0%, and thus the average cost tends to that of the baseline, orthogonal design space.

Figure 5.10b shows that the change in cost after sorting is also increased for skewed design spaces, but only by up to 4%, and with a less clear relationship with the number of dimensions. The smaller increase in cost after sorting $\Delta\mathfrak{C}$ compared to cost before sorting ΔC suggests slightly increased sorting effectiveness, especially in design spaces with few dimensions. This is likely a consequence of the increased diversity of dimension lengths introduced by skew, which diminishes the significance of some dimensions relative to others, therefore reducing the effective number of dimensions. This effect is similar to that caused by an increased aspect ratio, and is a further indication that sorting effectiveness is affected not just by the dimensionality of a problem, but also by the relative lengths of its degrees of freedom.

5.6 Effect of Initialisation Method

All results presented hitherto in this chapter were generated using randomly initialised permutations of design points. In order to establish the effect that different distributions of points around the design space might have on sorting effectiveness, comparative tests were carried out using two alternative initialisation methods: Sobol sequences and Latin hypercube sampling (see Section 2.1.3.1). Figure 5.11 presents contour plots of ΔC and $\Delta\mathfrak{C}$ for the two initialisation methods relative to random sampling, obtained using greedy sorting, and an orthogonal and equilateral design space with $B_0 = 100$.

As seen in Fig. 5.11a, initialisation using Sobol sequences significantly increases the average unsorted cost at $N \leqslant 10$, regardless of permutation length. This is likely a consequence of the more uniform distribution of points, which reduces the proportion of closely-neighbouring pairs, which are often generated by random sampling. Moreover, as the generated permutations were not shuffled prior to fitness evaluation, the order of the Sobol sequences might have further affected average unsorted cost. On the other hand, cost after sorting is similar between Sobol sequences and random sampling, suggesting that a Sobol-initialised distribution of points does not significantly increase the total cost once sorting has been applied. The only exception is at $N \leqslant 5$ and $L \leqslant 20$, where $\mathfrak{C}$ is higher than for random sampling by up to 20%. It is hypothesised that this is caused by an even spread of the first few points, which reduces the sorting potential, especially in permutations with few points and dimensions. This suggests that using Sobol sequences to initialise small problems might increase the operational cost of fitness evaluation, although improved coverage of the design space might offset this by yielding more effective convergence.



(a) Sobol initialisation, ΔC.

(b) Sobol initialisation, $\Delta\mathfrak{C}$.

(c) Latin hypercube initialisation, ΔC.

(d) Latin hypercube initialisation, $\Delta\mathfrak{C}$.

FIGURE 5.11: Contour plots showing the change in cost ΔC and $\Delta\mathfrak{C}$, compared to the baseline case, for different initialisation methods.

Latin hypercube sampling has a negligible impact on cost relative to random sampling, both before and after sorting, with the exception of problems with short permutations and low dimensionality, as seen in Fig. 5.11c and 5.11d. The cause of this trend is hypothesised to be similar to the case of Sobol sequence sampling, as the first few points are calculated to provide the best possible coverage of the design space, reducing the likelihood of closely-neighbouring pairs of points, and therefore increasing the average total cost of the permutation, as well as reducing the potential for improvement due to sorting. As permutation length is increased, the relative significance of this effect is diminished, and when the number of dimensions is increased, the probability for pairs of points to be close together in all dimensions also decreases, as previously explained in Section 5.2.

A more detailed analysis of the characteristics of Sobol sequences and Latin hypercubes could shed more light on the trends outlined above, as well as help to predict what effect changing the sampling method might have on problems of different kinds. Although investigating this in detail is beyond the scope of the present study, the results discussed in this section are a strong indication that the choice of initialisation method is consequential for both the operational cost and the effectiveness of sorting, and therefore its impact should be considered when planning experiments.

5.7 Application to Hardware-in-the-Loop Optimisation

In order to validate the results of the population sorting study and the effectiveness of the operational time reduction methodology, sorting was applied to the hardware-in-the-loop aerodynamic optimisation problem discussed in Chapter 4. The large-scale experimental system, as described in detail in Section 3.2, was utilised for the validation. The full-sidewall and open-cavity configuration was used, in order to minimise vibrations and friction in the model. A summary of the degrees of freedom of the optimisation system, together with their ranges, is presented in Table 5.1.

Table 5.1: Degrees of freedom of the HIL optimisation system.

Degree of freedom	Min. value	Max. value	Resolution	Number of points
Ride height	10 mm	90 mm	0.4 mm	201
Rake angle	0°	4.8°	0.058°	84
Diffuser angle	0°	38.5°	0.277°	140

5.7.1 Optimisation Setup

The optimisation was carried out with a genetic algorithm, as it is the most commonly used method in HIL aerodynamic optimisation (see Table 2.1), and GAs were shown to perform consistently in previous tests (see Section 4.2.1). The binary-encoded implementation was used for its superior flexibility, which made it possible to adapt the algorithm to the needs of this study. The algorithm was described in detail in Section 4.1.2, and the following settings were used for the optimisation runs:

- A population size of 30;
- Random initialisation;
- Exponential fitness scaling;
- Roulette selection;
- Elitism, carrying over 1 best individual;
- 2-point crossover with a crossover rate of 80%;
- And fitness-biased mutation with a baseline mutation rate of 8%.

The optimisation objective was to maximise the downforce generated by the model. For every configuration generated by the GA, the following procedure was used to evaluate its performance:

1. Send the target positions to the actuation system;
2. Wait until all actuators reach their targets;
3. Wait for the flow to settle for 1 s;
4. Sample downforce for 1 s at 10 kHz;
5. Calculate the average downforce value.

The optimisation runs were carried out with a 60 min time limit, which was previously shown to allow sufficient convergence with a similar system (see Chapter 4).

5.7.2 Population Sorting Setup

In order to experimentally validate the effectiveness of population sorting, but also to test the sorting optimisation methodology presented in Section 5.4, a total of 9 optimisation runs were carried out in 3 configurations, 3 trials each:

- No population sorting;
- Suboptimal population sorting;
- And optimal population sorting.

The optimal sorting method, as well as the associated optimal sorting time limit, were selected using the methodology from Section 5.4, from the same group of 6 algorithms. The two main parameters, N and L, were 3 and 30 respectively, as the system had 3 degrees of freedom, and the population size of the optimiser was 30.

Actuator speed was measured experimentally, with the system fully assembled, independently for each degree of freedom. This is because different types of actuators were used for the ride height and for the rake and diffuser angles, and the actuators were loaded differently, which also affected their speed.

For this purpose, actuation time, including acceleration and deceleration, was measured for 3 different displacements: 100%, 50% and 12.5% of the actuator range for each degree of freedom, 3 times in each direction, in order to include the effect that actuation distance or direction might have on acceleration or deceleration rates. The measurements were then averaged for each value of displacement, giving 3 data points of actuation time against displacement for each DoF. Linear functions were then fitted, yielding actuator speeds (slopes) and overheads (constants), the latter linked to actuator acceleration, deceleration, and possible overshooting. The results are presented in Table 5.2.

Table 5.2: Actuator speeds of the large-scale diffuser model.

Degree of freedom	Actuator range (mm)	Speed (mm/s)	Overhead (s)
Ride height	80.0	18.21	1.62
Rake angle	33.3	7.82	0.72
Diffuser angle	55.7	7.89	0.40

To calculate the maximum actuation time excluding the above-mentioned overheads, the range of each actuator was divided by its average speed. The resulting times were 4.39 s, 4.26 s and 7.06 s for ride height, rake and diffuser angles respectively, and were discretised by assuming an arbitrary unit of 0.01 s, resulting in constraints of $[1, 440]$, $[1, 427]$ and $[1, 707]$ for the respective degrees of freedom. This method of discretisation ensured that the resulting design space was sufficiently large to pose a challenge for the sorting algorithms, while maintaining the ratios between each DoF. 100 trials were computed using each of the 6 sorting algorithms, and for each trial a population of 30 was randomly initialised using the above-mentioned constraints. Importantly, this procedure was carried out at the same computer as the wind tunnel-based optimisation runs, as computer hardware affects sorting speed, and hence the optimal sorting time limit and method.

Figure 5.12 presents the average total operational time, as defined in Eq. 5.6, versus sorting time, for each method. The values of minimum operational time and optimal sorting time for each method are presented in Table 5.3.

Based on the data from Table 5.3, tabu search was selected as the optimal sorting method, with a sorting time limit of 0.423 s. The greedy algorithm was chosen as the suboptimal sorting method, despite the relatively small performance degradation compared to tabu search, due to its simplicity of implementation and extremely low

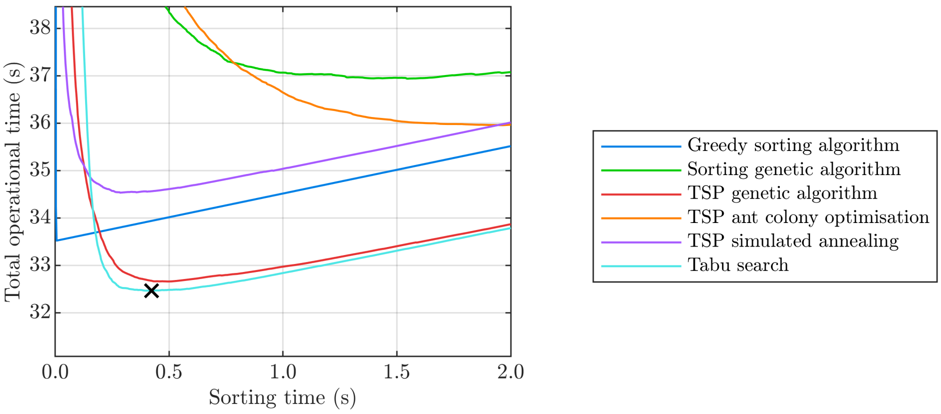


FIGURE 5.12: Plots of total operational time t_{op} against sorting time t_{sorting}, for the large-scale experimental system. The cross marks the lowest achievable operational time.

TABLE 5.3: Minimum operational time and optimal sorting time limit for all sorting methods, for the large-scale experimental system.

Sorting method	Minimum t_{op} (s)	t_{op} reduction	Optimal t_{sorting}
No sorting	86.38	-	-
Greedy sorting	33.52	61.2%	0.009
Sorting GA	36.94	57.2%	1.548
TSP GA	32.66	62.2%	0.498
TSP ACO	35.96	58.4%	1.968
TSP SA	34.54	60.0%	0.288
Tabu search	32.47	62.4%	0.423

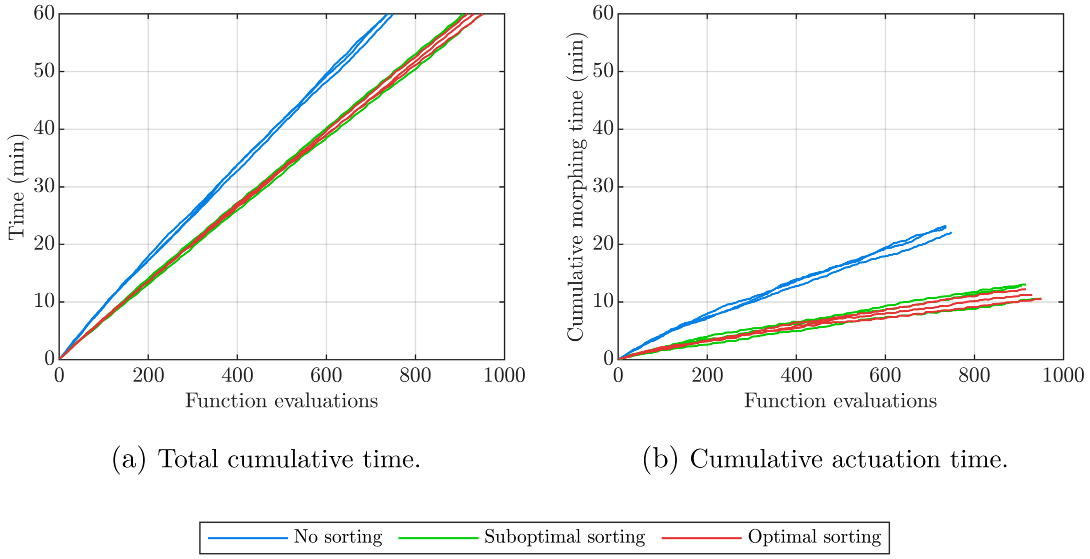
computational cost. As this algorithm exploits all possible starting points in less than a second for permutation lengths up to 1000, no sorting time limit was set.

5.7.3 Results and Discussion

The progression rates of optimisation runs with no sorting, suboptimal and optimal sorting, expressed as the time needed to carry out a certain number of function evaluations, are presented in Fig. 5.13.

Figure 5.13a shows total cumulative time, which is equivalent to wall-clock time. It is immediately apparent that population sorting, regardless of the method, provides a significant improvement, i.e. greatly reduces the time needed to test any number of configurations. The optimiser carried out on average 741, 924 and 932 evaluations within the 60 min time limit, with no sorting, suboptimal, and optimal sorting respectively. This corresponds to increases of 24.7% and 25.8% due to suboptimal and optimal sorting respectively. However, actuation constituted less than 40% of total operational time in these tests, due to the relatively long settling and sampling times (both equal to 1 s). If these were minimised, following the conclusions from Section 4.3, the proportion of actuation time could increase up to 65%, leading to a higher improvement in overall efficiency due to sorting.

In order to isolate the improvement in actuation time from other operations, such as measurement sampling, the plots from Fig. 5.13a are recreated in Fig. 5.13b with all



(a) Total cumulative time.

(b) Cumulative actuation time.

Figure 5.13: Comparison of the pace of the binary-encoded GA without sorting and with two sorting methods.

sorting-independent overheads subtracted, i.e. settling time, sampling time, actuation overheads, and additional overheads. The resultant plots indicate how much actuation time was needed to test a given number of configurations. Whereas the absolute differences in cumulative time are similar to those in Fig. 5.13a, the relative gains due to sorting are much greater.

In order to better understand the reduction of actuation time due to sorting, the total actuation time of consecutive generations of the optimiser are plotted in Fig. 5.14, excluding actuation time between the last configuration of a given generation and the first configuration of the subsequent one. The solid lines correspond to the average values of the 3 trials of each sorting method, and the surrounding patch plots indicate the minimum and maximum values. The crosses indicate the predicted values from the simulations (see Table 5.3).

Importantly, very good agreement can be seen between simulation results and experimental data, validating the methodology presented in Section 5.4. For the first, randomly generated population, the measured average actuation time is within 2 s of the simulated average, both without and with sorting.

As the population of design points converges towards the global maximum, actuation time per generation decreases from ∼ 85 s to ∼ 48 s without sorting, and from ∼ 39 s to ∼ 18 s with sorting. Hence, the reduction in actuation time due to sorting increases from 54% for a randomly-generated population to 62% for a converged population.

The comparison between suboptimal and optimal sorting did not reveal any significant difference in effectiveness. According to the simulations (see Table 5.3), the difference

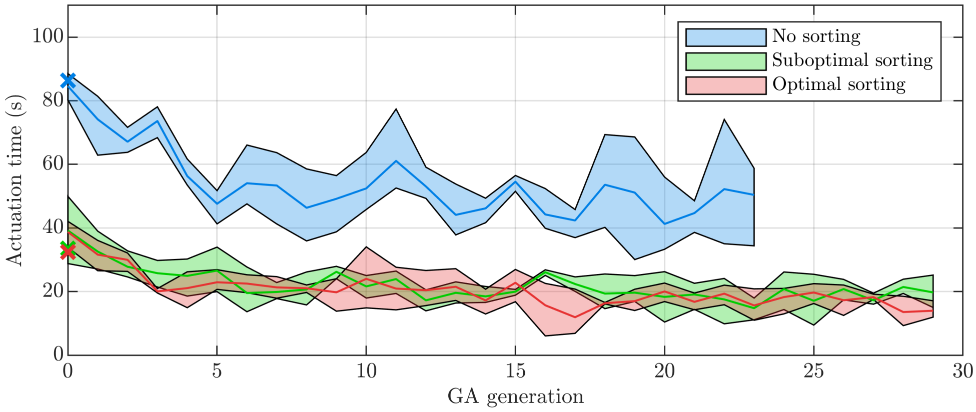


FIGURE 5.14: Comparison of actuation time per generation of the optimiser, without sorting and with two sorting methods.

in actuation time between greedy sorting (suboptimal) and tabu search (optimal) for the first generation is about 3%. However, the noise coming from random population generation and other environmental factors is of even higher magnitude, obscuring variations of this scale. This suggests that, although optimising the sorting process may be beneficial in systems where actuation is slow or costly, the majority of potential gains may be obtained with even the simplest of optimisers. Nevertheless, the experimental results were within the margin of error of the simulations, validating the numerical methodology, and crucially, confirming that the results presented in this chapter are applicable to real-life problems.

5.8 Summary

This chapter presented the results of a numerical study on population sorting for operational time reduction, equivalent to the Chebyshev travelling salesman problem, along with experimental validation of the simulations. One of the aims of this investigation was to determine the impact of several factors, including permutation length and design space shape and size, on sorting effectiveness using the n-dimensional Chebyshev cost metric. Tests on a wide range of combinatorial optimisation algorithms showed that sorting effectiveness, expressed as a relative reduction in operational time, decreases rapidly with increasing dimensionality, from up to 85% for 2-dimensional problems, down to less than 10% for 100-dimensional problems. This phenomenon is caused by the cost metric used, which reduces improvement potential with increasing dimensionality. On the other hand, increasing permutation length increases improvement potential, despite directly increasing problem complexity and hence slowing the convergence rate of the algorithms. These two results collectively indicate that the lower the number of degrees of freedom of the problem and the larger the number of points to sort, the greater the potential reduction in operational time due to sorting, and therefore the greater the incentive for implementing it.

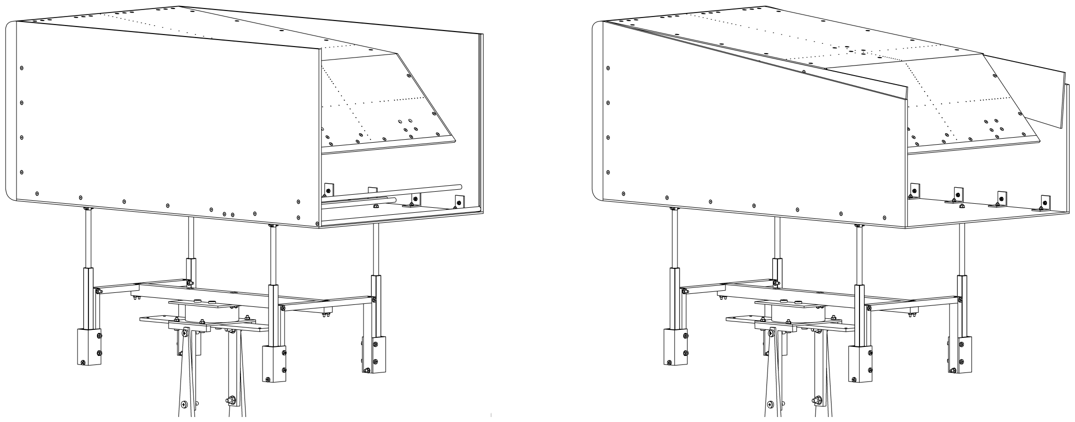
Furthermore, it was observed that this problem is a trade-off between reducing cost through sorting, and reducing the cost of sorting itself. In order to minimise the total operational time, a low-cost, computational methodology was devised and tested on a range of sample problems. The results showed that, given properties of the problem and several combinatorial optimisation algorithms, the optimal sorting method and optimal sorting time limit may be determined. Crucially, this approach is applicable to any combinatorial optimisation problem where reducing operational time is the goal, regardless of the cost metric used, and its low setup and running costs make it attractive even for small-scale problems. In order to validate its effectiveness, the methodology was applied to a hardware-in-the-loop optimisation system with 3 degrees of freedom, and very good agreement was observed between simulation and experimental data. The tests showed a significant reduction in operational time, confirming that population sorting is a viable method for improving the efficiency of hardware-in-the-loop aerodynamic optimisation.

Finally, further tests on a range of design space shapes and sizes showed that the property most closely correlated with Chebyshev cost is the maximum dimension length. Therefore, increasing design space aspect ratio or non-orthogonality, while maintaining a constant number of points, increases the total average cost of a single permutation. However, it was also observed that increasing relative differences between dimension lengths, for example by increasing the average aspect ratio, causes a reduction of the effective dimensionality, hence increasing sorting effectiveness. Replacing random initialisation with a more uniform sampling method, such as Latin hypercubes or Sobol sequences, might also increase the total Chebyshev cost of a permutation, but this effect is significant only in problems with few dimensions and short permutations.

Chapter 6

Aerodynamics of Automotive Diffusers with Rake

The current chapter presents the results and findings of a wind tunnel campaign on the aerodynamics of automotive underbody diffusers in ground effect and in the presence of rake. The first series of tests was carried out using the small-scale experimental system, which was developed for the purpose of the hardware-in-the-loop optimisation tests presented in Chapter 4. The aim of the initial experiments was to gain preliminary insight into the impact of rake by acquiring high-resolution performance maps of the diffuser model with respect to its three degrees of freedom: ride height, rake and diffuser angles. The data collected during these experiments, presented briefly in Section 4.1.3 and analysed in detail in a previous publication [127], revealed that rake has a significant impact on diffuser aerodynamics, including increased downforce, delayed stall with respect to diffuser angle, and eliminated downforce loss at low ground separations. In order to gain greater understanding of the characteristics of those trends, and crucially, to study the underlying flow physics, the large-scale experimental system was used to acquire force and surface pressure data for independent sweeps of ride height, rake and diffuser angles. The effect of quasi-static variations in ride height was previously investigated by other researchers, therefore the analogous ride height sweeps are the main focus of the discussion in this chapter. On the other hand, sweeps of rake and diffuser angles did not reveal significant new flow patterns or performance trends, and are therefore omitted for conciseness. However, a complete dataset, including measurements obtained during sweeps of all three degrees of freedom, is published for the benefit of the research community [128].



(a) Full sidewalls.

(b) Sliding sidewalls.

Figure 6.1: Configurations of the large-scale model used during wind tunnel tests.

Two configurations of the model were tested, i.e. the full-sidewall and sliding-sidewall configurations, as previously outlined in Section 3.2.3, and as illustrated for convenience in Fig. 6.1. The purpose of these tests was to investigate the impact of rake on diffuser aerodynamics both with and without end plates along the entire length of the model, as end plates on the sides of the diffuser itself were previously found to have a significant impact on aerodynamic performance. The full-sidewall configuration has rectangular sidewalls that are always parallel to the ground, regardless of the rake angle, which act as end plates at positive rake. This corresponds to a two-stage diffuser with end plates at a constant height above the ground. Conversely, the sliding-sidewall configuration has a double-sidewall setup, where one pair is attached to the underfloor, deflecting when rake angle is changed. The latter configuration more closely corresponds to previously published rake experiments, where entire models were pitched, although the roof of the model was kept horizontal in this study, in order to isolate the aerodynamic effects of inclining the underfloor. The full-sidewall configuration experiments are, to the author's best knowledge, the first published tests of a diffuser-equipped body with rake and end plates along the entire length of the underfloor.

6.1 Diffuser Aerodynamics with Underfloor End Plates

Discussed first are the tests carried out using the full-sidewall model configuration. Full sweeps of ride height, reducing from the maximum of $h_1/H = 0.290$ down to 0.032 (indicated by solid lines in all relevant figures) and increasing back to 0.290 (dashed lines), were carried out for all combinations of five diffuser angles ($\theta = 0.0°$, 6.2°, 11.2°, 16.3°, 24.7°) and three rake angles ($\gamma = 0.0°$, 1.8°, 4.8°). The above diffuser angle values were selected due to their distinctive performance trends, as reported by Cooper et al. [99], Ruhrmann and Zhang [103] and Jowsey and Passmore [114], and as established during tests of the small-scale model. The two positive rake angles were chosen to represent low- and high-rake configurations.

6.1.1 Zero Rake Angle

Quasi-static variations of ride height at a zero rake angle were previously presented by other researchers, as outlined in Section 2.3.3, and therefore they are discussed first, with a two-fold purpose. Firstly, to validate the design and methodology of this study through a direct quantitative comparison with published results, and secondly, to introduce the reader into the force and surface pressure measurements acquired in this work, and how this data and the trends observed therein may be interpreted. Satisfaction of these two goals is essential for a meaningful discussion of the impact of rake, which is presented in subsequent sections.

Figure 6.2 presents the variation of downforce coefficient C_L with ride height h_1/H, at zero rake ($\gamma = 0.0°$) and all the diffuser angles. The trends are compared to those presented by Cooper et al. [99], as their geometry and ground condition most closely match those from the current investigation. However, there are two key differences between the studies. Firstly, Cooper et al. [99] used a $\sim$ 50% smaller model and a 50% greater wind velocity, resulting in a Reynolds number 25% lower than in this study. Secondly, they used a model with a closed tail cavity, i.e. with diffuser plates extended

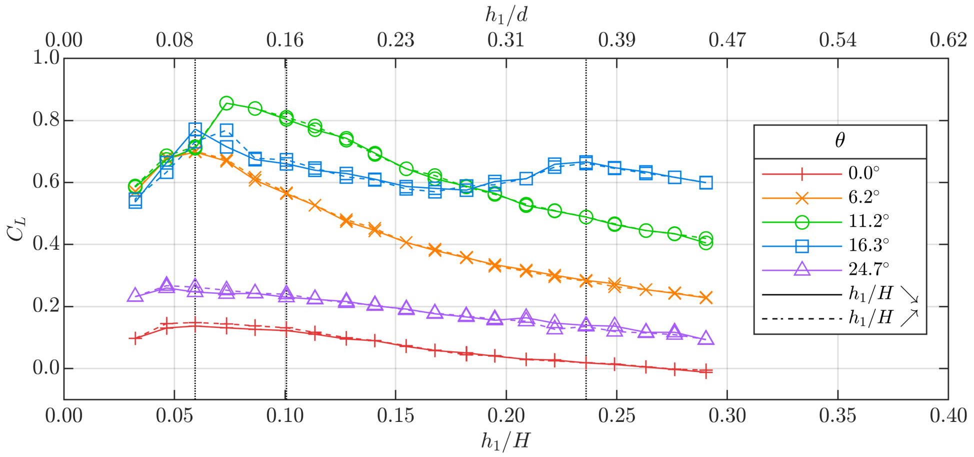


FIGURE 6.2: Plots of downforce coefficient against non-dimensional ride height, at a 0.0° rake angle and a range of diffuser angles. Vertical dotted lines mark the ride heights displayed in Fig. 6.3 to 6.5.

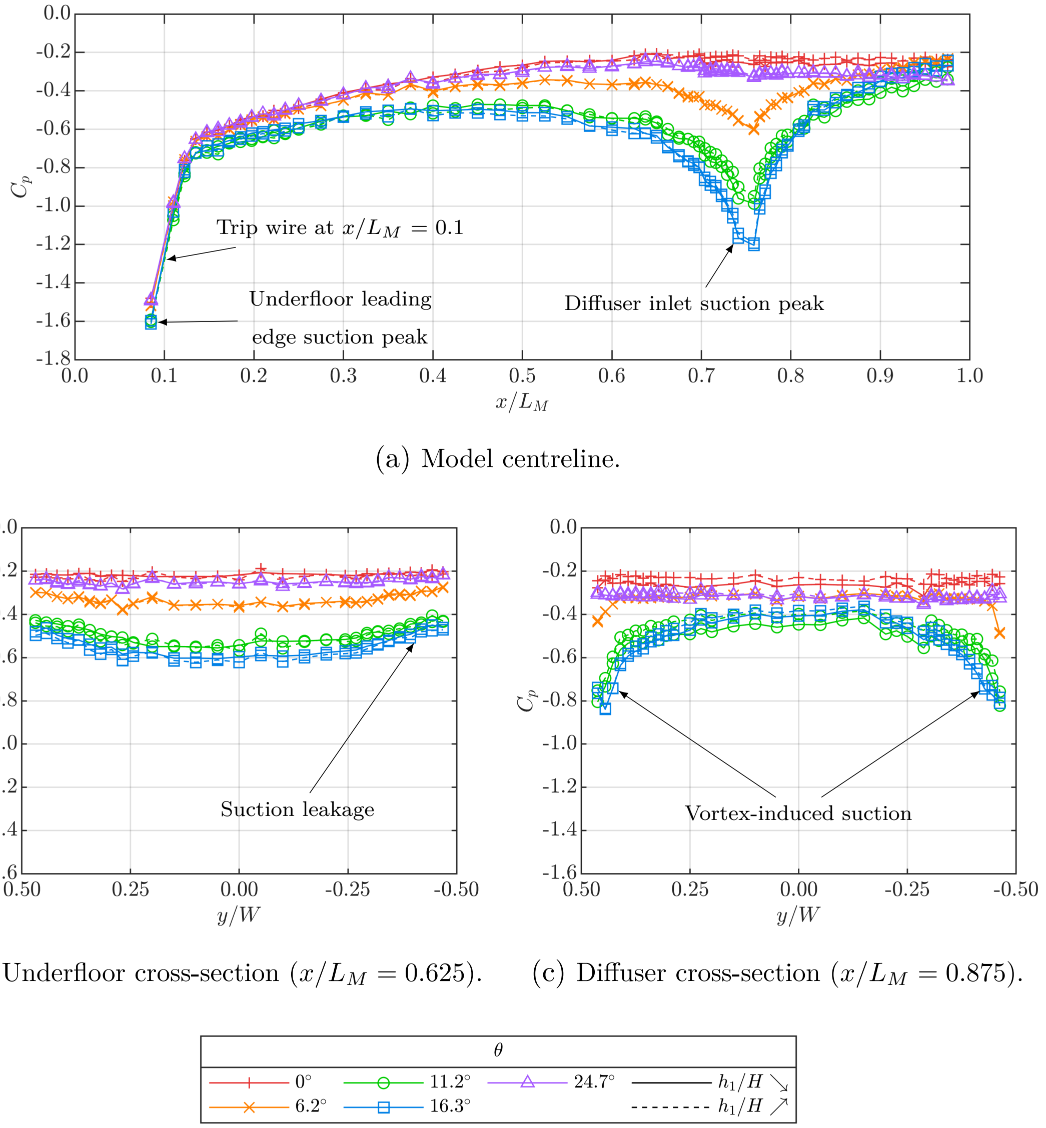
to the trailing edge of the model, and connected to the roof with solid surfaces. This contrasts with the current experiments, which were carried out using a fixed-length diffuser plate and an open cavity, as illustrated in Fig. 6.1a, although the impact of closing the cavity is discussed in Appendix A.

With no diffuser upsweep ($\theta = 0.0°$), a gradual increase in downforce is observed when ride height is reduced, up to a peak of approximately 0.14, followed by a gradual decrease below $h_1/H = 0.059$. Notably, a non-zero $\mathrm{d}C_L/\mathrm{d}h_1$ gradient is present at the maximum ride height of $h_1/H = 0.290$, confirming that the model is in ground effect even at that height. A 6.2° diffuser adds a large amount of downforce at the maximum ride height, but also greatly reinforces the impact of ground effect, increasing peak downforce to $C_{L_{\max}} = 0.7$ at $h_1/H = 0.059$, followed by a gradual reduction. Cooper et al. [99] observed analogous variations, with downforce peaks of 0.2 at $\theta = 0.0°$ and 0.82 at 6.82°, both occurring at $h_1/H = 0.05$ to 0.06. Their increased magnitude of $C_{L_{\max}}$ at $\theta \approx 6°$ may be explained by the 0.6°-steeper angle of the diffuser.

Increasing the diffuser angle to 11.2° increases C_L by approximately 0.2 at most ride heights, with a peak of $C_{L_{\max}} = 0.86$, but downforce reduction occurs slightly earlier, at $h_1/H = 0.074$. The 16.3° diffuser experiences an early downforce reduction at $h_1/H = 0.209$, but subsequently C_L keeps increasing, up until a peak of 0.77, slightly lower than at $\theta = 11.2°$. Similar values of peak downforce were reported by Cooper et al. [99] at similar angles (0.84 at $\theta = 11.54°$ and 0.72 at $\theta = 15.59°$), and the initial downforce reduction at $\theta \approx 16°$ occurred at a similar ride height ($h_1/H = 0.24$) in their study. However, the authors observed a similar temporary downforce drop for the 11.54° diffuser between $h_1/H = 0.12$ and 0.06, which did not occur in this study. Notably, all diffusers between $\theta = 6.2°$ and 16.3° generate the same levels of downforce at $h_1/H \leqslant 0.046$, a trend also observed by Cooper et al. [99].

Finally, the 24.7° diffuser appears to be stalled at all ride heights, producing only slightly higher downforce than the $\theta = 0.0°$ configuration. The only region of hysteresis can be observed at $\theta = 16.3°$, where the downforce peak is delayed when ride height is being increased. Neither of these trends can be compared to the study by Cooper et al. [99], as increasing ride height and diffusers above 16° were not tested. However, Jowsey and Passmore [114] reported a significant downforce loss at $\theta = 25°$ with a static ground, which became even more severe at $\theta \geqslant 28°$, whereas Ruhrmann and Zhang [103] observed significant hysteresis at $\theta \geqslant 15°$.

Figure 6.3 shows static pressure coefficient plots at $h_1/H = 0.236$, where all diffuser angles are still experiencing downforce enhancement. The centreline plot reveals a suction peak near the underfloor leading edge, caused by flow acceleration around the model nose, followed by strong pressure recovery. Whereas other researchers observed a separation bubble just downstream of the leading edge of the underfloor [99, 114, 116], there is no evidence of it in Fig. 6.3a, with a relatively smooth gradient change present instead. Although a trip wire was located at $x/L_M = 0.1$ in these experiments, it is possible that the boundary layer transitioned even earlier, due to the rough surface of the machined nose. This would make it less likely to separate than in the past studies, where no tripping was used. In the diffuser section ($0.75 \leqslant x/L_M \leqslant 1$), significant



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

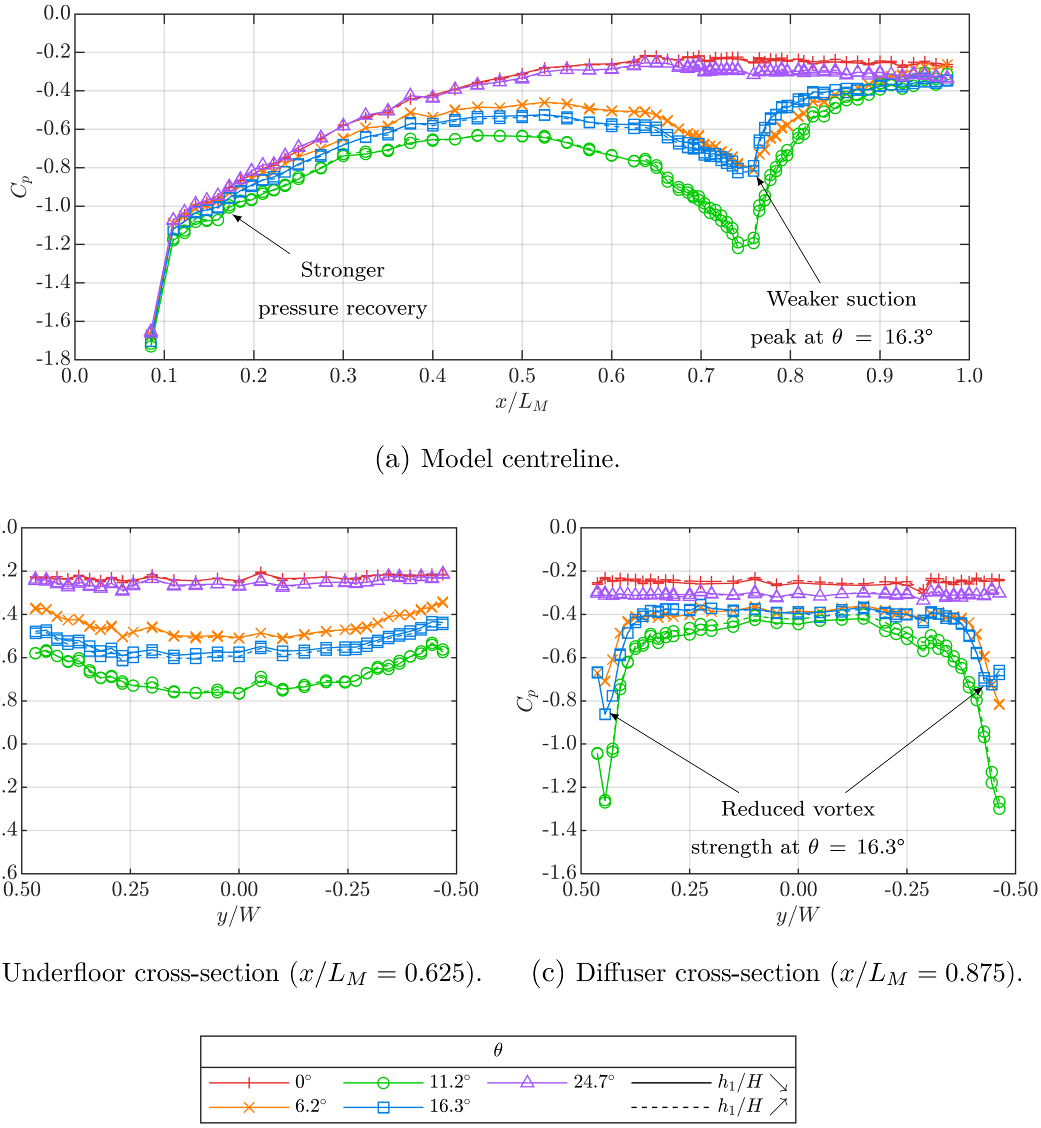
FIGURE 6.3: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, at $h_1/H = 0.236$, $\gamma = 0.0°$, and a range of diffuser angles.

pressure recovery can be observed at angles between $\theta = 6.2°$ and 16.3°, with the suction peak at the diffuser inlet ($x/L_M = 0.75$) growing with increasing diffuser angle. This increased suction propagates upstream, causing reduced C_p under the majority of the underfloor, both along the length of the model (see Fig. 6.3a), and across its width (see Fig. 6.3b).

The underfloor cross-section plot (Fig. 6.3b) reveals distinct suction leakage towards the side edges of the model, which is a symptom of flow entrainment caused by a pressure difference between the underfloor and the sides of the model. The diffuser cross-section plot (Fig. 6.3c) shows large suction regions near the sidewalls at $\theta = 11.2°$ and 16.3°, which indicate the presence of a pair of streamwise, counter-rotating vortices. Vortex-induced suction is also present at $\theta = 6.2°$, although much smaller in width and magnitude. Finally, the 24.7° diffuser is completely stalled, with no pressure recovery or diffuser vortices. However, the diffuser exit pressure $C_{p_{\text{exit}}}$, defined as surface pressure at the trailing edge of the diffuser plate, is slightly lower than at all the other diffuser angles (see Fig. 6.3a), which is the cause of the slightly increased downforce relative to the $\theta = 0.0°$ configuration, as seen in Fig. 6.2. It is worth noting that the C_p bumps present at several pressure tap locations (most notably at $x/L_M = 0.825$ at the model centreline, at $y/W = -0.05$ at the underfloor cross-section, and at $y/W = -0.29$ at the diffuser cross-section) are present in all pressure plots in this chapter, and are believed to be artefacts of transducer calibration or pressure tap installation precision, rather than geometry-induced aerodynamic effects.

As ride height is reduced below $h_1/H = 0.236$, the 16.3° diffuser experiences a small downforce reduction (see Fig. 6.2). Centreline pressure plots revealed that the diffuser inlet suction peak, which reached a magnitude of -1.15 at $h_1/H = 0.236$ (Fig. 6.3a) started gradually decreasing in strength at lower ride heights, down to -0.73 at $h_1/H = 0.101$ (Fig. 6.4a). Furthermore, the pressure recovery profile along the diffuser centreline became less gradual, with a sharp pressure increase just downstream of the diffuser inlet, followed by a shallow gradient towards the exit, as seen in Fig. 6.4a. Both of these trends were previously attributed to the appearance of a separation bubble on the diffuser ramp surface [102, 110, 116]. However, both vortices are still present despite the localised flow separation, and the majority of downforce is preserved.

A further reduction in ride height causes an increase in downforce at all diffuser angles. Static pressure plots at $h_1/H = 0.101$, where the 6.2° and 11.2° diffusers are still under-going downforce enhancement, are displayed in Fig. 6.4. The curves show strengthened suction regions at the diffuser inlet and near diffuser end plates at $\theta = 6.2°$ and $11.2°$, which are responsible for the increased downforce, whereas the 16.3° diffuser remains partly stalled, with a weaker suction peak at the diffuser inlet, and reduced vortex-induced suction. Importantly, the magnitude of the spanwise suction peaks is used as an indicator of vortex strength throughout this work, as the two have previously been shown to be closely correlated [102, 110, 116]. Furthermore, stronger pressure recovery compared to $h_1/H = 0.236$ is visible near the underfloor leading edge (between $x/L_M = 0.1$ and 0.3) at all the diffuser angles. This trend is analogous to past experiments [99, 114], and is likely caused by reduced flow entrainment along the length of



(a) Model centreline.

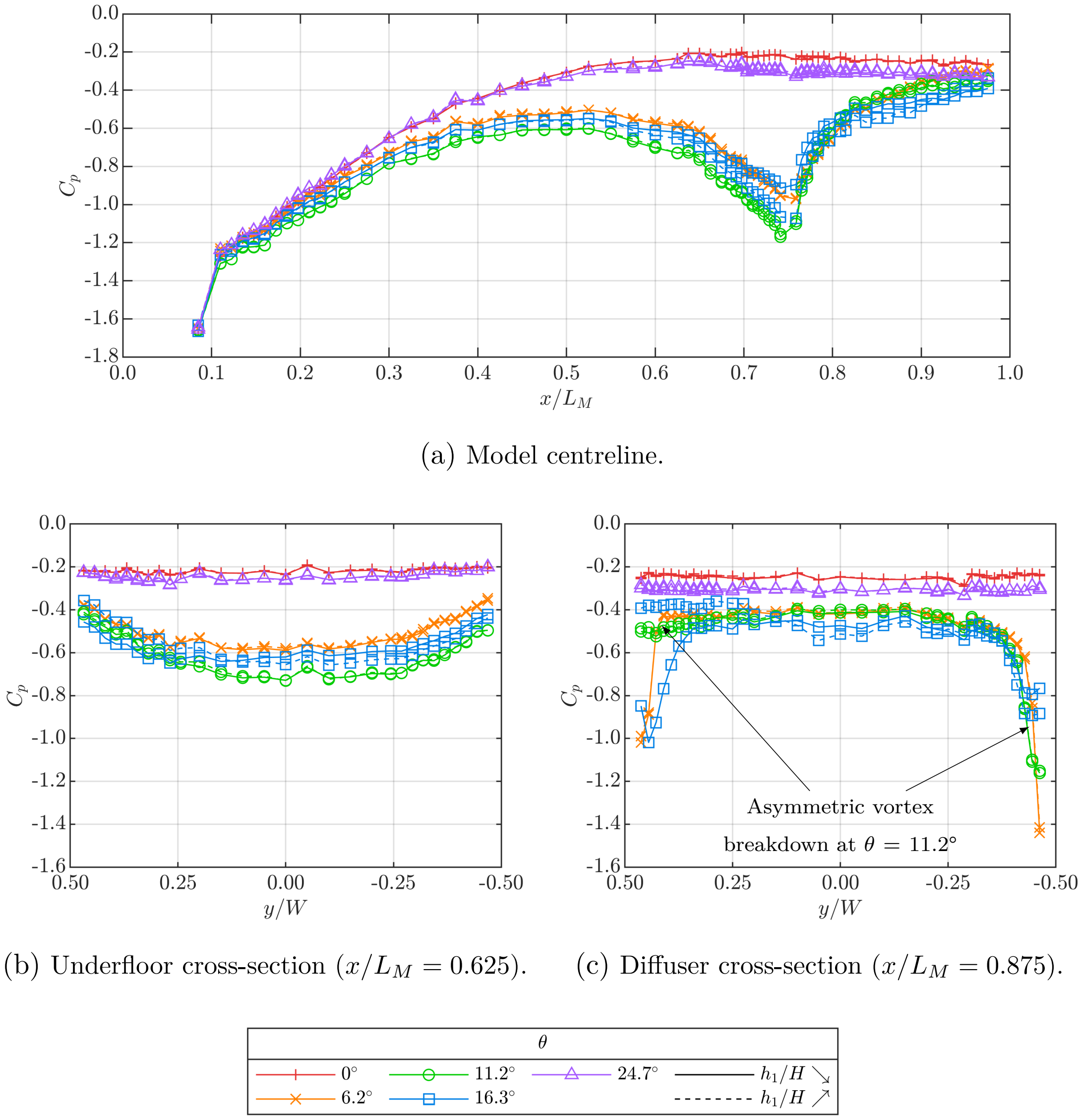
(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.4: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, at $h_1/H = 0.101$, $\gamma = 0.0°$, and a range of diffuser angles.

the underfloor.

Figure 6.5 displays the underbody pressure distribution at $h_1/H = 0.059$, where the 6.2° and 16.3° diffusers are at peak downforce, while the 11.2° diffuser is already experiencing downforce reduction (see Fig. 6.2). The deteriorated performance at $\theta = 11.2°$ is caused by asymmetric vortex breakdown, as evidenced by the disappearance of vortex-induced suction on one side of the model [110] (see Fig. 6.5c). However, a strong suction peak at the diffuser inlet is still maintained. The 6.2° and 16.3° diffusers retain both vortices and their strength is greater than at $h_1/H = 0.101$, but significant asymmetry in vortex strength has appeared in both cases. It is noteworthy that the 16.3° diffuser experiences delayed stall compared to $\theta = 11.2°$. This indicates that the reduced adverse pressure



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.5: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, at $h_1/H = 0.059$, $\gamma = 0.0°$, and a range of diffuser angles.

gradient caused by partly separated flow, as evidenced by the weaker diffuser inlet suction peak at $\theta = 16.3°$ seen in Fig. 6.4, contributes to delaying vortex breakdown and the resulting sudden downforce loss.

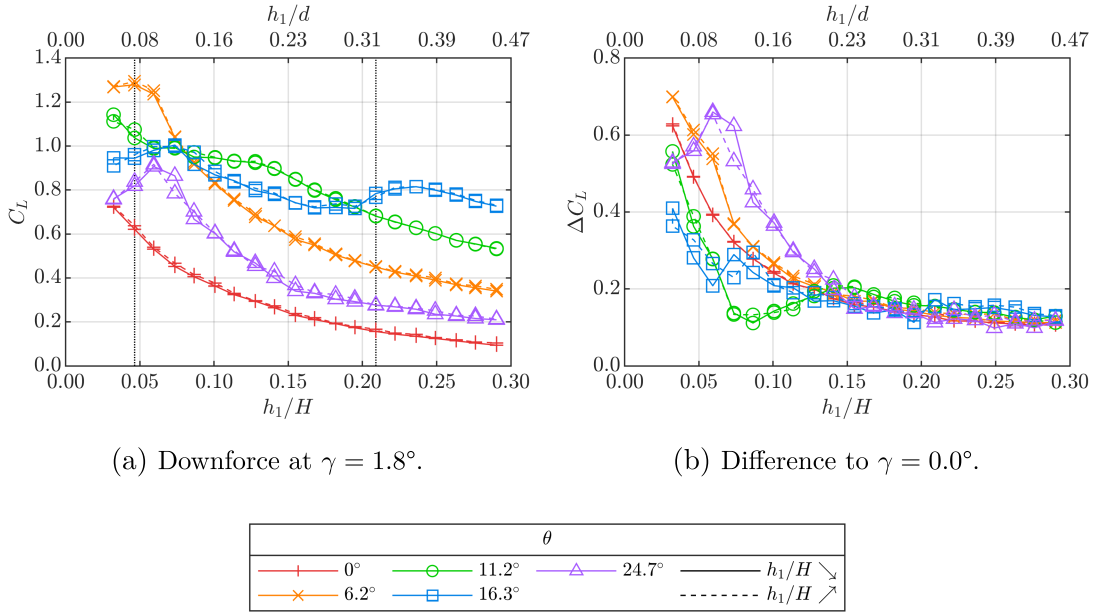
At the lowest ride height of $h_1/H = 0.032$, all diffusers between 6.2° and 16.3° had undergone asymmetric vortex breakdown, but retained the diffuser inlet suction peaks, although with slightly reduced peak magnitudes. Notably, vortex breakdown occurred on the same side in every case, suggesting that small asymmetries in model geometry or setup may have been present.

The trends discussed in this section are in good agreement with previously published results, both qualitatively and quantitatively. The values and variations of downforce coefficient measured at a range of ride heights and diffuser angles closely correspond to those observed by Cooper et al. [99] in analogous tests. Furthermore, all the main flow patterns with respect to decreasing ride height were observed, including growing suction peaks at the underfloor and diffuser inlets, the formation of a separation bubble on the diffuser surface, and a pair of counter-rotating vortices and their subsequent asymmetric breakdown.

6.1.2 Moderate Rake Angle

Figure 6.6 presents the trends in downforce with ride height at a moderate rake angle of $\gamma = 1.8°$, as well as the differences in downforce compared to the baseline, zero-rake case discussed above, in Section 6.1.1. The plots show that downforce at the maximum ride height ($h_1/H = 0.290$) is increased by between 0.1 and 0.15 relative to the $\gamma = 0.0°$ case. Furthermore, downforce gradients with ride height $\mathrm{d}C_L/\mathrm{d}h_1$ are increased at all diffuser angles, which leads to significantly higher downforce peaks than at zero rake. In addition, no downforce reduction takes place at $\theta \leqslant 11.2°$, even at the lowest attainable ride height.

The surface pressure distribution at $h_1/H = 0.209$, where the 16.3° diffuser is experiencing the initial downforce reduction, is shown in Fig. 6.7. Pressure profiles across the underfloor in Fig. 6.7b reveal that the spanwise pressure gradients seen at $\gamma = 0.0°$ in Fig. 6.3b, 6.4b and 6.5b are eliminated, and this occurs at all ride heights and diffuser angles at $\gamma = 1.8°$. Moreover, small regions of low C_p are apparent near the sides of the underfloor, indicating the formation of small edge vortices. When no end plates are present along the underfloor, as in the $\gamma = 0.0°$ case, high-pressure flow from the sides



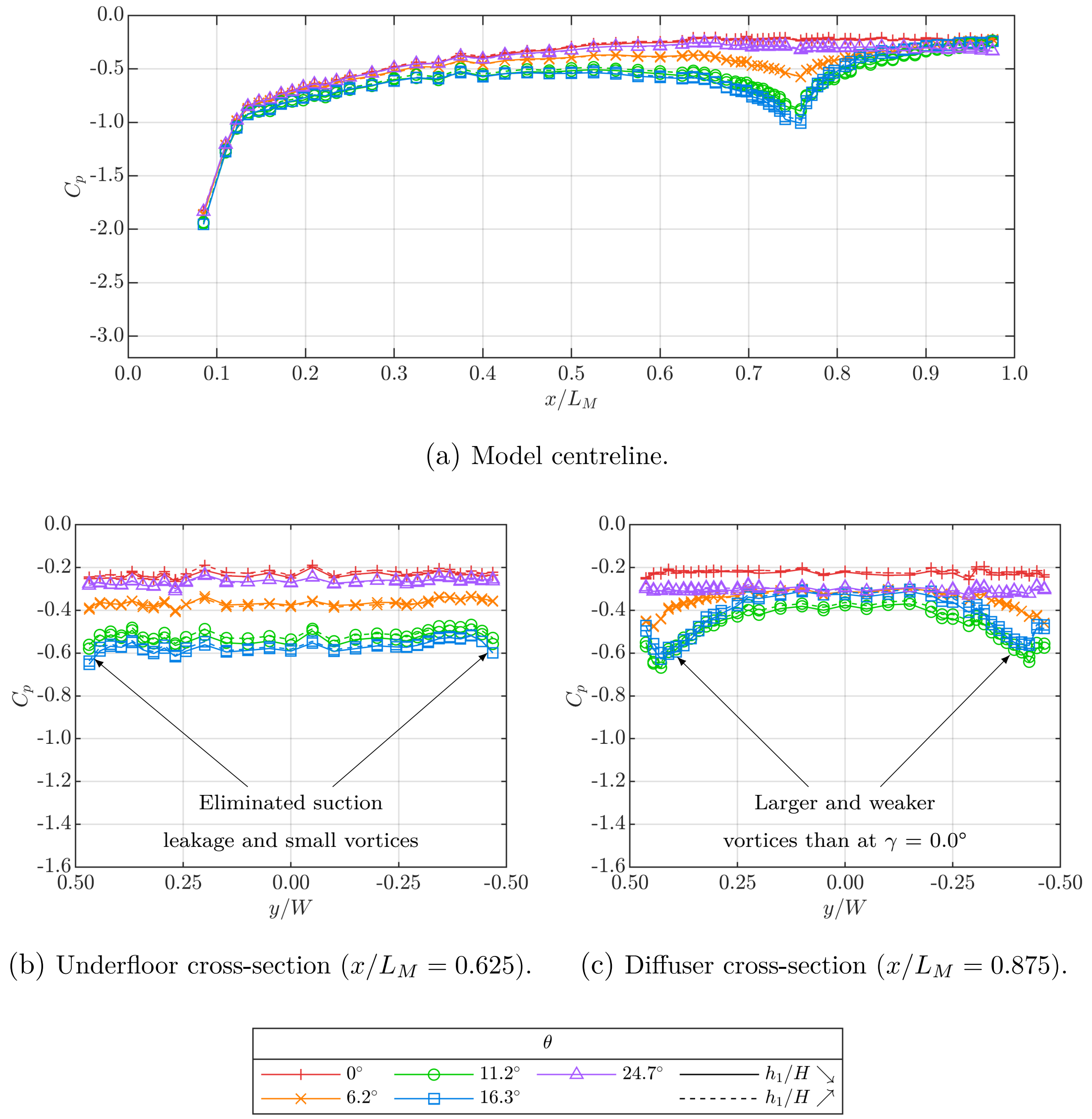
(a) Downforce at $\gamma = 1.8°$.

(b) Difference to $\gamma = 0.0°$.

Figure 6.6: Plots of downforce coefficient against non-dimensional ride height, at a 1.8° rake angle and a range of diffuser angles, and the differences to $\gamma = 0.0°$. Vertical dotted lines mark the ride heights displayed in Fig. 6.7 and 6.9.

of the model is entrained underneath, and then moves inwards while being convected downstream, as evidenced by previous oil flow experiments [116]. As a result, static pressure under the model is increased, leading to reduced downforce. This situation is analogous to a lift-generating wing with no end plates, where the pressure difference between the two surfaces causes vortex roll-up and a lift reduction. On the other hand, when the underfloor is raked and end plates are present, the entrained flow is rolled-up into vortices and carried downstream with their high streamwise momentum, which prevents further entrainment into the central part of the underfloor. Thus, the impact of the high-pressure flow from the sides of the model is decreased, which appears as the reduced spanwise pressure gradients seen in Fig. 6.7b.

The existence of the underfloor vortices is further corroborated by the pressure distributions across the diffuser, as illustrated in Fig. 6.7c. The vortex-induced suction regions are much wider and further inboard than at $\gamma = 0.0°$, but also of slightly reduced magnitudes. Mahon et al. [111] and Ehirim et al. [104, 112] established that vortices in a diffuser with no rake expand, reduce in strength and move inboard as they propagate downstream. Therefore, the current results suggest that the vortices observed in the diffuser (at $x/L_M = 0.875$) had not formed at the diffuser inlet, but further upstream, which is consistent with their visibility at $x/L_M = 0.625$ in Fig. 6.7b.



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.7: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, at $h_1/H = 0.209$, $\gamma = 1.8°$, and a range of diffuser angles.

The plots in Fig. 6.7 also show the onset of stall of the 16.3° diffuser, with slightly reduced diffuser vortex strength and weaker pressure recovery along the diffuser centreline. Furthermore, the streamwise pressure recovery under the front of the underfloor is stronger compared to the $\gamma = 0.0°$ case at similar ride heights. This is caused by the inclination of the underfloor, which induces additional pressure recovery, leading to increased suction under the nose.

To understand the aerodynamic behaviour of the model as ride height is reduced further, it is helpful to track the magnitudes of suction peaks on the underbody. Fig. 6.8 shows their progression with ride height at four locations:

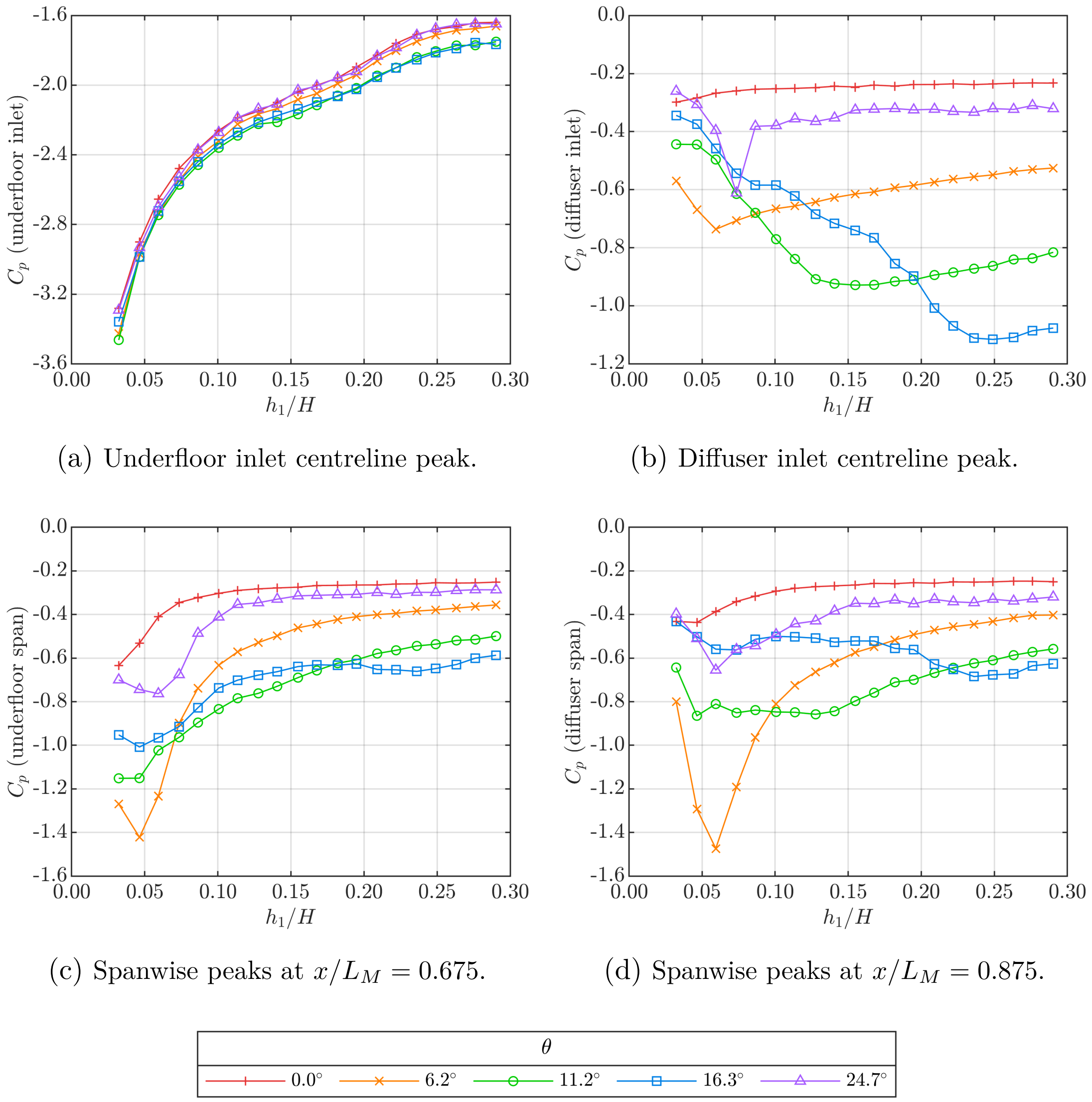
- The underfloor inlet, defined as the centreline tap closest to the front of the model

($x/L_M = 0.085$);

- The diffuser inlet, defined as the centreline tap inside the diffuser and closest to the inlet ($x/L_M = 0.759$);
- The underfloor span, defined as the lowest C_p across the span at $x/L_M = 0.625$;
- And the diffuser span, defined as the lowest C_p across the span at $x/L_M = 0.875$.

The lowest C_p was measured at the underfloor inlet for every configuration, and it decreased from ~ -1.8 to ~ -3.4 as ride height was reduced from $h_1/H = 0.290$ to 0.032, as seen in Fig. 6.8a. This was caused by the aforementioned additional pressure recovery along the entire length of the underfloor, which acts as a preliminary diffuser, expanding the flow and therefore inducing even higher flow velocity and lower pressure under the nose. This phenomenon is responsible for a large proportion of the additional downforce relative to $\gamma = 0.0°$, where the underfloor inlet C_p does not exceed -1.75, and is broadly independent of flow attachment inside the diffuser. Simultaneously, the vortex-induced suction at the underfloor span (see Fig. 6.8c) also grows at accelerating rates, but the characteristics of this growth are strongly related to the flow separation and vortex strength inside the diffuser, and some suction is lost at the lowest ride height at all positive diffuser angles.

The suction peaks at the diffuser inlet and span follow analogous patterns (see Fig. 6.8b and 6.8d). At the maximum ride height ($h_1/H = 0.290$), the magnitudes of the peaks grow with increasing diffuser angle at an approximately linear rate, and drop significantly for the 24.7° diffuser, closely following the downforce trend from Fig. 6.6a. As ride height is reduced, the 6.2°, 11.2° and 16.3° diffusers experience early stall (at $h_1/H = 0.046$, 0.114 and 0.222 respectively, as evidenced by the significant drops in suction in Fig. 6.8b), with similar characteristics to the $\gamma = 0.0°$ case. First, as the adverse pressure gradient in the diffuser grows, the flow briefly separates off the ramp before being reattached by the vortices, forming a separation bubble, reducing pressure recovery along the diffuser and hence weakening the diffuser inlet suction peak, as seen in Fig. 6.8b. Simultaneously, the growing adverse pressure gradient causes the vortices to expand and their cross-flow velocities to decrease [111], which results in reduced suction across the diffuser span, as seen in Fig. 6.8d. This effect propagates upstream, as evidenced by the reduced suction across the underfloor seen in Fig. 6.8c, although to a varying degree depending on the diffuser angle. As ride height is reduced further, the increasing adverse pressure gradient causes the separation bubble to grow and move



(a) Underfloor inlet centreline peak.

(b) Diffuser inlet centreline peak.

(c) Spanwise peaks at $x/L_M = 0.675$.

(d) Spanwise peaks at $x/L_M = 0.875$.

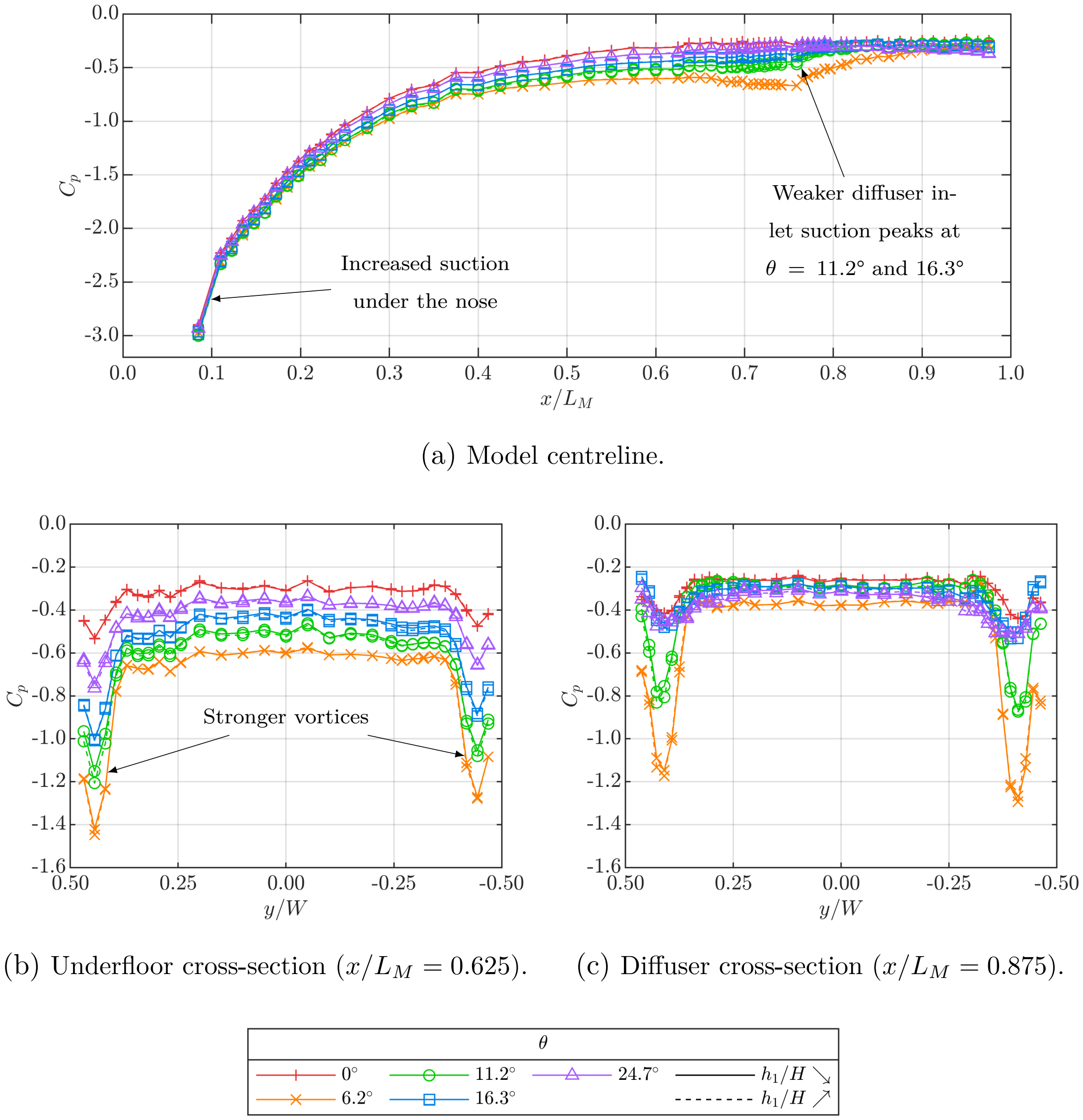
FIGURE 6.8: Progression of suction peaks under the model with decreasing ride height, at $\gamma = 1.8°$ and a range of diffuser angles. In Fig. (c) and (d), the lowest C_p values from each spanwise distribution are shown for each configuration.

upstream, and leads to further reductions in the strength of the diffuser inlet peak and the streamwise vortices. However, despite the weaker suction in the diffuser, the growing suction under the nose of the model is sufficient to prevent downforce loss even at the lowest ride heights, as seen in Fig. 6.6a for the 6.2° and 11.2° diffusers. These phenomena are illustrated in Fig. 6.9, which shows the underbody pressure distribution at $h_1/H = 0.046$.

It is worth noting that the ride heights at which the initial flow separation occurs at $6.2° \leqslant \theta \leqslant 16.3°$, as evidenced by the weakening of the diffuser inlet suction peaks seen in Fig. 6.8b, correspond to diffuser area ratios h_3/h_2 between 1.70 and 1.90, suggesting a strong correlation between the flow characteristics and the area ratio of the diffuser.

However, the initial separation appeared at the same or similar ride heights at zero rake, which correspond to higher values and a wider range of h_3/h_2. These results indicate that, although a relationship between the area ratio and diffuser stall exists, the value of the ratio may not be used to predict whether stall has already occurred.

A secondary downforce drop may be observed for the 16.3° diffuser at $h_1/H \approx 0.059$. Inspecting the centreline pressure distribution, as seen in Fig. 6.9a, revealed a complete loss of pressure recovery in the diffuser, indicating that the flow separated off the diffuser inlet and did not subsequently reattach. At $\gamma = 0.0°$, this occurred together with asymmetric vortex breakdown, but symmetrical vortices are still present in the diffuser at the 1.8° rake angle (see Fig. 6.9c), similar to the 6.2° and 11.2° diffusers. This might



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

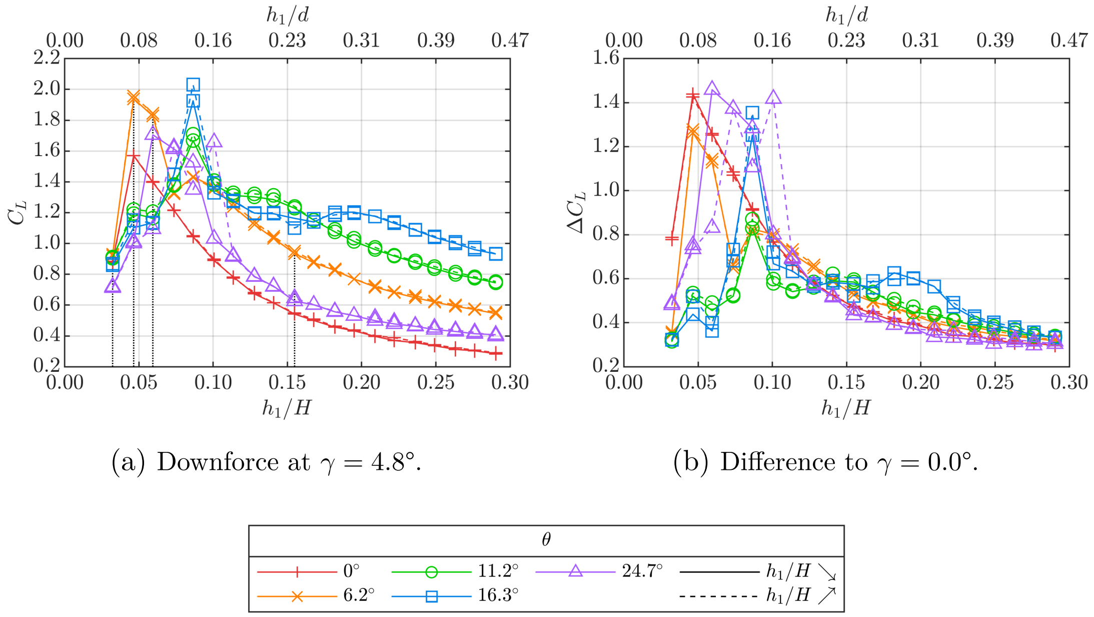
(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.9: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, at $h_1/H = 0.046$, $\gamma = 1.8°$, and a range of diffuser angles.

be caused by their increased strength, i.e. axial and cross-flow velocities, upstream of the diffuser inlet, as shown in Fig. 6.9b, which would allow them to withstand a higher adverse pressure gradient in the diffuser. Furthermore, the underfloor vortices are sufficiently strong and resistant to breakdown at $h_1/H \leqslant 0.074$ to remain partly attached in the 24.7° diffuser (Fig. 6.9c), significantly increasing downforce for those configurations.

6.1.3 High Rake Angle

Increasing the rake angle to 4.8° results in increased downforce at all ride heights and diffuser angles, as seen in Fig. 6.10. The initial trend with decreasing h_1/H is accelerating downforce increase, similar to the $\gamma = 1.8°$ case, but with higher magnitudes of C_L, and the stall at $\theta = 16.3°$ occurring at a lower ride height. The latter, evidenced by a drop in downforce at $h_1/H = 0.168$ compared to $h_1/H = 0.209$ at $\gamma = 1.8°$ (see Fig. 6.6a), may be caused by the reduced effective diffuser area ratio, as increasing the rake angle has the effect of moving both the inlet and the outlet of the diffuser away from the moving ground. However, the performance trends are significantly different at $h_1/H \leqslant 0.101$, with all diffusers experiencing abrupt peaks in downforce, followed by sudden downforce loss. In addition, significant hysteresis occurs at $\theta = 24.7°$.



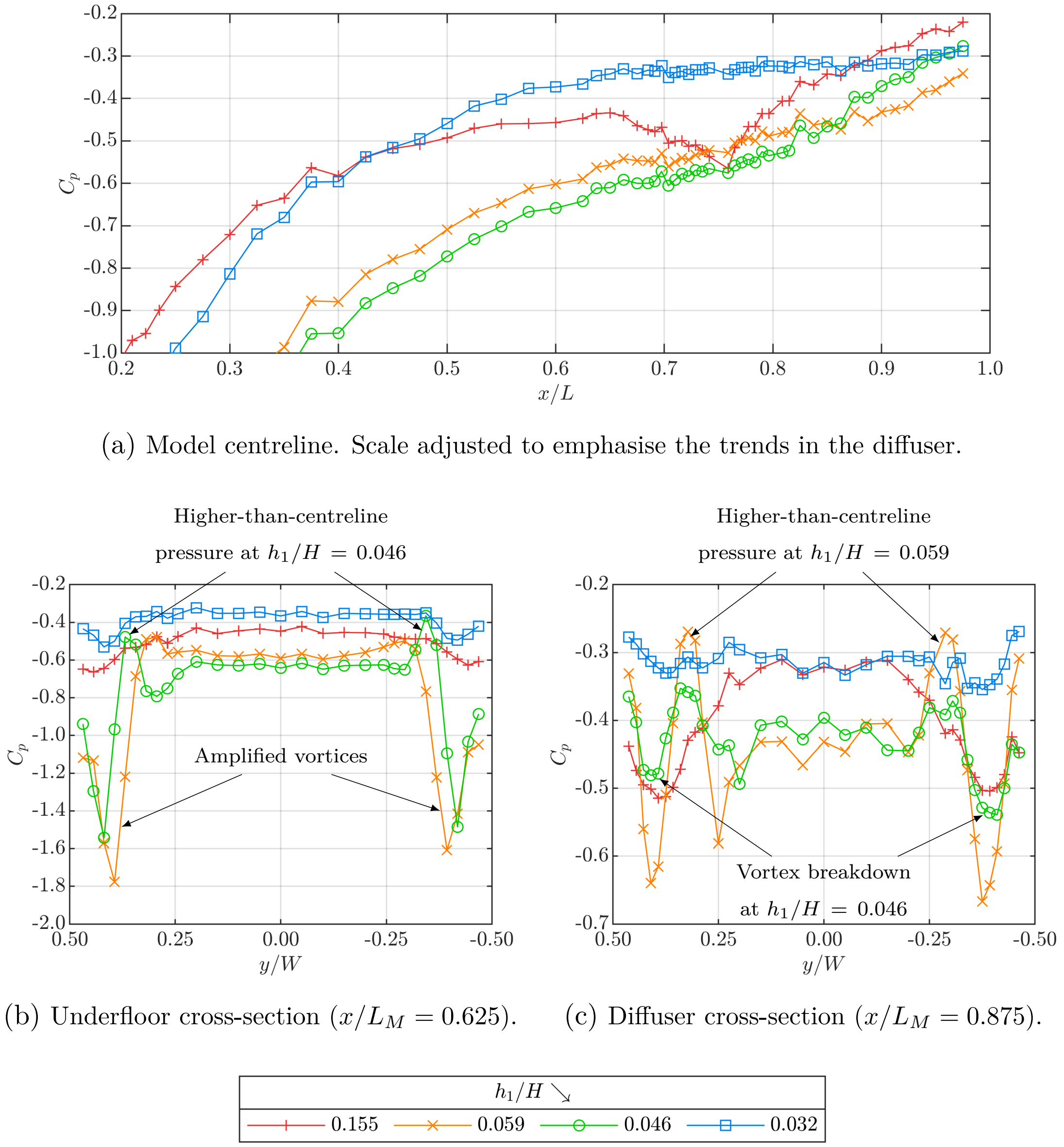
(a) Downforce at $\gamma = 4.8°$.

(b) Difference to $\gamma = 0.0°$.

FIGURE 6.10: Plots of downforce coefficient against non-dimensional ride height, at a 4.8° rake angle and a range of diffuser angles, and the differences to $\gamma = 0.0°$. Vertical dotted lines mark the configurations displayed in Fig. 6.11.

Inspection of pressure distributions on the underfloor and diffuser plates revealed that, at a narrow range of ride heights between $h_1/H = 0.101$ and 0.046, the vortices forming off the underfloor edges are significantly amplified, with the peak vortex-induced C_p on the underfloor surface dropping by up to 0.8, causing increased suction under the entire underfloor. This phenomenon, hitherto referred to as vortex amplification, is illustrated in Fig. 6.11, where the pressure distribution of the 6.2° diffuser at $\gamma = 4.8°$ is presented at four distinctive ride heights. At ride heights above this amplification region (above $h_1/H = 0.046$ to 0.059 at $\theta = 6.2°$, $h_1/H = 0.086$ at $\theta = 11.2°$ and $16.3°$, and $h_1/H = 0.059$ to 0.101 at $\theta = 24.7°$), the vortices weaken significantly by the diffuser mid-point ($x/L_M = 0.875$) for all diffuser angles, with larger width and lower peak suction than at $x/L_M = 0.625$. This process also occurs at $\theta = 0.0°$, suggesting that it is a part of natural vortex evolution at this rake angle, rather than vortex detachment or breakdown due to the sharp diffuser inlet and the following strong adverse pressure gradient. However, at ride heights in the amplification region and at diffuser angles of $\theta \geqslant 11.2°$, the vortices retain more of their strength at $x/L_M = 0.875$. Furthermore, diffusers with angles between 6.2° and 16.3° in the amplification region expand to an exit pressure coefficient ~ 0.1 lower than at other ride heights (see Fig. 6.11a in the $h_1/H = 0.059$ case). At $\theta = 24.7°$ (not shown here), the amplified cases were the only ones to exhibit pressure recovery along the diffuser centreline, as well as significant vortex-induced suction at underfloor sides, which are jointly responsible for the increased downforce seen in Fig. 6.10. The hysteresis in downforce at this diffuser angle is caused by a slight delay of vortex amplification when ride height is increased, as evidenced by pressure distributions at those ride heights. This is analogous to typical diffuser hysteresis with respect to ride height, where the flow reattaches and the vortices reform at a higher ride height when it is being increased.

A unique phenomenon was observed in the amplification region at $\theta = 0.0°$ and $6.2°$, where the vortex-induced suction regions moved outboard and became slightly narrower, accompanied by a narrow region of higher-than-centreline pressure between the vortices and the model centreline. At $h_1/H = 0.059$, this was observed in the diffuser for both of the above-mentioned diffuser angles (see Fig. 6.11c). However, at $h_1/H = 0.046$, this trend was observed at the underfloor in both cases, simultaneously with severe reduction in vortex strength in the diffuser (see Fig. 6.11b and 6.11c). Therefore, the higher-than-centreline pressure regions inboard of the vortices appear to be an indication of vortex breakdown having already occurred further downstream.



(a) Model centreline. Scale adjusted to emphasise the trends in the diffuser.

(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.11: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, at $\gamma = 4.8°$, $\theta = 6.2°$, and at four distinctive ride heights.

The suction peak at the leading edge of the underfloor, not shown here for conciseness, grows at an accelerating rate with decreasing ride height for all diffuser angles, from between -1.83 and -2.12 at $h_1/H = 0.290$, to between -5.14 and -6.27 at $h_1/H = 0.046$. At $h_1/H = 0.032$, it drops sharply to ~ -4.90 at $\theta \leqslant 16.3°$. At $\theta = 11.2°$ and $16.3°$, the rate of suction peak growth slows already at $h_1/H = 0.074$, but the lowest magnitude still occurs at $h_1/H = 0.046$. However, at $\theta = 24.7°$, the lowest C_p peak of -5.42 was measured at $h_1/H = 0.059$, followed by a linear decrease, down to -4.60 at $h_1/H = 0.032$. In all cases where this suction drop was experienced, including where the growth rate slowed at $\theta = 11.2°$ and $16.3°$, a fast pressure rise was observed along the first half of the underfloor centreline, so that the base pressure of the model was reached already upstream of the diffuser inlet, followed by approximately constant C_p along the diffuser centreline. This was accompanied by a reduction in vortex strength at the underfloor ($x/L_M = 0.625$), and their nearly complete disappearance in the diffuser ($x/L_M = 0.875$). These trends, seen in Fig. 6.11 at $h_1/H = 0.032$, are indications of flow separation off the underfloor surface and early vortex breakdown, although the extent and characteristics of this separation, and of possible reattachment, are unknown. Importantly, no significant flow asymmetry was observed for any configuration at $\gamma = 4.8°$.

Figure 6.11, referred to in the discussions above, presents the underbody pressure distributions of the 6.2° diffuser at four ride heights. The centreline plot reveals gradually increasing suction in the front half of the underfloor with reducing ride height, as well as the stall at $h_1/H = 0.032$. The cross-section plots give an insight into vortex evolution, showing the amplification at $h_1/H = 0.059$ and 0.046, the significantly reduced strength in the diffuser, and the high-C_p regions inboard of the vortices in the amplification region ($h_1/H = 0.059$ and 0.046 in Fig. 6.11b and 6.11c). It is noteworthy that the faster streamwise vortex weakening at $h_1/H = 0.046$ compared to $h_1/H = 0.059$ was accompanied by faster pressure recovery along the diffuser centreline, which resulted in a higher C_p at the exit, suggesting that the amplification effect is already not as strong at the lower ride height, despite a further downforce increase due to increased suction at the front of the underfloor.

6.2 Diffuser Aerodynamics with Sliding Sidewalls

The following section presents the results obtained using the sliding-sidewall configuration of the large-scale diffuser model, where no end plates are present along the length of the underfloor, as illustrated in Fig. 6.12. The purpose of these tests, and the main aim of the discussion below, is to quantitatively study the impact of rake on the aerodynamics of a diffuser-equipped bluff body with no underfloor end plates.

Similar to the tests presented in Section 6.1, a single ride height sweep involved reducing the ride height from the maximum of $h_1/H = 0.290$ down to 0.032 and increasing it back to 0.290. The same rake and diffuser angles were used for consistency, i.e. $\gamma = 0.0°$, 1.8°, 4.8°, and $\theta = 0.0°$, 6.2°, 11.2°, 16.3°, 24.7°.

The model geometry used in the ride height sweeps at $\gamma = 0.0°$ closely corresponded to

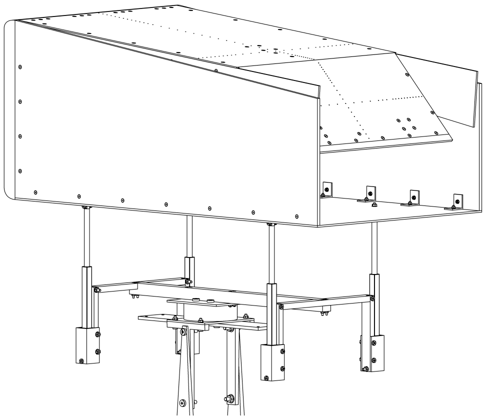
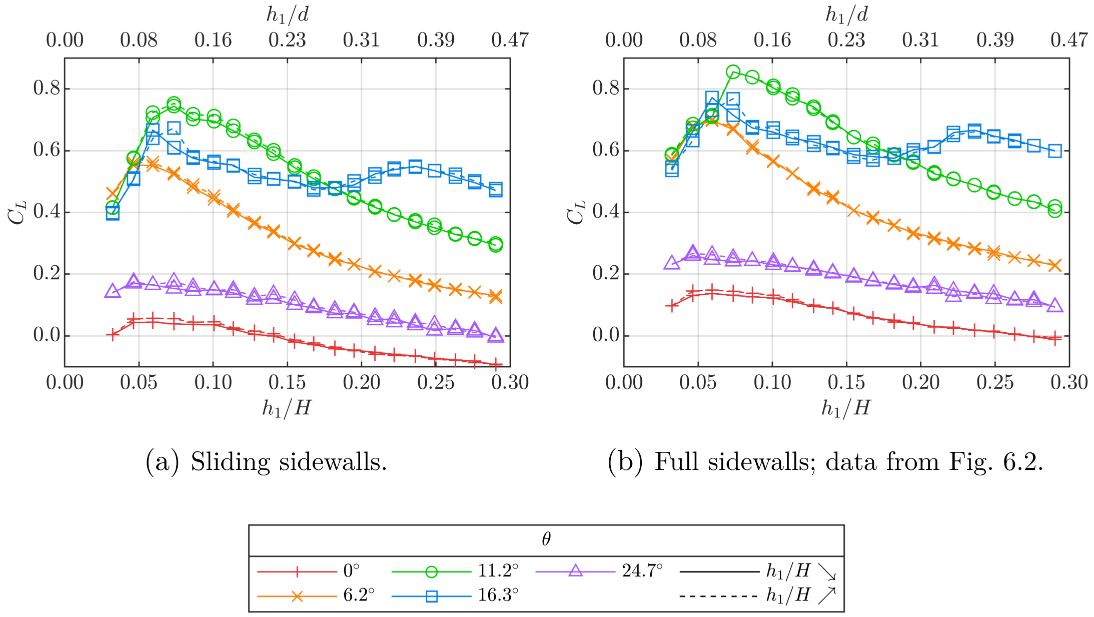


FIGURE 6.12: The sliding-sidewall configuration of the large-scale model.

that of the full-sidewall configuration. There were only two differences in the geometry: the first was the slightly narrower diffuser plate (380 mm, compared to 387 mm when full sidewalls were installed), due to the additional pair of sidewalls between the outer ones and the diffuser plate; the second was the slanted edges of the external sidewalls, as seen in Fig. 6.12. The trends in downforce between the two configurations were similar, but the sliding-sidewall configuration experienced values of downforce coefficient smaller by approx. 0.1 at all ride heights and diffuser angles, as illustrated for comparison in Fig. 6.13. This was not an artefact of non-dimensionalisation, as the same frontal area was used for both sidewall configurations. Moreover, surface pressure measurements on the underbody revealed a similar trend, where the pressure coefficient was higher



(a) Sliding sidewalls.

(b) Full sidewalls; data from Fig. 6.2.

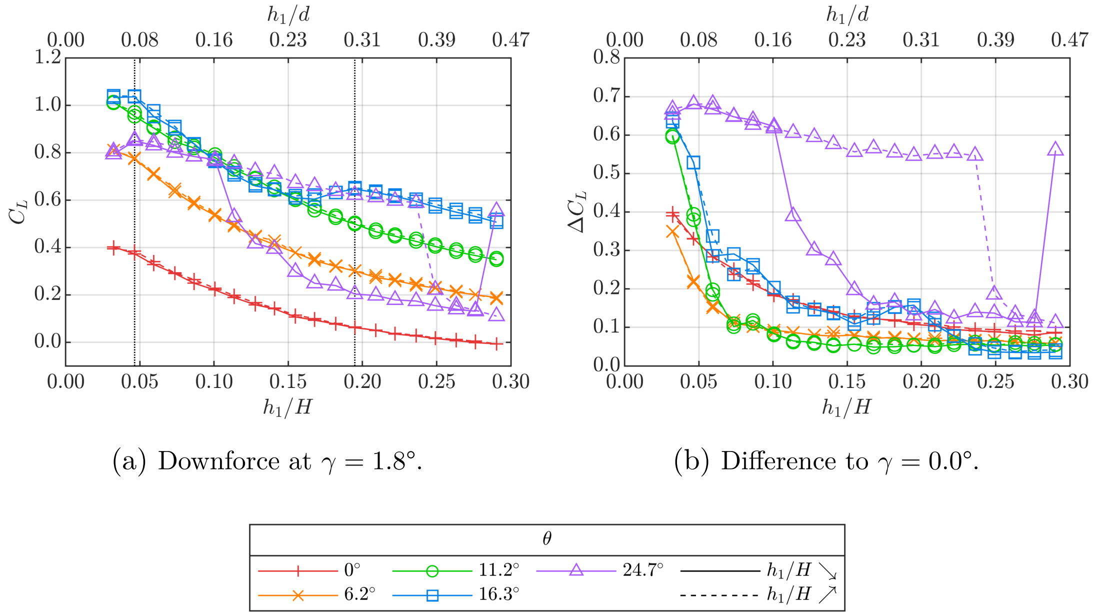
FIGURE 6.13: Plots of downforce coefficient against non-dimensional ride height, for both sidewall configurations, at a 0.0° rake angle and a range of diffuser angles.

by approx. 0.1 at all pressure tap locations, at all ride heights and diffuser angles. The agreement between the force and pressure data suggests that this was an aerodynamic effect, and the presence of this change at $\gamma = 0.0°$ and $\theta = 0.0°$ indicates that the cause was the slanted sidewall edges, rather than the modified diffuser plate width. However, the lack of a survey of the near-wake prevents a more in-depth analysis of this phenomenon.

Crucially, the following discussion on the impact of rake with no underfloor end plates uses the sliding-sidewall configuration (Fig. 6.13a) as the zero-rake reference, rather than the full-sidewall configuration (Fig. 6.13b) discussed in Section 6.1.1.

6.2.1 Moderate Rake Angle

Presented first are the trends in downforce with changing ride height for the moderate rake configuration, i.e. $\gamma = 1.8°$. Figure 6.14 shows C_L plots with non-dimensional ride height, at all the diffuser angles, as well as the difference in downforce compared to $\gamma = 0.0°$ in the same sidewall configuration. The 0.0°, 6.2° and 11.2° diffusers demonstrated similar trends, with a monotonic downforce increase from $h_1/H = 0.290$ until $h_1/H = 0.046$, and a slower increase to the lowest ride height at $\theta = 0.0°$ and 6.2°. Moreover, the gradient $\mathrm{d}C_L/\mathrm{d}h_1$ increased with diffuser angle up to $\theta = 11.2°$. However,



(a) Downforce at $\gamma = 1.8°$.

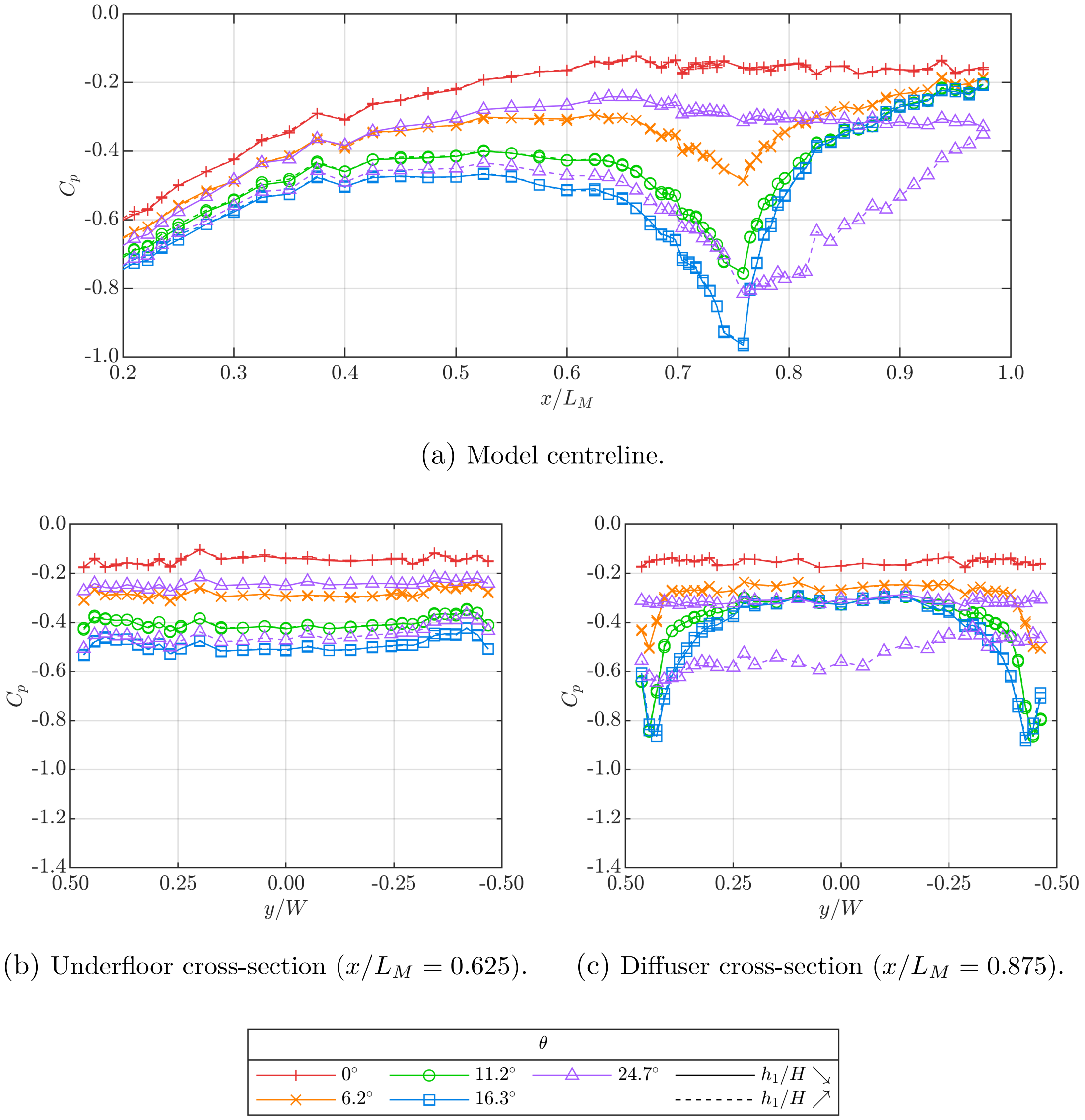
(b) Difference to $\gamma = 0.0°$.

FIGURE 6.14: Plots of downforce coefficient against non-dimensional ride height, for the sliding-sidewall configuration, at a 1.8° rake angle and a range of diffuser angles, and the differences to $\gamma = 0.0°$. Vertical dotted lines mark the ride heights displayed in Fig. 6.15 and 6.16.

the 11.2° diffuser experienced a mild slowdown of downforce growth at $h_1/H = 0.086$, with a total C_L increase of approximately 0.67 between the maximum and minimum ride heights. This is consistent with the findings of Cooper et al. [99], who recorded a C_L increase of 0.70 between similar ride heights, using a model with a 9.17° diffuser at a 1.60° pitch angle. The 16.3° diffuser followed the above-mentioned trends at $h_1/H \geqslant 0.236$, but experienced a subsequent downforce reduction at $0.195 \geqslant h_1/H \geqslant 0.182$, similar to the $\gamma = 0.0°$ configuration with full sidewalls (Fig. 6.2). A further ride height reduction results in a continued downforce increase, until $h_1/H = 0.046$, with a subsequent drop.

Finally, the steepest diffuser of $\theta = 24.7°$ experienced significantly different behaviour. At the maximum ride height, it generated the highest downforce of all the diffusers at $C_L = 0.65$, which dropped to 0.22 just one ride height step later. This was followed by a slow downforce growth at a level similar to $\theta = 6.2°$, but a two-stage C_L increase was recorded between $h_1/H = 0.141$ and 0.101. A downforce plateau and a small downforce reduction were observed at $0.101 \geqslant h_1/H \geqslant 0.046$ and $h_1/H = 0.032$ respectively, both with decreasing and increasing ride height. However, significant hysteresis occurred between $h_1/H = 0.101$ and 0.236, where the diffuser remained in the high-downforce regime much longer when ride height was being increased. The reason behind the switching between the two regimes is not known, but underbody pressure distributions revealed how the underlying flow patterns changed when moderate rake with no end plates was introduced. Pressure profiles at two representative ride heights, $h_1/H = 0.195$ and 0.059, are shown in Fig. 6.15 and 6.16 respectively, with an analysis of the flow patterns presented in the following paragraphs. The leading edge parts of the centreline plots ($x/L_M < 0.2$) were cropped, in order to improve the clarity of the trends occurring in the diffuser.

The centreline C_p plot at $h_1/H = 0.195$ in Fig. 6.15a shows typical profiles for diffusers with $\theta \leqslant 16.3°$, with slow pressure recovery downstream of the underfloor leading edge, pronounced suction peaks at the diffuser inlet, and gradual pressure recovery towards an approximately constant exit pressure $C_{p_{\text{exit}}}$. The magnitude of the suction peak grows with increasing diffuser angle, but the suction is weaker than at the corresponding ride heights at $\gamma = 0.0°$ by up to 0.16. The latter trend might be explained by the increased exit pressure with sliding sidewalls, which is illustrated in Fig. 6.17 and discussed later in this section.



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

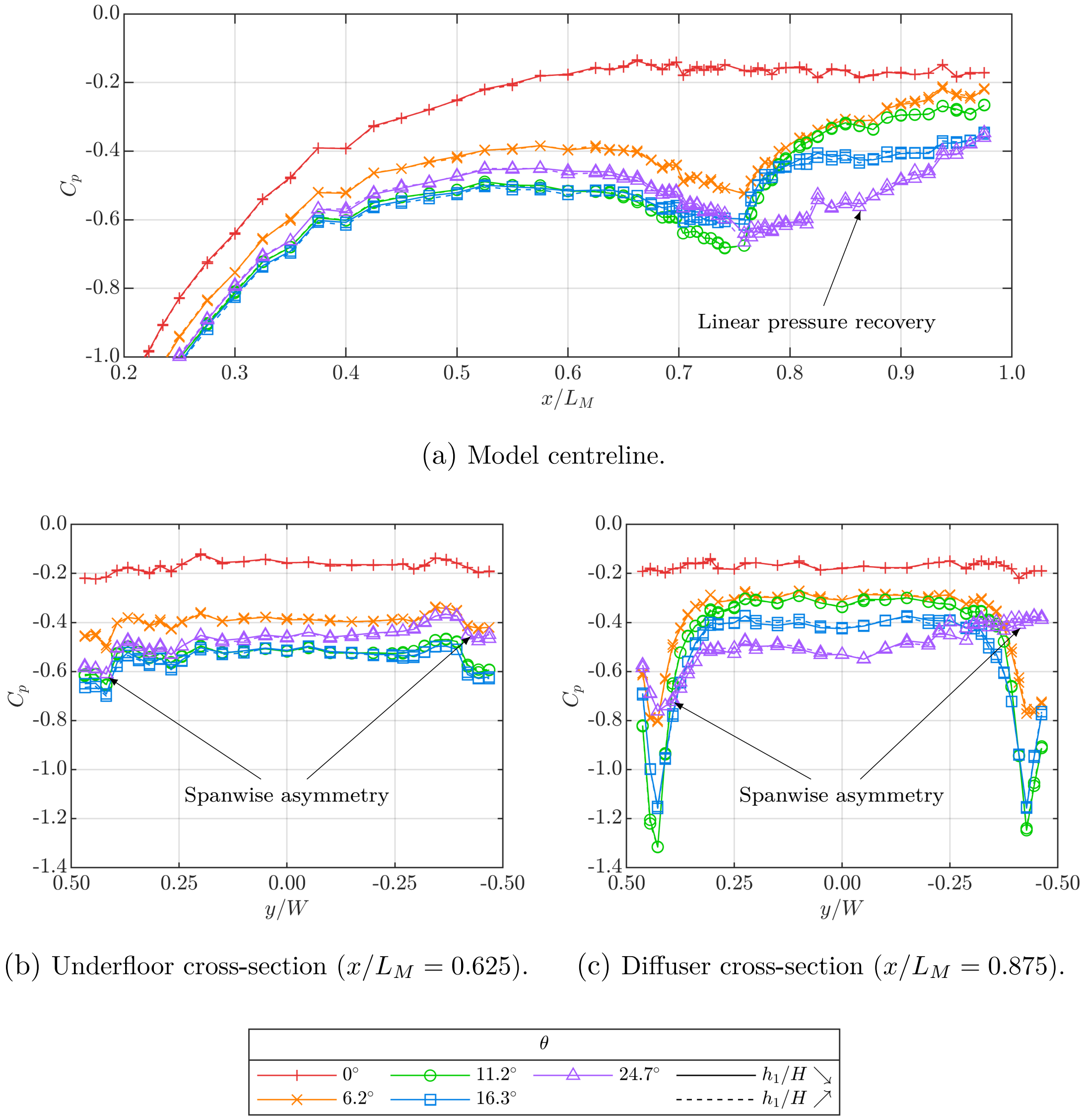
(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.15: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, for the sliding-sidewall configuration, at $h_1/H = 0.195$, $\gamma = 1.8°$, and a range of diffuser angles.

The spanwise profiles across the underfloor (Fig. 6.15b) have similar values at the centreline as at $\gamma = 0.0°$, but suction leakage is largely eliminated. Furthermore, small regions of reduced C_p can be seen near the edges of the underfloor, suggesting that weak vortices have formed along its sides. These reduce flow entrainment into the central part of the underfloor, resulting in smaller spanwise pressure gradients relative to when no vortices are present. Further downstream, the diffuser cross-section (Fig. 6.15c) experiences very strong vortex-induced suction near the end plates. The magnitudes of these suction peaks grow with diffuser angle up until 11.2° and maintain similar strength at 16.3°, with similar C_p values to the $\gamma = 0.0°$ case.

As ride height is reduced below $h_1/H = 0.195$, the majority of the downforce increase for the low-angle diffusers ($\theta \leqslant 16.3°$) is due to an accelerating strengthening of the suction peak under the underfloor leading edge (not shown here), which grew from $-1.78 \leqslant C_p \leqslant -1.68$ at $h_1/H = 0.290$ to $-3.37 \leqslant C_p \leqslant -3.16$ at $h_1/H = 0.032$, its gradient still increasing at the lowest ride height. An additional contribution comes from the edge vortices in the diffuser, which grow in strength and size as ride height is reduced, as seen in Fig. 6.15c and 6.16c.

The 6.2° diffuser experiences a slight reduction in the diffuser inlet suction peak strength at $h_1/H = 0.046$, likely caused by the appearance of a separation bubble on the diffuser



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

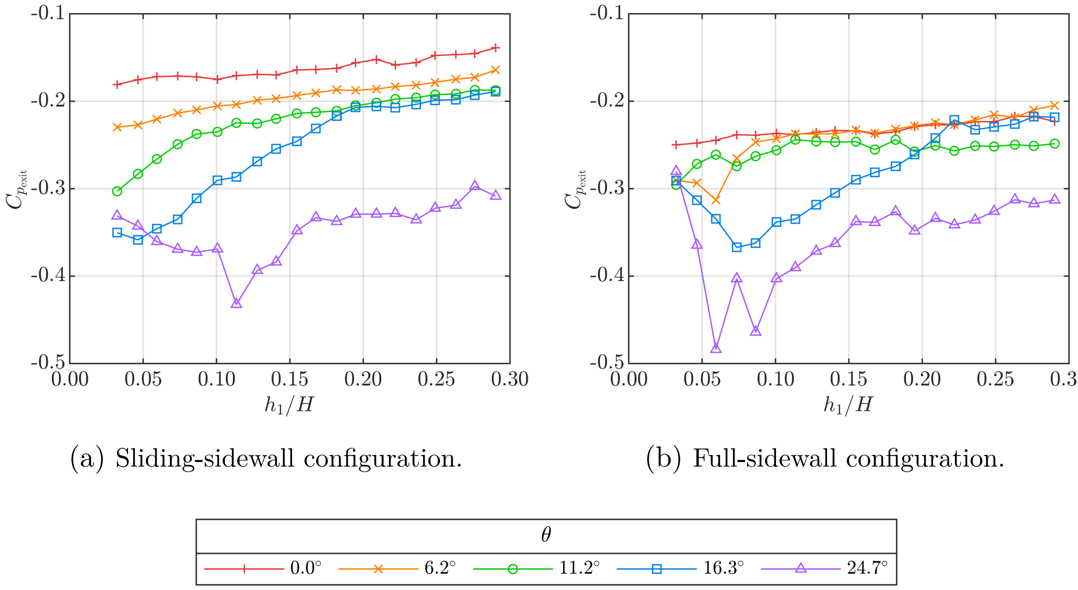
(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE 6.16: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, for the sliding-sidewall configuration, at $h_1/H = 0.046$, $\gamma = 1.8°$, and a range of diffuser angles.

surface, which progresses at $h_1/H = 0.032$ and is responsible for the reduction in downforce growth rate seen in Fig. 6.14a. However, vortex strength at $x/L_M = 0.875$ keeps increasing until $h_1/H = 0.032$. A similar reduction in diffuser inlet peak C_p occurs for the 11.2° diffuser at $h_1/H = 0.101$, which continues as ride height is reduced further, and is accompanied by a reduction in exit pressure $C_{p_{\text{exit}}}$, as seen in Fig. 6.16a. Vortex-induced suction drops marginally at $h_1/H = 0.046$, and down to ~ -1.07 at $h_1/H = 0.032$, which indicates an early stage of vortex breakdown caused by an excessive streamwise adverse pressure gradient. The initial stall occurs much earlier for the 16.3° diffuser, with the diffuser inlet suction peak strength beginning to drop already at $h_1/H = 0.209$. However, the diffuser vortices keep growing in strength until $h_1/H = 0.059$, similar to $\theta = 11.2°$. It is worth noting that the initial separation in the diffuser is once again closely linked to the diffuser area ratio h_3/h_2, which is between 1.76 and 1.93 at $6.2° \leqslant \theta \leqslant 16.3°$ when the diffuser inlet suction peak begins to weaken.

A clear relationship between diffuser angle and exit pressure, or the surface pressure at the trailing edge of the diffuser plate, may be seen in Fig. 6.16a. This relationship was found to change significantly with ride height, and is plotted explicitly in Fig. 6.17a, alongside an analogous plot for the full-sidewall configuration. The figure reveals two clear trends for the sliding-sidewall configuration; firstly, that increasing the diffuser angle causes a direct reduction in the exit pressure of the diffuser, and secondly, that $C_{p_{\text{exit}}}$ drops with decreasing ride height at an accelerating rate. Moreover, the rate of $\mathrm{d}C_{p_{\text{exit}}}/\mathrm{d}h_1$ increases when a separation bubble appears in the diffuser ($h_1/H = 0.087$ at $\theta = 11.2°$ and $h_1/H = 0.182$ at $\theta = 16.3°$). The configuration with full sidewalls, shown in Fig. 6.17b, also experiences a gradual reduction of the exit pressure with progressing flow separation (most notably at $\theta = 16.3°$), but $C_{p_{\text{exit}}}$ is mostly independent of diffuser angle and ride height for attached flows, suggesting that underfloor end plates reduce the sensitivity of the wake to those two degrees of freedom.

The 24.7° diffuser exhibits behaviour vastly different to the shallower diffusers, also depending on the direction of ride height variation, as illustrated in Fig. 6.15. When ride height is decreasing through $h_1/H = 0.195$, the flow is fully separated inside the diffuser, with no streamwise pressure recovery downstream of $x/L_M = 0.7$ and flat spanwise pressure profiles. However, strong linear pressure recovery takes place along the diffuser centreline when ride height is increasing through the same position. The linearity of the pressure profile, atypical of diffuser flows and not previously reported for plane diffusers, suggests that the flow initially separates off the diffuser inlet, and



(a) Sliding-sidewall configuration.

(b) Full-sidewall configuration.

FIGURE 6.17: Plots of exit pressure $C_{p_{\text{exit}}}$ with decreasing ride height ($h_1/H \searrow$) for a range of diffuser angles at $\gamma = 1.8°$.

subsequently curves upwards, towards the trailing edge of the diffuser plate. This inference is corroborated by the results of Huminic and Huminic [125], who simulated the flow around a bluff body with a circular diffuser at $h_1/H = 0.174$, and observed a similar pressure recovery profile to the 24.7° diffuser in Fig. 6.16a, with an inflection point near the diffuser inlet, rather than the sharp peak typical for attached diffuser flows.

The linear pressure recovery along the centreline in the high-downforce regime is accompanied by increased suction across the entire diffuser plate, although with clear spanwise asymmetry. The same flow condition was observed when decreasing the ride height (solid line in Fig. 6.14a) at $h_1/H = 0.290$, as well as at $h_1/H \leqslant 0.101$, as shown in Fig. 6.16. The spanwise asymmetry at $x/L_M = 0.875$ (as seen in Fig. 6.16c) is much more pronounced at low ride heights, where the magnitude of the suction peak suggests the presence of a single coherent vortex in the diffuser. Moreover, a noticeably lower exit pressure was measured for the 24.7° diffuser compared to the other diffuser angles, at all ride heights.

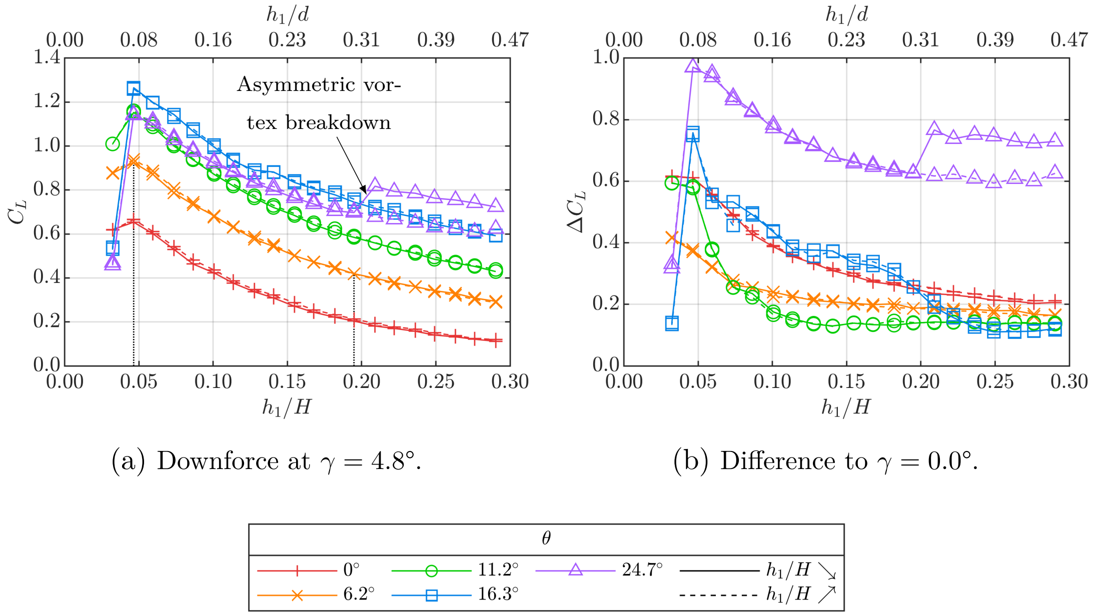
It appears that the bimodal behaviour of the 24.7° diffuser is strongly linked with the geometry of the diffuser volume, as it was only observed at this diffuser angle, only at the 1.8° rake angle, and only without underfloor end plates. It also appears that it does not strictly depend on ride height and that its nature is highly unstable, with the diffuser experiencing different regimes following no clear pattern. A detailed survey of the flow inside the diffuser and in the near-wake, obtained using PIV or another volumetric method, might help to explain this phenomenon in greater detail, but it falls outside of the scope of this work.

The main difference between the trends discussed in the paragraphs above and those obtained using the full-sidewall configuration (see Section 6.1) is the lack of strong vortices near the underfloor edges with no underfloor end plates, and the presence of strong vortex-induced suction regions in the diffuser, which suggests that the vortices formed off the edges of the diffuser inlet. When full sidewalls were installed, suction peaks of up to $C_p = -1.5$ were observed at $x/L_M = 0.625$, with a 0.9 difference between the peaks and the centreline values, compared to the maximum peaks of $C_p = -0.8$ with sliding sidewalls.

Moreover, with the underfloor end plates in place, the appearance of a separation bubble on the diffuser surface was accompanied by a reduction in vortex strength in the diffuser, but it did not influence the flow upstream of the diffuser inlet. However, without the underfloor end plates, vortex breakdown was delayed relative to the flow separation at the centreline. Edge vortices in diffusers had previously been shown to gradually expand and weaken in the streamwise direction [104, 109, 111, 112]. Therefore, vortices that form at the underfloor inlet, as in the case of the full-sidewall configuration, would weaken by the time they reach the diffuser, and therefore be prone to breakdown. On the other hand, vortices that originate at the diffuser inlet would be more concentrated, and therefore more resistant to breakdown, at the same location in the diffuser.

6.2.2 High Rake Angle

Finally, the trends in performance of the high-rake configuration ($\gamma = 4.8°$) with sliding sidewalls are presented in this section. Plots of downforce against ride height for the five previously tested diffuser angles are shown in Fig. 6.18, as well as the differences in downforce relative to the $\gamma = 0.0°$ case. The curves are similar to those obtained at $\gamma = 1.8°$, but downforce is increased at all ride heights, as is the rate of downforce growth $\mathrm{d}C_L/\mathrm{d}h_1$. The 0.0°, 6.2° and 11.2° diffusers demonstrate analogous performance, with an accelerating downforce growth until $h_1/H = 0.046$, and a small drop at $h_1/H = 0.032$. The 16.3° diffuser does not experience an early downforce reduction, in contrast to the $\gamma = 0.0°$ and 1.8° cases, but a small dip in the gradient is still present at $h_1/H = 0.128$. Furthermore, a very sharp downforce reduction takes place at the lowest ride height. The 24.7° diffuser is the only one to experience hysteresis, although



(a) Downforce at $\gamma = 4.8°$.

(b) Difference to $\gamma = 0.0°$.

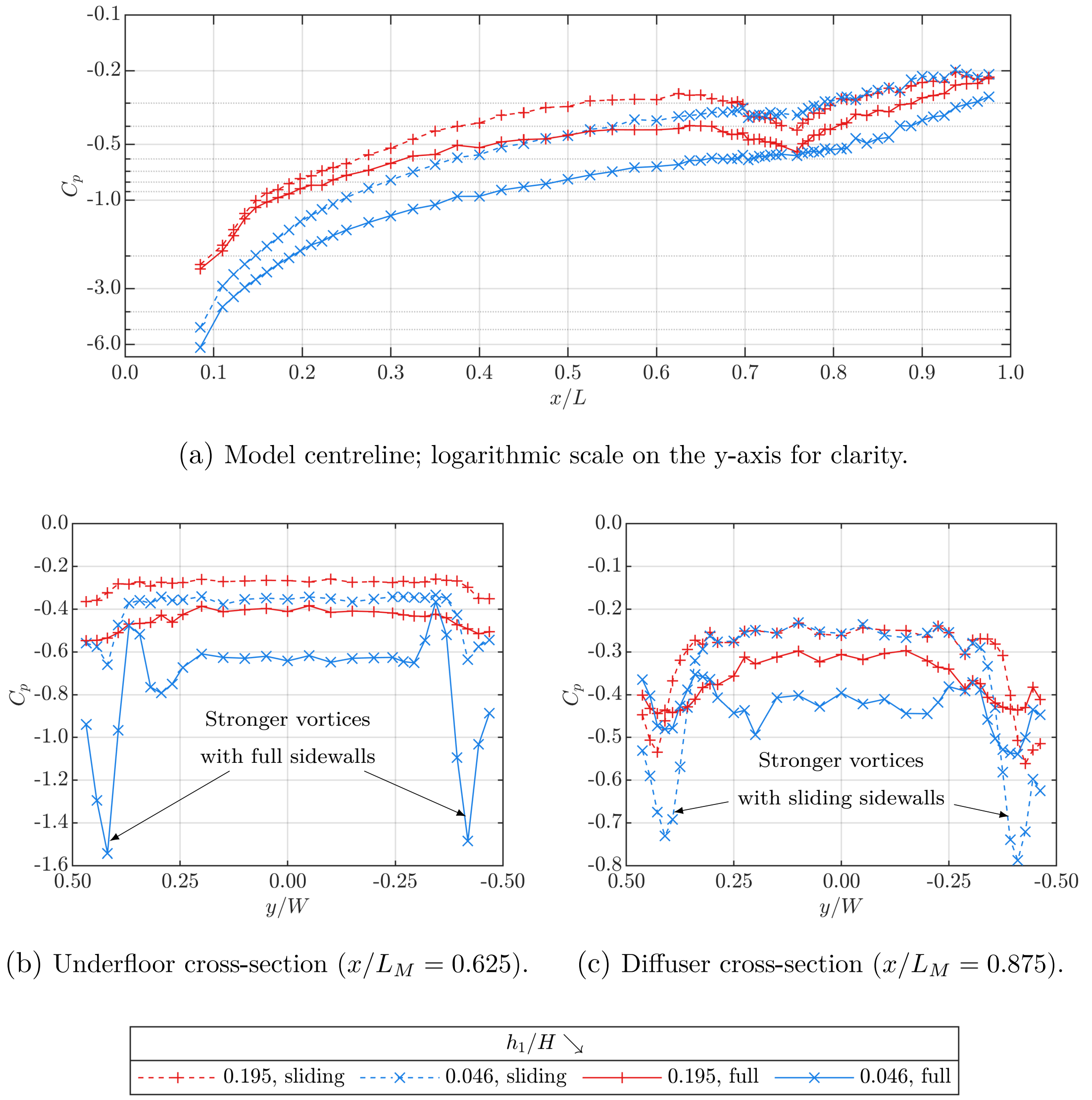
FIGURE 6.18: Plots of downforce coefficient against non-dimensional ride height, for the sliding-sidewall configuration, at a 4.8° rake angle and a range of diffuser angles, and the differences to $\gamma = 0.0°$. Vertical dotted lines mark the ride heights displayed in Fig. 6.19.

the trend is different to $\gamma = 1.8°$. Initially, at $h_1/H = 0.290$, C_L is higher than at $\theta = 16.3°$, and grows slowly with decreasing ride height, but experiences a sudden drop to a level slightly below that of $\theta = 16.3°$ at $h_1/H = 0.195$. Subsequently, downforce increases at a growing rate until a peak of $C_{L_{\max}} = 1.14$ at $h_1/H = 0.046$, and drops to $C_L = 0.46$ at the lowest ride height. When ride height is increased back to the maximum, no sudden change occurs around $h_1/H = 0.2$, and instead the curve follows the level of the 16.3° diffuser.

Pressure profiles on the underbody showed that, similar to $\gamma = 1.8°$, the majority of the downforce increase with decreasing ride height is a result of an accelerating growth of the suction peak at the underfloor leading edge, from $-1.99 \leqslant C_p \leqslant -1.83$ at $h_1/H = 0.290$ down to $-5.21 \leqslant C_p \leqslant -4.61$ at $h_1/H = 0.046$. At the lowest ride height ($h_1/H = 0.032$), the magnitude of the peak drops (becomes less negative) by ~ 0.1 at $\theta \leqslant 11.2°$, and by ~ 0.8 at $\theta \geqslant 16.3°$. Weak pressure recovery is present along the diffuser centreline for all the configurations except $h_1/H = 0.032$ at $\theta \geqslant 16.3°$. The diffuser inlet suction peak achieves its maximum strength at $h_1/H \approx 0.222$ and slowly weakens at lower ride heights for all diffusers up to $\theta = 16.3°$; at $\theta = 24.7°$ it is decreasing already at $h_1/H = 0.290$. However, the variations of C_p in the diffuser with decreasing ride height are small compared to the front half of the underfloor, so downforce growth is not inhibited. Centreline pressure profiles of the 6.2° diffuser at $h_1/H = 0.195$ and 0.046 are shown as dashed lines in Fig. 6.19a, highlighting the difference in suction upstream of the diffuser inlet.

When ride height is reduced to $h_1/H = 0.032$, the remnants of the diffuser inlet suction peak disappear completely for the 6.2° and 11.2° diffusers, but there is still gradual pressure recovery from the underfloor leading edge to the diffuser exit, similar to $h_1/H = 0.046$ (Fig. 6.19a). At $\theta \geqslant 16.3°$, the significant drop in leading edge suction propagates downstream, with significantly higher C_p along the entire underfloor, and approximately constant pressure from $x/L_M = 0.7$.

An additional downforce contribution comes from vortex-induced suction, which grows



(a) Model centreline; logarithmic scale on the y-axis for clarity.

(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

Figure 6.19: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, for the sliding- and full-sidewall configurations, at $\gamma = 4.8°$, $\theta = 6.2°$, and two ride heights.

with decreasing ride height down to $h_1/H = 0.046$ at both $x/L_M = 0.625$ and $x/L_M = 0.875$, at all diffuser angles. The suction peaks upstream of the diffuser inlet are slightly more pronounced than at $\gamma = 1.8°$, and move inboard and increase in magnitude by the time they reach the diffuser mid-point ($x/L_M = 0.875$). In all cases, the vortex-induced suction is greater in the diffuser than in the underfloor, supporting the conclusion that the vortices form at the edges of the diffuser inlet, unlike in the full-sidewall configuration, where the vortices observed in the diffuser had formed upstream of the diffuser inlet, and reduced in strength as they entered the diffuser. This can be clearly seen in Fig. 6.19b and 6.19c, where vortex strength with sliding sidewalls increases with decreasing h_1/H at both streamwise locations, and is greater at $x/L_M = 0.875$ than at $x/L_M = 0.625$ at both ride heights.

Similar to the leading edge suction peak, the behaviour of the vortices at $h_1/H = 0.032$ significantly depends on the diffuser angle. At $\theta \leqslant 11.2°$, their strength reduces by up to 0.15, both at the underfloor and in the diffuser, resulting in small downforce drops. However, at $\theta = 16.3°$ the vortices break down completely, as evidenced by approximately flat spanwise C_p profiles at both $x/L_M = 0.625$ and $x/L_M = 0.875$, likely due to separation phenomena near the underfloor leading edge. The 24.7° diffuser is the only one to experience asymmetric vortex breakdown, already at $h_1/H = 0.195$ when ride height is reduced, which is responsible for the discrete downforce drop seen in Fig. 6.18a. However, both vortices maintain most of their strength at the underfloor ($x/L_M = 0.625$), and the strength of the surviving diffuser vortex keeps increasing until $h_1/H = 0.046$. Similar to the $\theta = 16.3°$ case, reducing the ride height to $h_1/H = 0.032$ causes symmetric vortex breakdown, visible already at $x/L_M = 0.625$. When ride height is increased, vortices re-form on both sides of the underfloor at $h_1/H = 0.046$, closely mirroring the strength from the first part of the sweep. However, only one vortex re-forms in the diffuser, the asymmetric flow condition remaining until the maximum ride height is reached.

The performance trends outlined above differ significantly from those observed for the full-sidewall configuration. To illustrate this, Fig. 6.19 also displays pressure profiles of the 6.2° diffuser with full sidewalls. At the large ride height of $h_1/H = 0.195$ (red lines in the figure), underfloor end plates cause a stronger pressure recovery along the diffuser centreline, resulting in reduced pressure under the entire underfloor. However, vortex strength at the diffuser mid-point is significantly lower, as no new vortices are able to form at the diffuser inlet edges. As ride height is reduced to $h_1/H = 0.046$ (blue lines), the full-sidewall configuration is able to maintain a lager pressure recovery along the underfloor centreline due to the limited flow entrainment, causing greatly reduced C_p under the front half of the model. Further downstream, very strong vortices can be observed upstream of the diffuser inlet, which break down by the time they reach the diffuser mid-point, contrary to the sliding-sidewall configuration, where vortex strength increases from $x/L_M = 0.625$ to 0.875.

6.3 Summary

The current chapter presented the results of an experimental campaign, whose aim was to examine how introducing rake affects the performance trends and flow patterns of an automotive underbody diffuser in ground effect. The experiments were carried out using the large-scale model in a wind tunnel facility equipped with a moving ground. In order to enable investigation of flow patterns, surface pressure measurements were collected on the underfloor and diffuser plates, and two sidewall configurations were tested to examine the impact of rake with and without underfloor end plates. Quasi-static sweeps of ride height, rake and diffuser angles were performed for a wide range of diffuser configurations, the latter two not previously reported in published literature.

Ride height variations at zero rake reproduced trends well-documented in past studies. Lowering the model with no diffuser into stronger ground effect causes it to generate downforce of only $C_L \approx 0.1$, but increasing the diffuser angle significantly amplifies this effect, increasing the downforce coefficient up to a peak of nearly 0.9. This effect is caused by a combination of a strong suction peak at the diffuser inlet, growing with diffuser angle, and symmetrical regions of suction near diffuser end plates induced by a pair of counter-rotating vortices. Increasing the diffuser area ratio, and therefore the adverse pressure gradient inside it, beyond a critical threshold causes the formation of a separation bubble on the diffuser surface, which weakens the inlet suction peak, as well as simultaneous weakening of the edge vortices, both of which contribute to a small downforce drop. This phenomenon was observed for the $\theta = 16.3°$ diffuser at $h_1/H = 0.236$, which nevertheless keeps generating increasing levels of downforce at lower ride heights, thanks to increased suction under the front half of the underfloor. Diffusers between $\theta = 0.0°$ and $16.3°$ reach their critical ride heights between $h_1/H = 0.046$ and 0.074, and subsequently suffer from asymmetric vortex breakdown, where one of the edge vortices collapses completely and the flow separates over a large part of the diffuser plate. The steepest diffuser of $\theta = 24.7°$ is fully separated at all ride heights, with no pressure recovery along the diffuser centreline and no discernible vortices.

Introducing a rake angle of 1.8° with underfloor end plates causes strong downforce enhancement at most ride heights and diffuser angles, up to a peak of $C_L = 1.3$ at $h_1/H = 0.046$ and $\theta = 6.2°$. The majority of this growth is due to increased suction under the front half of the underfloor, caused by pressure recovery along its entire length. An additional contribution is made by the appearance of vortices near the edges of the underfloor, which reduce flow entrainment from the sides of the model and eliminate suction leakage. The strength of the edge vortices at the mid-point of the diffuser was found to be reduced relative to the $\gamma = 0.0°$ configuration, and in most cases the vortices are weaker in the diffuser than upstream of its inlet. This suggests that no new vortices form off the diffuser inlet edges, but instead the vortices that formed off the edges of the underfloor leading edge expand, reduce in strength and shift inboard as they enter the diffuser. Furthermore, no asymmetric vortex breakdown occurs in any of the diffusers at $\gamma = 1.8°$, and the vortices are able to both survive flow separation at $\theta = 16.3°$, and briefly attach to the diffuser surface at $\theta = 24.7°$.

Increasing rake to 4.8° results in a further increase in suction under the front part of the underfloor, with peak C_L reaching up to 2.0. Furthermore, a narrow range of ride heights was observed to significantly amplify the underfloor vortices, their peak C_p jumping by up to 0.7 in one ride height step, reaching maximum suction of $C_p = -1.8$. It is hypothesised that this unusually strong vortex roll-up is induced by a favourable combination of ride height, which allows the right level of flow entrainment, and rake, which allows the right amount of space for the vortices to roll up, but additional tests are required to determine the exact circumstances required to achieve this phenomenon. The ride height sweeps at $\gamma = 4.8°$ also revealed a unique phenomenon not previously documented in automotive diffuser literature, viz. narrow regions of higher-than-centreline C_p just inboard of the vortex-induced suction regions. This feature was observed in the diffuser at $\theta = 0.0°$ and 6.2° at $h_1/H = 0.059$, but when ride height was reduced to 0.046, just prior to downforce loss, a similar pattern was observed upstream of the diffuser inlet, simultaneously with symmetric vortex breakdown in the diffuser. Therefore, this phenomenon appears to take place when vortex breakdown has already occurred further downstream, or when it is imminent. The underlying causes of the higher-than-centreline pressure inboard of vortices near breakdown are not known, and are recommended for future investigations.

Removing underfloor end plates does not significantly alter the mechanisms of downforce generation of a raked body with a diffuser, but several notable differences in flow patterns and performance trends were observed. Firstly, the coefficients of downforce and suction on the underbody are reduced by ~ 0.1 at zero rake. This appears to be caused by increased base pressure of the model, which might be due to the slant of the external sidewalls. Crucially, when the model was raked, only very small and weak vortices were detected upstream of the diffuser inlet in the absence of underfloor end plates, which did not get amplified at any ride height, rake or diffuser angle. In most configurations, the magnitude of vortex-induced suction was greater in the diffuser, suggesting that, even at a high rake angle, the vortices observable in the diffuser form at its inlet. The absence of strong vortices along the edges of the underfloor reduces total downforce in most configurations, but also enables increased flow entrainment, which prevents diffuser stall even at the lowest ride height. Another unique phenomenon was also observed, viz. linear pressure recovery from the diffuser inlet towards the exit in some configurations of the 24.7° diffuser. Surface pressure measurements are not sufficient to fully explain the nature of this anomaly. However, in a fully-attached channel flow, linear pressure recovery could only be achieved using a convex nozzle, i.e. a diverging section whose cross-sectional area increases at an increasing rate in the streamwise direction. This suggests that the flow separates off the sharp inlet edge, and subsequently curves upwards, possibly enclosing a recirculation region beneath the diffuser plate.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

This written work is the culmination of a multi-disciplinary research project centred on the concept of hardware-in-the-loop aerodynamic optimisation, and using the devised systems to investigate the aerodynamics of automotive underbody diffusers. Although the goals of this study have evolved over time, the primary and constant aim was to expand the expertise regarding automated, closed-loop optimisation of aerodynamic bodies, and the outcomes of the project satisfied this in several aspects.

7.1.1 Hardware-in-the-Loop Optimisation

The allure of this type of optimisation stems chiefly from the promising combination of fast shape modifications and near-instantaneous data acquisition, as opposed to resource-intensive CFD simulations. However, the potential for increased optimisation efficiency was not exploited by any of the studies in relevant literature, where issues such as repeatability and precision of shape articulation, and resistance of the geometries to aerodynamic loads dominated the discussions. As part of this project, two independent HIL optimisation systems were devised, constructed and extensively tested, both of them based on a bluff body with an underbody diffuser. Apart from distinctive geometries, the systems utilised different actuation, control and measurement hardware, and were installed in different facilities. However, both were designed with the aim of providing as quick and as reliable real-time shape optimisation as possible.

Conventional mechanical actuation and shape control limited to rigid-body motion resulted in fast, precise and repeatable shape articulation, which together with a fully-integrated control system enabled operationally efficient optimisation. Both systems approached 1000 function evaluations per hour in live wind tunnel tests, and were capable of seemingly indefinite continuous and fully autonomous operation, surpassing previously-demonstrated capabilities unequivocally. Furthermore, despite the rudimentary design of several subsystems, high reliability enabled over 100 hours of real-time optimisation runs and a total of over 100,000 function evaluations across all optimisation and performance tests, validating the appeal of experimental aerodynamic optimisation for industrial problems.

Crucially, reliable convergence was attained in a large proportion of the hardware-in-the-loop optimisation tests. Although the design problem of the automotive underbody diffuser did not exhibit significant local maxima with respect to downforce, numerous discontinuities were present in the search space, in addition to experimental noise and aerodynamic hysteresis. Tests using a range of types and configurations of optimisation algorithms revealed that population-based algorithms, viz. genetic algorithms and particle swarm optimisation, demonstrate the most effective and reliable convergence. However, their performance is sensitive to their settings, with a steady balance between exploration and exploitation, and binary encoding in the case of genetic algorithms, providing the best results. On the other hand, simulated annealing and pattern search were found to be sensitive to experimental noise and starting position, and did not provide adequate reliability for this problem. Furthermore, the immunity of genetic algorithms to noise and hysteresis made it possible to significantly reduce the settling and sampling times, typically conservative in aerodynamic testing, with negligible impact on convergence performance. It was shown that sampling times in the order of 0.01 s or 0.1 s could provide the highest improvements in overall operational efficiency.

7.1.2 Population Sorting

Another advantage of utilising population-based optimisation methods is the ability to implement combinatorial optimisation to sort the successive populations for minimised actuation time. This constitutes a Chebyshev travelling salesman problem, entering a well-established optimisation domain. However, what sets this case apart from traditional problems is its multidimensionality, which is determined by the number of independent actuators, or degrees of freedom, of the system. Simulated tests revealed that increasing dimensionality diminishes the potential benefits of sorting, while increasing the permutation length, or the number of points to sort, has the opposite effect, despite directly corresponding to problem size. Therefore, sorting is most beneficial in HIL optimisation problems with few degrees of freedom and large populations. Secondly, it was recognised that any sorting problem where the cost metric corresponds to time, such as actuation time in the case of HIL optimisation, might benefit from simultaneous selection of the optimal sorting algorithm and sorting time limit. A simple yet effective method for solving this problem was presented, and later validated using wind tunnel-based tests, demonstrating reductions in actuation time of up to 60%, and validating its applicability to real-life problems.

7.1.3 Diffuser Aerodynamics

The immense data collection potential of the experimental systems was utilised to investigate the aerodynamics of the optimised geometries—automotive underbody diffusers—in the presence of rake, or an inclination of the underbody relative to the ground. Tests using both systems in two wind tunnels confirmed that rake results in a significant downforce growth, and surface pressure measurements on the underbody revealed that the majority of this increase is due to additional pressure recovery along the entire length of the body, which strengthens the suction beneath the front half of the model. Moreover, strong counter-rotating vortices were detected upstream of the diffuser inlet when the underfloor with end plates was raked, which not only contributed additional downforce, but also greatly increased the diffuser's resistance to stall, and eliminated asymmetric stall in almost all configurations. Furthermore, the vortices were significantly amplified at a narrow range of ride heights and diffuser angles, leading to even greater levels of downforce. However, these vortices were significantly weaker in the absence of end plates along the length of the model, which resulted in increased flow entrainment and reduced downforce, albeit the additional flow entrainment prevented stall even at the lowest attainable ride height. Finally, a new phenomenon was observed, where in some high diffuser configurations the flow appeared to detach from the diffuser inlet before curving upwards, leading to linear pressure recovery in the diffuser, and unusually high levels of downforce.

The real-time optimisation runs carried out for the purpose of validation of sorting effectiveness were used to identify the configurations of the large-scale diffuser model, in the configuration with underfloor end plates, that produced the most downforce.

Several best configurations in every optimisation run had ride heights between $h_1/H = 0.08$ and 0.09, rake angles between 4.68° and 4.75° (with 4.75° being the physical limit), and diffuser angles between 16.6° and 18.8°. For comparison, the best configurations of the small-scale model with no underfloor end plates, as obtained during the comparative optimisation runs, had ride heights between $h_1/H = 0.07$ and 0.08, rake angles between 4.8° and 5.1°, and diffuser angles between 24.2° and 25.4°. The larger optimal diffuser angle of the small-scale model is likely due to the absence of underfloor end plates, which results in stronger vortices inside the diffuser, as well as the significantly smaller width of the diffuser plate relative to its length, which causes the vortices to occupy a greater proportion of the diffuser width, and therefore to reduce flow separation more effectively.

To summarise, the results presented in this work contribute to improving the understanding of performance trends and flow patterns in diffusers in the presence of rake, while highlighting the crucial impact of the presence and evolution of vortices.

7.2 Future Work Recommendations

Although this work adds a few drops to the sea of knowledge, a few areas from the ocean of unknowns were also brought into light, and this section presents them for the benefit of future researchers.

One key aspect of hardware-in-the-loop aerodynamic optimisation that remained beyond the scope of this project was advanced shape articulation. Industrial optimisation problems frequently involve complex and precise geometry variations, which are easy to recreate using digital models, but challenging to articulate using a remotely-controlled physical body. Design of the actuated geometry likely remains the most significant upfront cost of HIL optimisation, and therefore a crucial obstacle to its application on an industrial scale. Demonstrations of closed-loop optimisation using more advanced geometries, for example incorporating multiple bodies or a combination of rigid body motion and flexible body transformation, would reinforce the viability of this technique to real-life problems.

The detailed investigation of diffuser aerodynamics revealed several phenomena and trends that had not been previously documented. However, force and surface pressure measurements were not sufficient to discern some of the underlying flow patterns, and crucially, to observe the changes in the wake of the model. The first of these phenomena were the regions of higher-than-centreline static pressure just inboard of the vortex-induced suction regions, which appeared to be linked to vortex breakdown having already occurred further downstream. Vortex evolution was previously studied by Mahon et al. [111] using PIV and by Ehirim [116] using CFD, but only in situations where vortices were either present, or broken down. A similar analysis of the configurations highlighted in the discussion in this thesis could shed light onto what mechanism is responsible for the narrow high-pressure regions, and onto the physics of vortex breakdown in those circumstances.

The next novel phenomenon is the linear pressure recovery observed in some diffusers with steep inclinations. The pressure profile suggests that the flow separates off the sharp diffuser inlet edge, before curving upwards, towards the trailing edge of the diffuser plate. This had an immense impact on the performance of the model, generating a strong suction peak at the diffuser inlet, and the lowest pressure coefficient of all the tested configurations at the mid-point of the diffuser plate. The origin and nature of this phenomenon might be revealed through surveys of cross-flow velocities inside the diffuser and in the near-wake, similar to those presented by Ehirim et al. [112, 116].

Finally, the exit pressure of the diffuser was found to vary significantly with model configurations, most notably when no underfloor end plates were present. This is at odds with the traditional assumption that the base pressure of a diffuser-equipped body in ground effect remains approximately constant. The magnitude of these variations was significant, even for attached flows, and the trends with diffuser angle and ride height were consistent, therefore understanding their source in terms of near-wake flow patterns is essential in the endeavour to maximise aerodynamic performance.

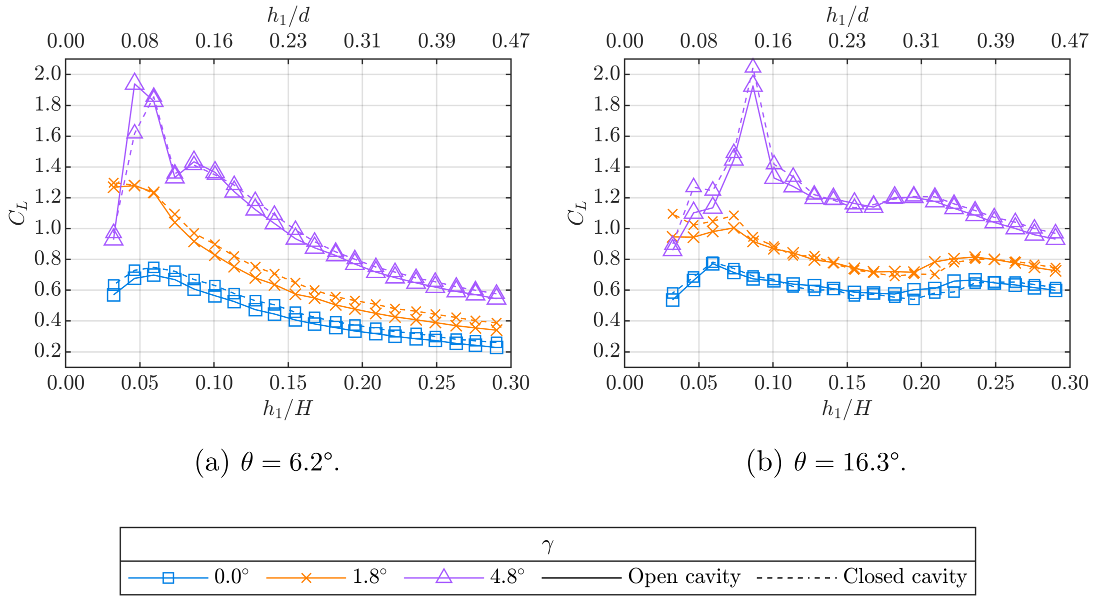
Appendix A

Effect of Closing the Tail Cavity

The results presented in Chapter 6 were obtained using model configurations with an open tail cavity, i.e. with no surface connecting the trailing edges of the diffuser plate and the model roof. In order to quantify the impact of closing the cavity, all ride height, rake and diffuser angle sweeps were repeated with the thin-film system installed on the full-sidewall configuration, as described in detail in Section 3.2.3. However, the main discussion in Chapter 6 was carried out on the open-cavity configuration due to issues with curving and occasional flutter of the film.

A comparison of the two tail configurations with full sidewalls is shown in Fig. A.1, where downforce coefficient is plotted against ride height at several rake and diffuser angles. The plots show very good agreement between the open- and closed-cavity configurations, despite the presence of the thin-film system and the experiments being carried out across several days. Only two diffuser angles are illustrated in the figure for conciseness, but similar agreement was observed for the remaining angles. Across all ride height sweeps, 93.7% of C_L measurements were within 0.1 of the corresponding open-tail results. Among those, 74.3% saw an increase, and the remaining 19.4% a decrease in downforce. Although a large number of these small variations fall within measurement uncertainty, the repeatability of the trends suggests physical variations, which were identified in the pressure distributions on the underbody.

Firstly, at $\theta = 0.0°$ and at all rake angles, the static pressure coefficient at the rear-most centreline tap, $C_{p_{\text{exit}}}$, is between 0.05 and 0.1 lower with the film installed, with suction under the entire model increased by a similar amount. At $\theta = 0.0°$ there is close to zero pressure recovery between $x/L_M = 0.7$ and 1.0 in most configurations, and at $\theta = 0.0°$



(a) $\theta = 6.2°$.

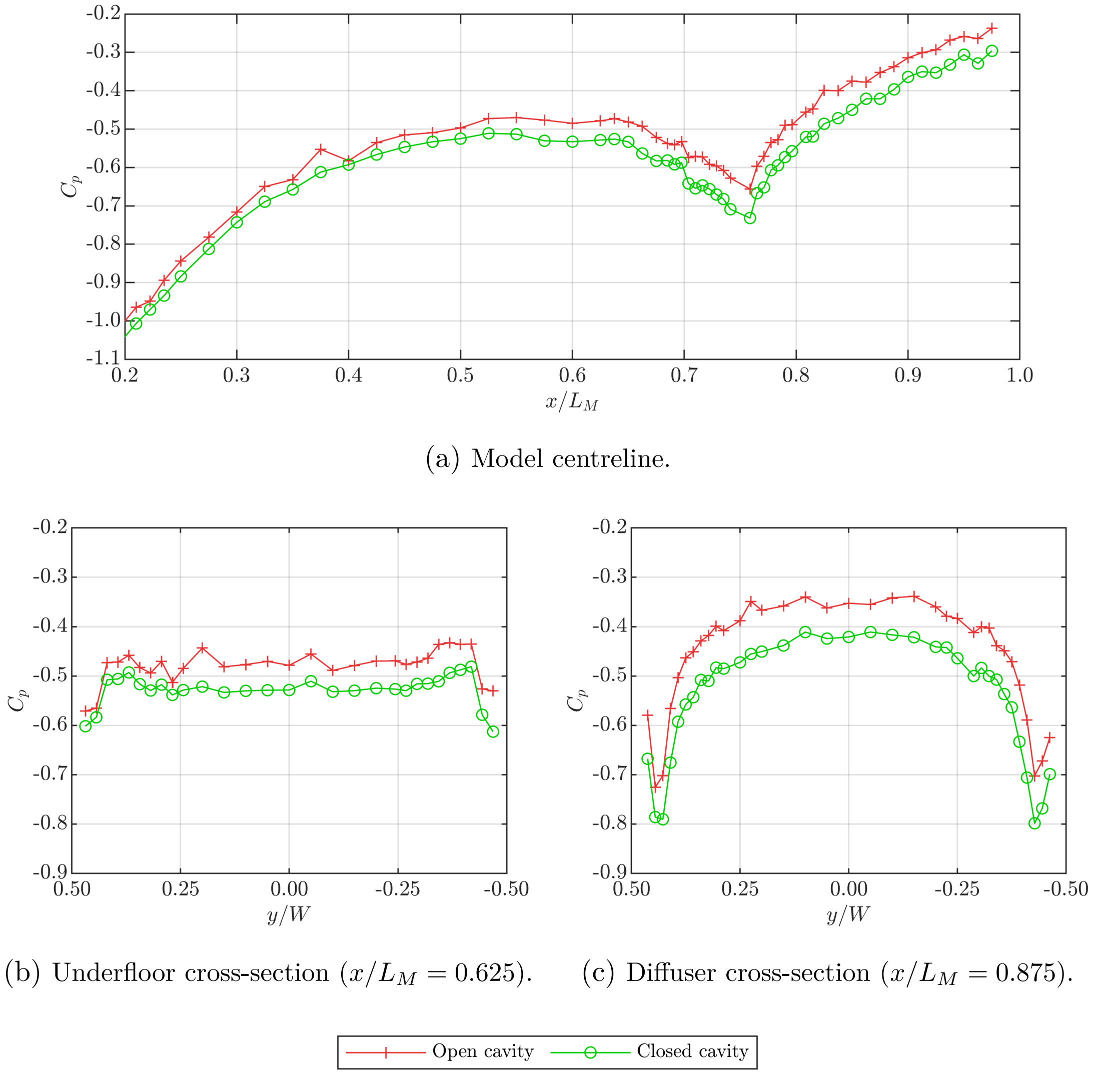
(b) $\theta = 16.3°$.

Figure A.1: Trends of downforce coefficient with ride height, for the open- and closed-cavity configurations with full sidewalls, at two diffuser angles and three rake angles.

and $\gamma = 0.0°$ the film does not increase the length of the diffuser. This suggests that, with the film in place, the base pressure of the model with no diffuser is consistently reduced, even in the presence of rake.

A similar trend was observed at $\theta = 6.2°$, although the magnitude of the $C_{p_{\text{exit}}}$ drop is smaller than at $\theta = 0.0°$, and in some cases negligible. A representative pressure distribution is shown in Fig. A.2, where a clear reduction in exit pressure may be observed with the closed cavity, and this reduced pressure can be seen to propagate both upstream, and across the width of the model. This trend might be caused by a combination of reduced base pressure and extended pressure recovery due to the film increasing the effective diffuser length by up to 6 mm at $\gamma = 4.8°$ and $\theta = 6.2°$, as the diffuser plane is extended to coincide with the trailing edges of the sidewalls, as illustrated in Fig. A.3. However, some configurations of the 11.2° and 16.3° diffusers demonstrated the opposite trend, with slightly increased (less negative) $C_{p_{\text{exit}}}$ and reduced pressure recovery along the diffuser centreline. The inconsistency between the trends at $6.2° \leqslant \theta \leqslant 16.3°$ implies that they may have been caused by random variations of model orientation or diffuser angle, rather than by the closing of the cavity.

The second consistent trend was observed in fully separated configurations, i.e. at most ride heights and rake angles at $\theta = 24.7°$, and in some instances of shallower diffusers at low ride heights and high rake angles. These configurations tend to exhibit a slightly



(a) Model centreline.

(b) Underfloor cross-section ($x/L_M = 0.625$).

(c) Diffuser cross-section ($x/L_M = 0.875$).

FIGURE A.2: Plots of static pressure coefficient along the centreline and across the underfloor and the diffuser, for the open- and closed-tail configurations, at $h_1/H = 0.114$, $\gamma = 1.8°$ and $\theta = 6.2°$.

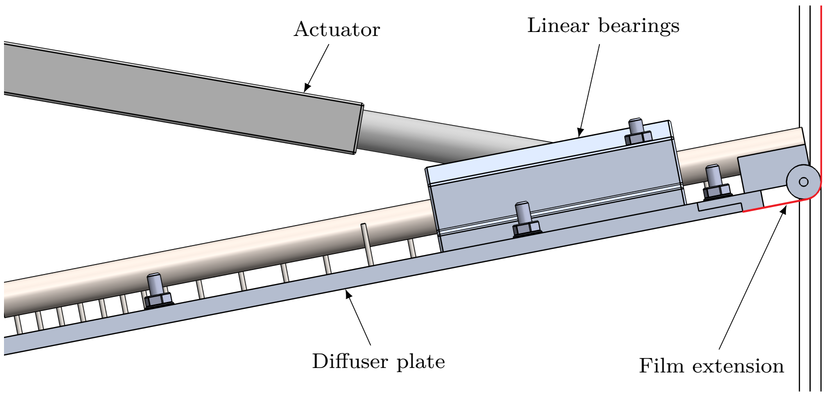


FIGURE A.3: Detail of the extension of the diffuser plate ($\gamma = 4.8°$, $\theta = 6.2°$).

negative C_p gradient from the diffuser inlet to the exit with the open tail cavity, as seen in Fig. 6.9a for the 24.7° diffuser. For all of the fully-separated configurations with an open tail cavity, the change between C_p at the diffuser inlet (averaged across the two taps on both sides of the inlet edge) and $C_{p_{\text{exit}}}$ is -0.0107. With the cavity closed, this negative gradient is eliminated in most cases, with an average change in C_p of $+0.0217$. Although the magnitudes of these gradients are of a similar order of magnitude to the uncertainty of the pressure measurements, the consistency of the gradient change across all separated configurations, at a wide range of ride heights and rake and diffuser angles, suggests that the recirculation region in the immediate wake of a separated diffuser is consistently affected by the closure of the cavity.

The remaining 6.3% of ride height sweep measurements, which comprised differences in C_L of up to 0.7, were found to exhibit small shifts in the occurrence of discrete events, such as a slightly narrower vortex amplification region, downforce loss at low ride heights occurring one ride height step earlier or later, or a slightly delayed formation of a separation bubble. There appears to be no commonality between these shifts, and no clear connection to the variations in underbody C_p of other configurations, suggesting that these differences are caused by random model, setup or flow variations.

The rake and diffuser angle sweeps were also repeated with the thin-film system installed, but the results revealed no significant trends other than what has been discussed in the preceding paragraphs, with similar agreement in downforce measurements to the ride height sweeps.

To summarise, closing the tail cavity of the model had little impact on its performance, confirming that studies using closed- and open-tail models are mutually comparable. The most notable effects of closing the cavity were slightly reduced exit pressures of the $\theta = 0.0°$ and 6.2° diffusers, and slightly modified centreline pressure profiles of the stalled configurations, but the differences in downforce coefficient and underbody surface pressure coefficient between the two configurations did not exceed 0.1 in the majority of cases.

References

[1] J. R. R. A. Martins, “Wing design via numerical optimization,” *SIAG/OPT Views and News*, vol. 23, no. 2, pp. 3–9, 2014.

[2] J. Sobieszczanski-Sobieski and R. T. Haftka, “Multidisciplinary aerospace design optimization: Survey of recent developments,” *Structural optimization*, vol. 14, pp. 1–23, 1997.

[3] E. S. Levinsky and R. L. Palko, “Tests of an improved, computer-controlled, self-optimizing, variable-geometry wing,” in *12th Aerodynamic Testing Conference*, Williamsburg, VA, USA, pp. 215–223, 1982. AIAA 82-599.

[4] E. S. Levinsky and R. L. Palko, “Semispan wind tunnel test of a computer-controlled self-optimising flexible technology wing,” in *10th Aerodynamic Testing Conference*, San Diego, CA, USA, pp. 123–135, 1978. AIAA 78-786.

[5] E. S. Levinsky and R. L. Palko, “Supercritical tests of a self-optimising variable-camber wind tunnel model,” in *Advanced Technology Airfoil Research, Volume 1*, Hampton, VA, USA, pp. 297–313, NASA, 1979. N79-20048.

[6] J. L. Rioual, P. A. Nelson, and M. J. Fisher, “Experiments on the automatic control of boundary-layer transition,” *Journal of Aircraft*, vol. 31, no. 6, pp. 1416–1418, 1994.

[7] R. Hunt, G. S. Hornby, and J. D. Lohn, “Toward evolved flight,” in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, Washington, DC, USA, pp. 957–964, ACM, 2005.

[8] F. Boria, B. Stanford, S. Bowman, and P. Ifju, “Evolutionary optimization of a morphing wing with wind-tunnel hardware in the loop,” *AIAA Journal*, vol. 47, no. 2, pp. 399–409, 2009.

[9] R. Cosin, M. V. Angelo, F. M. Catalano, and F. T. Bonemer De Salvi, “Mission adaptive wing optimization with wind tunnel hardware in the loop,” in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, USA, 2010. AIAA 2010-9201.

[10] A. V. Popov, L. T. Grigorie, R. Botez, M. Mamou, and Y. Mébarki, “Real time morphing wing optimization validation using wind-tunnel tests,” *Journal of Aircraft*, vol. 47, no. 4, pp. 1346–1355, 2010.

[11] A. V. Popov, L. T. Grigorie, R. Botez, M. Mamou, and Y. Mébarki, “Closed-loop control validation of a morphing wing using wind tunnel tests,” *Journal of Aircraft*, vol. 47, no. 4, pp. 1309–1317, 2010.

[12] D. Coutu, V. Brailovski, P. Terriault, M. Mamou, and Y. Mébarki, “Aerostructural model for morphing laminar wing optimization in a wind tunnel,” *Journal of Aircraft*, vol. 48, no. 1, pp. 66–76, 2011.

[13] B. Perseghetti, J. Goppert, S. Yantek, E. Matson, I. Hwang, and J. Gallagher, “Optimizing energy efficiency of a flapping robotic bird through application of evolutionary algorithms,” in *AIAA Infotech @ Aerospace*, Kissimmee, FL, USA, 2015. AIAA 2015-1456.

[14] T. A. Weisshaar, “Morphing aircraft technology – new shapes for aircraft design,” in *RTO-MP-AVT-141 - Multifunctional Structures/Integration of Sensors and Antennas*, Neuilly-sur-Seine, France, 2006.

[15] Z. Hui, Y. Zhang, and G. Chen, “Aerodynamic performance investigation on a morphing unmanned aerial vehicle with bio-inspired discrete wing structures,” *Aerospace Science and Technology*, vol. 95, no. 105419, 2019.

[16] D. A. Neal, J. Farmer, and D. Inman, “Development of a morphing aircraft model for wind tunnel experimentation,” in *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Newport, RI, USA, 2006. AIAA 2006-2141.

[17] A. Moosavian, F. Xi, and S. M. Hashemi, “Design and motion control of fully variable morphing wings,” *Journal of Aircraft*, vol. 50, no. 4, pp. 1189–1201, 2013.

[18] E. A. Bubert, B. K. S. Woods, K. Lee, C. S. Kothera, and N. M. Wereley, “Design

and fabrication of a passive 1D morphing aircraft skin," *Journal of Intelligent Material Systems and Structures*, vol. 21, no. 17, pp. 1699–1717, 2010.

[19] M. Maki and D. Hirabayashi, "Experimental study of a morphing wing configuration with multi-slotted variable-camber mechanism," in *AIAA Atmospheric Flight Mechanics Conference*, Washington, DC, USA, 2016. AIAA 2016-3849.

[20] S. Kota, J. Hetrick, R. Osborn, D. Paul, E. Pendleton, P. Flick, and C. Tilmann, "Design and application of compliant mechanisms for morphing aircraft structures," in *Smart Structures and Materials 2003: Industrial and Commercial Applications of Smart Structures Technologies*, San Diego, CA, USA, pp. 24–33, 2003.

[21] S. Barbarino, E. I. Saavedra Flores, R. M. Ajaj, I. Dayyani, and M. I. Friswell, "A review on shape memory alloys with applications to morphing aircraft," *Smart Materials and Structures*, vol. 23, no. 6: 063001, 2014.

[22] B. K. S. Woods, O. Bilgen, and M. I. Friswell, "Wind tunnel testing of the fish bone active camber morphing concept," *Journal of Intelligent Material Systems and Structures*, vol. 25, no. 7, pp. 772–785, 2014.

[23] G. Molinari, M. Quack, A. F. Arrieta, M. Morari, and P. Ermanni, "Design, realization and structural testing of a compliant adaptable wing," *Smart Materials and Structures*, vol. 24, no. 10: 105027, 2015.

[24] F. Previtali, G. Molinari, A. F. Arrieta, M. Guillaume, and P. Ermanni, "Design and experimental characterisation of a morphing wing with enhanced corrugated skin," *Journal of Intelligent Material Systems and Structures*, vol. 27, no. 2, pp. 278–292, 2016.

[25] O. Şugar Gabor, A. Koreanschi, R. M. Botez, M. Mamou, and Y. Mebarki, "Numerical simulation and wind tunnel tests investigation and validation of a morphing wing-tip demonstrator aerodynamic performance," *Aerospace Science and Technology*, vol. 53, pp. 136–153, 2016.

[26] A. Emiliavaca, C. J. de Araújo, C. R. Souto, and A. Ries, "Characterization of shape memory alloy micro-springs for application in morphing wings," *Smart Materials and Structures*, vol. 28, no. 1: 015010, 2018.

[27] O. Bilgen, M. I. Friswell, K. B. Kochersberger, and D. J. Inman, "Surface actu-

ated variable-camber and variable-twist morphing wings using piezocomposites," in *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Denver, CO, USA, 2011. AIAA 2011-2072.

[28] M. Abdulrahim, H. Garcia, and R. Lind, "Flight characteristics of shaping the membrane wing of a micro air vehicle," *Journal of Aircraft*, vol. 42, no. 1, pp. 131–137, 2005.

[29] H. Rodrigue, S. Cho, M.-W. Han, B. Bhandari, J.-E. Shim, and S.-H. Ahn, "Effect of twist morphing wing segment on aerodynamic performance of UAV," *Journal of Mechanical Science and Technology*, vol. 30, no. 1, pp. 229–236, 2016.

[30] D. A. Perkins, J. L. Reed, and E. Havens, "Morphing wing structures for loitering air vehicles," in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, CA, USA, 2004. AIAA 2004-1888.

[31] J. L. Pinkerton and R. W. Moses, "A feasibility study to control airfoil shape using THUNDER," NASA Langley Research Center, 1997. NASA Technical Memorandum 4767.

[32] V. Giurgiutiu, "Recent advances in smart-material rotor control actuation," in *41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, Atlanta, GA, US, 2000. AIAA-2000-1709.

[33] T. D. Usher, K. R. Ulibarri, and G. S. Camargo, "Piezoelectric microfiber composite actuators for morphing wings," *ISRN Materials Science*, vol. 2013: 189659, 2013.

[34] M. R. Angelino and G. N. Washington, "Design and construction of a piezoelectric point actuated active aperture antenna," *Journal of Intelligent Material Systems and Structures*, vol. 13, pp. 125–136, 2002.

[35] A. C. Henry, G. Molinari, J. R. Rivas-Padilla, and A. F. Arrieta, "Smart morphing wing: Optimization of distributed piezoelectric actuation," *AIAA Journal*, vol. 57, no. 6, pp. 2384–2393, 2019.

[36] L. Ionov, "Soft microorigami: self-folding polymer films," *Soft Matter*, vol. 7, no. 15, pp. 6786–6791, 2011.

[37] K. Kalaitzidou and A. J. Crosby, "Adaptive polymer particles," *Applied Physics Letters*, vol. 93, no. 4: 041910, 2008.

[38] V. Stroganov, S. Zakharchenko, E. Sperling, A. K. Meyer, O. G. Schmidt, and L. Ionov, "Biodegradable self-folding polymer films with controlled thermo-triggered folding," *Advanced Functional Materials*, vol. 24, no. 27, pp. 4357–4363, 2014.

[39] A. Agrawal, T. Yun, S. L. Pesek, W. G. Chapman, and R. Verduzco, "Shape-responsive liquid crystal elastomer bilayers," *Soft Matter*, vol. 10, pp. 1411–1415, 2014.

[40] E. Smela, O. Inganäs, and I. Lundström, "Controlled folding of micrometer-size structures," *Science*, vol. 268, no. 5218, pp. 1735–1738, 1995.

[41] B. Simpson, G. Nunnery, R. Tannenbaum, and K. Kalaitzidou, "Capture/release ability of thermo-responsive polymer particles," *Journal of Materials Chemistry*, vol. 20, no. 17, pp. 3496–3501, 2010.

[42] H.-W. Huang, A. J. Petruska, M. S. Sakar, M. Skoura, F. Ullrich, Q. Zhang, S. Pané, and B. J. Nelson, "Self-folding hydrogel bilayer for enhanced drug loading, encapsulation, and transport," in *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Orlando, FL, USA, pp. 2103–2106, 2016.

[43] T. Deng, C. Yoon, Q. Jin, M. Li, Z. Liu, and D. H. Gracias, "Self-folding graphene-polymer bilayers," *Applied Physics Letters*, vol. 106, no. 20: 203108, 2015.

[44] A. W. Feinberg, A. Feigel, S. S. Shevkoplyas, S. Sheehy, G. M. Whitesides, and K. K. Parker, "Muscular thin films for building actuators and powering devices," *Science*, vol. 317, no. 5843, pp. 1366–1370, 2007.

[45] A. Agrawal, P. Luchette, P. Palffy-Muhoray, S. L. Biswal, W. G. Chapman, and R. Verduzco, "Surface wrinkling in liquid crystal elastomers," *Soft Matter*, vol. 8, pp. 7138–7142, 2012.

[46] J. D. W. Madden, B. Schmid, M. Hechinger, S. R. Lafontaine, P. G. A. Madden, F. S. Hover, R. Kimball, and I. W. Hunter, "Application of polypyrrole actuators: Feasibility of variable camber foils," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp. 738–749, 2004.

[47] The MathWorks, Inc., "Global Optimization Toolbox - MATLAB," 2017. Available: https://www.mathworks.com/products/global-optimization.html [Accessed: 05 June 2017].

[48] S. S. Rao, *Engineering Optimization: Theory and Practice.* John Wiley & Sons, 4 ed., 2009.

[49] R. M. Paiva, A. Carvalho, C. Crawford, L. Felix, A. A. Gomes, and A. Suleman, "A robust and reliability based design optimization framework for wing design," in *2nd International Conference on Engineering Optimization*, Lisbon, Portugal, 2010.

[50] R. Hooke and T. A. Jeeves, "'Direct search' solution of numerical and statistical problems," *Journal of the ACM*, vol. 8, no. 2, pp. 212–229, 1961.

[51] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 1–25, 1997.

[52] R. M. Lewis and V. Torczon, "Pattern search algorithms for bound constrained minimization," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1082–1099, 1999.

[53] A. Sóbester, A. I. J. Forrester, D. J. J. Toal, E. Tresidder, and S. Tucker, "Engineering design applications of surrogate-assisted optimization techniques," *Optimization and Engineering*, vol. 15, pp. 243–265, 2012.

[54] A. I. J. Forrester, N. W. Bressloff, and A. J. Keane, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 462, no. 2071, pp. 2177–2204, 2006.

[55] M. Mitchell, *An Introduction to Genetic Algorithms.* The MIT Press, 1 ed., 1996.

[56] C. Xiu and L. Lu, "Parallel chaos optimization algorithm," in *2010 International Conference on Electrical and Control Engineering*, Wuhan, China, pp. 1463–1465, 2010.

[57] Z. Jian and W. Jing, "A new parallel chaos optimization algorithm with the number of variables reduced," in *2010 International Conference on Computer Application and System Modeling*, Taiyuan, China, pp. V5–446–V5–449, IEEE, 2010.

[58] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the*

1995 IEEE International Conference on Neural Networks, Perth, WA, Australia, pp. 1942–1948, 1995.

[59] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[60] N. M. Sabri, M. Puteh, and M. R. Mahmood, “A review of gravitational search algorithm,” *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. 3, pp. 1–39, 2013.

[61] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[62] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of Statistical Physics*, vol. 34, no. 5/6, pp. 975–986, 1984.

[63] O. Tekinalp and M. Bingol, “Simulated annealing for missile optimization: Developing method and formulation techniques,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 4, pp. 616–626, 2004.

[64] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers.* World Scientific Publishing, 1 ed., 1999.

[65] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks.* Springer, 1 ed., 2000.

[66] P. Biswas, “Particle Swarm Optimization.” Available: `https://www.mathworks.com/matlabcentral/fileexchange/43541-particle-swarm-optimization-pso` [Accessed: 11 May 2017].

[67] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39–43, IEEE, 1995.

[68] R. C. Eberhart and Y. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” in *Proceedings of the 2000 Congress on Evolutionary Computation*, La Jolla, CA, USA, pp. 84–88, IEEE, 2000.

[69] M. D. McKay, R. J. Beckman, and W. J. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[70] I. M. Sobol', "On the systematic search in a hypercube," *SIAM Journal on Numerical Analysis*, vol. 16, no. 5, pp. 790–793, 1979.

[71] D. E. Goldberg, K. Deb, and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, no. 4, pp. 333–362, 1992.

[72] C. R. Reeves and J. E. Rowe, *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, 1 ed., 2002.

[73] N. M. Razali and J. Geraghty, "Genetic algorithm performance with different selection strategies in solving TSP," in *Proceedings of the World Congress on Engineering 2011 Vol II*, London, UK, 2011.

[74] P. V. Raja and V. M. Bhaskaran, "Improving the performance of genetic algorithm by reducing the population size," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 8, pp. 86–91, 2013.

[75] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[76] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, USA, pp. 1945–1950, IEEE, 1999.

[77] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings*, Anchorage, AK, USA, pp. 69–73, 1998.

[78] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, San Diego, CA, USA, pp. 591–600, 1998.

[79] R. M. Lewis and V. Torczon, "Pattern search methods for linearly constrained minimization," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 917–941, 2000.

[80] C. Bogani, M. G. Gasparo, and A. Papini, "Generalized pattern search methods for a class of nonsmooth optimization problems with structure," *Journal of Computational and Applied Mathematics*, vol. 229, no. 1, pp. 283–293, 2009.

[81] S. Obayashi and T. Tsukahara, "Comparison of optimization algorithms for aerodynamic shape design," *AIAA Journal*, vol. 35, no. 8, pp. 1413–1415, 1997.

[82] T. L. Holst and T. H. Pulliam, “Aerodynamic shape optimization using a real-number-encoded genetic algorithm,” in *19th AIAA Applied Aerodynamics Conference*, Anaheim, CA, USA, 2001. AIAA 2001-2473.

[83] M. Wetter and J. Wright, “Comparison of a generalized pattern search and a genetic algorithm optimization method,” in *Proceedings of the Eighth International IBPSA Conference*, Eindhoven, Netherlands, pp. 1401–1408, 2003.

[84] X. Wang and M. Damodaran, “Aerodynamic shape optimization using computational fluid dynamics and parallel simulated annealing algorithms,” *AIAA Journal*, vol. 39, no. 8, pp. 1500–1508, 2001.

[85] M. Bellmore and G. L. Nemhauser, “The travelling salesman problem: A survey,” *Operations Research*, vol. 16, no. 3, pp. 538–558, 1968.

[86] J. K. Lenstra and A. R. Kan, “Some simple applications of the travelling salesman problem,” *Journal of the Operational Research Society*, vol. 26, no. 4, pp. 717–733, 1975.

[87] B. Gavish and S. C. Graves, “The travelling salesman problem and related problems,” Massachusetts Institute of Technology, 1978. Operations Research Center Working Paper OR 078-78.

[88] E. Duman and I. Or, “Precedence constrained TSP arising in printed circuit board assembly,” *International Journal of Production Research*, vol. 42, no. 1, pp. 67–78, 2004.

[89] A. F. Alkaya and E. Duman, “A new generalization of the traveling salesman problem,” *Applied and Computational Mathematics*, vol. 9, no. 2, pp. 162–175, 2010.

[90] J. C. Picard and M. Queyranne, “The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling,” *Operations research*, vol. 26, no. 1, pp. 86–110, 1978.

[91] T. Leipälä and O. Nevalainen, “Optimization of the movements of a component placement machine,” *European Journal of Operational Research*, vol. 38, no. 2, pp. 167–177, 1989.

[92] Y. A. Bozer, E. C. Schorn, and G. P. Sharp, “Geometric approaches to solve the

Chebyshev traveling salesman problem," *IIE transactions*, vol. 22, no. 3, pp. 238–254, 1990.

[93] J. Kirk, "Traveling Salesman Problem - Genetic Algorithm," 2014. Available: `https://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm` [Accessed: 8 March 2018].

[94] A. Seshadri, "Traveling Salesman Problem (TSP) using Simulated Annealing," 2006. Available: `https://www.mathworks.com/matlabcentral/fileexchange/9612-traveling-salesman-problem-tsp-using-simulated-annealing` [Accessed: 8 March 2018].

[95] Yarpiz, "Ant Colony Optimization (ACO)," 2015. Available: `https://www.mathworks.com/matlabcentral/fileexchange/52859-ant-colony-optimization-aco` [Accessed: 3 March 2018].

[96] Yarpiz, "Tabu Search (TS)," 2015. Available: `https://www.mathworks.com/matlabcentral/fileexchange/52902-tabu-search-ts` [Accessed: 14 March 2018].

[97] J. Katz, "Aerodynamics of race cars," *Annual Review of Fluid Mechanics*, vol. 38, pp. 27–63, 2006.

[98] X. Zhang, W. Toet, and J. Zerihan, "Ground effect aerodynamics of race cars," *Applied Mechanics Reviews*, vol. 59, no. 1, pp. 33–49, 2006.

[99] K. R. Cooper, T. Bertenyi, G. Dutil, J. Syms, and G. Sovran, "The aerodynamic performance of automotive underbody diffusers," *SAE Technical Paper 980030*, 1998.

[100] S. R. Ahmed, G. Ramm, and G. Faltin, "Some salient features of the time-averaged ground vehicle wake," *SAE Technical Paper 840300*, 1984.

[101] A. Cogotti, "A parametric study on the ground effect of a simplified car model," *SAE Technical Paper 980031*, 1998.

[102] A. E. Senior and X. Zhang, "The force and pressure of a diffuser-equipped bluff body in ground effect," *Journal of Fluids Engineering*, vol. 123, no. 1, pp. 105–111, 2001.

[103] A. Ruhrmann and X. Zhang, "Influence of diffuser angle on a bluff body in ground effect," *Journal of Fluids Engineering*, vol. 125, no. 2, pp. 332–338, 2003.

[104] O. Ehirim, K. Knowles, A. Saddington, and M. Finnis, "Passive flow control on a ground-effect diffuser using an inverted wing," *SAE International Journal of Passenger Cars - Mechanical Systems*, vol. 11, no. 4, pp. 273–295, 2018.

[105] O. H. Ehirim, K. Knowles, A. J. Saddington, and M. V. Finnis, "Aerodynamics of a convex bump on a ground-effect diffuser," *ASME Journal of Fluids Engineering*, vol. 140, no. 9: 091102, 2018.

[106] A. R. George, "Aerodynamic effects of shape, camber, pitch, and ground proximity on idealized ground-vehicle bodies," *Journal of Fluids Engineering*, vol. 103, pp. 631–637, 1981.

[107] A. R. George and J. E. Donis, "Flow patterns, pressures, and forces on the underside of idealized ground effect vehicles," in *Aerodynamics of Transportation - II: Presented at the Joint ASME-CSME Applied Mechanics, and Fluids Engineering*, vol. 7, Boston, MA, USA, pp. 69–79, 1983.

[108] S. S. Desai, C.-M. B. Lo, and A. R. George, "A computational study of idealized bluff bodies, wheels, and vortex structures in ground effect," *SAE Technical Paper 2008-01-0327*, 2008.

[109] L. S. Puglisevich and G. Page, "Large eddy simulation of the flow around a diffuser-equipped bluff body in ground effect," in *Proceedings of the ASME 2011 International Mechanical Engineering Congress & Exposition*, Denver, CO, USA, 2011.

[110] X. Zhang, A. Senior, and A. Ruhrmann, "Vortices behind a bluff body with an upswept aft section in ground effect," *International Journal of Heat and Fluid Flow*, vol. 25, no. 1, pp. 1–9, 2004.

[111] S. Mahon, X. Zhang, and C. Gage, "The evolution of edge vortices underneath a diffuser equipped bluff body," in *12th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 2004.

[112] O. Ehirim, K. Knowles, A. Saddington, M. Finnis, and N. Lawson, "On the near-wake of a ground-effect diffuser with passive flow control," *International Journal of Automotive Technology*, vol. 20, no. 1, pp. 11–23, 2019.

[113] O. J. Breslouer and A. R. George, "Exploratory experimental studies of forces and flow structure on a bluff body with variable diffuser and wheel configurations," *SAE Technical Paper 2008-01-0326*, 2008.

[114] L. Jowsey and M. Passmore, "Experimental study of multiple-channel automotive underbody diffusers," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 224, no. 7, pp. 865–879, 2010.

[115] J. Knight, M. Spicak, A. Kuzenko, G. Haritos, and G. Ren, "Investigation of vehicle ride height and diffuser ramp angle on downforce and efficiency," in *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, pp. 2139–2145, 2018.

[116] O. H. Ehirim, *Aerodynamics and Performance Enhancement of a Ground-effect Diffuser.* PhD thesis, Cranfield University, 2017.

[117] J. P. Howell, "The influence of ground simulation on the aerodynamics of simple car shapes with an underfloor diffuser," in *Conference on Vehicle Aerodynamics*, Loughborough, UK, pp. 36.1–36.11, Royal Aerodynamic Society, 1994.

[118] K. R. Cooper, J. Syms, and G. Sovran, "Selecting automotive diffusers to maximise underbody downforce," *SAE Technical Paper 2000-01-0354*, 2000.

[119] A. Huminic and G. Huminic, "Computational study of flow in the underbody diffuser for a simplified car model," *SAE Technical Paper 2010-01-0119*, 2010.

[120] A. Huminic and G. Huminic, "Aerodynamic study of a generic car model with wheels and underbody diffuser," *International Journal of Automotive Technology*, vol. 18, no. 3, pp. 397–404, 2017.

[121] Y. Kuya, K. Takeda, X. Zhang, S. Beeton, and T. Pandaleon, "Flow separation control on a race car wing with vortex generators in ground effect," *Journal of Fluids Engineering*, vol. 131, no. 12: 121102, 2009.

[122] Y. Kuya, K. Takeda, X. Zhang, S. Beeton, and T. Pandaleon, "Flow physics of a race car wing with vortex generators in ground effect," *Journal of Fluids Engineering*, vol. 131, no. 12: 121103, 2009.

[123] L. Jowsey, *An Experimental Study of Automotive Underbody Diffusers.* PhD thesis, Loughborough University, 2013.

[124] O. H. Ehirim, K. Knowles, and A. J. Saddington, “A review of ground-effect diffuser aerodynamics,” *Journal of Fluids Engineering*, vol. 141: 020801, 2019.

[125] A. Huminic and G. Huminic, “Aerodynamics of curved underbody diffusers using CFD,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 205: 104300, 2020.

[126] K. Burgin, P. C. Adey, and J. P. Beatham, “Wind tunnel tests on road vehicle models using a moving belt simulation of ground effect,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 22, pp. 227–236, 1986.

[127] P. Kekus and D. Angland, “Automatic wind tunnel-based optimisation of an automotive underbody diffuser,” in *2018 AIAA Aerospace Sciences Meeting*, Kissimmee, FL, USA, 2018. AIAA 2018-0045.

[128] P. Kekus, “Database of aerodynamic measurements of automotive underbody diffusers in the presence of rake,” 2021. DOI: https://doi.org/10.5258/SOTON/D2040.