AIR FORCE ACADEMY

Multidisciplinary Optimisation of an Unmanned Aerial Vehicle With a Fuel Cell Powered Energy System

Bernardo Miguel Teixeira Alves

ALF/ENGAER 139425-J

Thesis to obtain the Master of Science Degree in

Military and Aeronautical Sciences - Aeronautical Engineering

Examination Committee

Chairperson:	COR/ENGAER Luís António Monteiro Pessanha
Supervisors:	Prof. André Calado Marta
	MAJ/ENGAER Luís Filipe da Silva Félix
Member of the Committee:	Prof. Pedro Vieira Gamboa

Sintra, December 2021

Rome was not built in one day...

Acknowledgments

This dissertation marks an important milestone in my life. It was the ultimate task before graduating from the academy and I could not have gone so far with the help and tremendous support from my family and my friends. I was lucky to meet brave and incredible people in 2015, when I first started this journey. Strangers at first, later friends and comrades for life. You helped me push through my physical and mental limits. Without your daily support, concluding the first year would had been nearly impossible. I salute you all.

I must also appreciate my family, to whom I owe everything. You supported me since the day I was born and provided me with everything that was needed for success. Without you, I would not had been able to even reach the first year of this journey.

My dearest friends, thank you for sharing incredible moments with me, be it in land or sea. All the great holidays, weekends and coffee breaks were only possible thanks to you. Even though I slept less during such times I always felt my tank had been re-fueled after them, and I would be ready to go, once again.

Finally, I also show my gratitude to my supervisors. Maj. Félix convinced me of taking this incredible challenge, and provided me with any material that I needed. Prof. Marta has been an extraordinary teacher and advisor. Your guidance and dedication in the last several months were crucial for this work to be completed. Furthermore, the constant discussions that we had, helped me solve many problems and also develop my cognitive skills, which I hope to further sharpen and put them to the best possible use that I can find. Every time I was with no ideas on how to solve a particular problem your suggestions were fundamental to move forward towards a solution.

I will take with me the acquired knowledge and I am certain it will be valuable in the years to come.

Resumo

Para explorar a utilização de células de combustível a hidrogénio como alternativa viável aos combustíveis nocivos em veículos aéreos não-tripulados, um conceito de UAV de classe I foi desenvolvido no Centro de Investigação da Força Aérea (CIAFA). Este trabalho foca-se nos estudos *trade-off* realizados durante a sua conceção e na subsequente otimização. Primeiro, uma abordagem de otimização multi-objetivo foi utilizada com o auxílio do algoritmo genético NSGA-II para balancear dois objetivos em conflito: peso reduzido; e elevada autonomia. Conclui-se que é possível voar mais de três horas com um peso máximo à descolagem de 21,6 kg, uma célula de hidrogénio de 800 W e 148 g de hidrogénio. Uma configuração mais pesada com maior potência nominal e mais combustível foi descartada devido a um constragimento na envergadura. Posteriormente, com um conceito que satisfaz os requisitos impostos, uma abordagem multi-disciplinar (MDO) foi utilizada para maximizar a autonomia. O software utilizado foi o OpenAeroStruct, método dos elementos finitos (FEM) e o método da malha de vórtices (VLM) para modelar superfícies sustentadoras. Inicialmente, uma condição de cruzeiro e de carga foram utilizadas com torção geométrica da asa como variável de projeto. Posteriormente, maior complexidade foi introduzida através da utilização de afilamento, corda e envergadura. Finalmente, uma terceira condição de voo foi introduzida com o intuito de garantir o requisito de perda. Com a utilização de MDO foi possível aumentar a autonomia em 21% satisfazendo todos os requisitos. Este trabalho marca um passo importante no desenvolvimento de um futuro protótipo no Centro de Investigação.

Palavras-chave: MDO, UAV, Veículo verde, Otimização Multi-Objetivo, eVTOL, Projeto Conceptual

Abstract

To explore the use of hydrogen fuel cells as a feasible alternative to pollutant fuels on Unmanned Aerial Vehicles (UAVs), a class I concept was designed at the Portuguese Air Force Research Centre. This work focuses on the trade-off studies performed during its design and on the optimisation that followed. First, a multi-objective optimisation approach was used with the aid of the Algorithm NSGA-II to balance between two conflicting objectives: low weight and high endurance. It was found that it is possible to fly for more than 3 hours with a Maximum Take-off Weight of 21.6 kg, an 800 W fuel cell and 148 g of hydrogen. A heavier configuration with more power and fuel was discarded due to a wingspan constraint. Later, after the concept satisfied the project requirements, Multi-Disciplinary Design Optimisation (MDO) was performed to achieve the maximum endurance possible. The software used was OpenAeroStruct, low fidelity Finite Element Analysis (FEA) and Vortex Lattice Method (VLM) to model lifting surfaces. Initially, a cruise and a load flight point were used with wing geometric twist only as design variable. After, more complexity was added by introducing taper, wing chord and span. Finally, a third flight point was introduced to ensure the stall requirements were satisfied. The use of MDO allowed a 21% increase in endurance with a smaller wing area. Other improvements could not be achieved without violation of the constraints. This work marks an important milestone in the development of a future prototype at the Research Centre.

Keywords: MDO, UAV, Green Aircraft, Multi-objective optimisation, eVTOL, Conceptual Design

Contents

List of Tables

List of Figures

Abbreviations

CFD Computational Fluid Dynamics

CIAFA Centro de Investigação da Academia da Força Aérea

CSM Computational Solid Mechanics

EA Evolutionary Algorithm

FEA Finite-Element Analysis

FEM Finite-Element Method

GA Genetic Algorithm

GPS Global positioning System

HALE High Altitude Long Endurance

MDO Multidisciplinary Design Optimisation

MOEA Multi-Objective Evolutionary Algorithm

MOOP Multi-Objective Optimisation Problem

MTOW Maximum Take-Off Weight

NASA National Aeronautics and Space Administration

NSGA Non-Dominated Sorting Genetic Algorithm

RAM Random Access Memory

SLSQP Sequential Least Squares Program

SOOP Single-Objective Optimisation Problem

UAS Unmanned Aircraft System

VLM Vortex Lattice Method

VTOL Vertical Take-Off and Landing

Nomenclature

Greek symbols

α	Angle of attack.
α_i	Angle of incidence of the tail
β	Angle of side-slip.
κ	Thermal conductivity coefficient.
λ	Taper ratio.
μ	Molecular viscosity coefficient.
ρ	Density.

Roman symbols

H_2	Hydrogen.
AR	Aspect Ratio.
C_D	Coefficient of drag.
C_L	Coefficient of lift.
C_M	Coefficient of moment.
C_{D_i}	Coefficient of drag due to lift.
C_{D_v}	Coefficient of drag due to viscuous effects.
C_{D_w}	Coefficient of drag due to compressible effects.
C_l	Sectional coefficient of lift.
C_{root}	Chord at root.
e	Specific energy.
L/D	Aerodynamic efficiency.
Re	Reynolds' number.

S_{ref}	Area of reference.
S_{wing}	Area of the wing.
V_{stall}	Stall speed.
W_0	Aircraft empty weight without structures.
W_s	Structural weight.
p	Pressure.
$\mathbf{u}$	Velocity vector.
u, v, w	Velocity Cartesian components.

Subscripts

∞	Free-stream condition.
i, j, k	Computational indexes.
n	Normal component.
x, y, z	Cartesian components.
ref	Reference condition.

Chapter 1

Introduction

1.1 Importance of Optimisation in Aircraft Design

The present work deals with the design optimisation of an Unmanned Aerial Vehicle (UAV). It is then appropriate to discuss the importance of optimisation in the design of aircraft.

A *design* can be defined as ”a subset of regular patterns and one that has a preconceived goal, requires planned actions to prepare, and serves a specific purpose”, (Gudmundsson, 2013, p.2). Aircraft design is the process through which the *design* of an aerial vehicle is achieved. It is a complex and cognitively demanding task in part due to the different disciplines involved in it, such as aerodynamics, structures, propulsion and control. Coupling effects among them exist such that they cannot be incorporated in the design process independently of each other. In addition, several disciplines also imply a high number of design variables which contributes to the complexity of the process. Oftentimes, these variables impact different disciplines in distinct manners.

Due to its complex nature, intuition alone is not sufficient to produce a good solution. This is where optimisation is crucial as it allows an efficient and automatic exploration of the design space to seek the best solution of all. Optimisation thus brings innovation.

Still, optimisation has another benefit. It can also be used to perform trade-off studies and provide key insights of what the cost of improving the design with respect to one goal in detriment of another would be. This information can then be used to refine the project requirements and guide subsequent optimisations.

An example of optimisation in aviation was the development of the *Spirit of St Louis* monoplane according to Charles Lindbergh’s requirements to fly non-stop from New York to Paris. Such achievement was first made in 1927 when few thought it would be possible. He understood the negative impact that carrying unnecessary weight would have on the airplane’s range and therefore decided to fly solo, without parachute, navigation tools or radio (Kelley, 2010). With the amount of weight saved by giving up on those items he maximised the amount of fuel carried at takeoff which ultimately allowed him to fly the 3633 miles that separated the two locations.

1.2 Unmanned Aircraft Systems: Past and Present Roles

Unmanned Aircraft Systems (UAS) is the term used to describe the whole system which is made of smaller sub-systems like the vehicle itself, its payloads, the control station, the communication system, the launch system or the transport system. Unmanned Aerial Vehicle (UAV) refers to the vehicle only (Austin, 2011).

The use of UAVs by the military is not new. Some early versions of cruise missiles like the British Army's *aerial Target* had been developed as early as 1914. Although several different types had been built and tested by the end of 1918 they were not effective enough to be used during World War I (Werrell, 1985).

During World War II, target UAVs were built and used by both British and American military personnel for gunnery practise. On the side of the Axis powers coalition, the V-1 cruise missil was used to bomb London. Although the system lacked accuracy, the main goals of causing terror and urban damage were accomplished (Zaloga, 2011). Some of these systems were already equipped with early forms of radio control.

In the post-war period, this type of vehicles started to be used as radar decoy systems. They were dropped from a parent aircraft to confuse the enemy's radar systems.

Amidst the Cold War, jet-propelled UAVs were being developed for reconnaissance and surveillance operations. In 1971, the Boeing Gull system was the winner of the Compass Cope competition organised by the United States Air Force (Newcome, 2004). The goal was to select the first High Altitude Long Endurance (HALE) surveillance UAS to be introduced in operations. The vehicle was capable of flying more than 20 hours and performing photo-reconnaissance in all weather conditions.

During the 1990 decade, the increase in availability of both the Global Positioning System (GPS) and satellite communications allowed UAVs to be operated outside radio tracking range and with increased navigational accuracy when compared to their predecessors, which were equipped with dead-reckoning systems that depended upon inboard gyroscope and data inputs. Some examples are the General Atomics *Gnat* and, later in the decade, the first model of the *Predator (A)* and also the Northrop-Grumman *Global Hawk*. Also developed during this decade was the synthetic aperture radar (SAR) systems (Cutrona, 1990; Tsunoda et al., 2000), which allowed images to be obtained from above the clouds. Such technology made possible to fly UAVs at higher altitudes even in presence of bad visibility ultimately contributing to harder detection and increased protection from ground fire by opposing forces.

In the 2000's decade the military UAV usage grew considerably. Some of the most known and employed vehicles being the *Predator* (General Atomics), *Global Hawk* (Northrop Grumman) and the *Scan Eagle* (Boeing/Insitu). Another role that has been performed by UAVs since this decade is Air-to-Ground Attacks. The United States Air Force (USAF) has been using a larger version of *Predator (B)*, also known as *Reaper*, with armament to conduct ground strikes on numerous armed conflicts (Drew, 2010; The Bureau of Investigative Journalism, 2011).

With the continuing technological advances, UAVs became useful for many applications. Some examples of civilian ones are: crop monitoring and spraying; search and rescue; environmental monitoring;

geographical, geological and archaeological survey; nature conservation; disaster control; fire detection and incident control.

In some roles, UAVs are more advantageous than their conventional counterparts, such as long surveillance missions, monitoring of environmental hazardous areas, reconnaissance missions and aeronautical research. For example, in long surveillance operations the aircrew spends many hours on duty without relief, which increases the chance of loss of concentration and overall mission effectiveness. If UAVs with adequate equipment are used, it is possible to conduct the exact same mission while allowing quick substitution of the ground-based operators when needed, which would contribute to an increase on the effectiveness of the operation. Furthermore, the overall cost would also be lower. Another role that would be best suited for the use of UAVs is monitoring operations in hazardous environments such as volcanic sites. Using drones allows the possibility to acquire important data without putting the aircrew at risk (Jordan, 2019).

Also, in reconnaissance operations where achieving stealth is key, using devices like UAVs, which have smaller radar signatures, improves its success rate. In addition, in the unfortunate event that the vehicle is taken down by the opposing forces, no aircrew is involved. The last example of where UAVs might be better suited to perform the given task is in aeronautical research. Building and testing smaller replicas (scaled models) in flight, if properly executed, is a cheaper alternative with regard to both cost and time to full scale models.

In spite of technological advances and the increasing number of mission roles where UAVs are best suited, the complete removal of aircrew is not expected in the near future. Some of the arguments include an increase of the total operational cost; nonacceptance by passengers in typical commercial flights; and on the military side some argue that UAVs are still at a disadvantage when compared to $4th$ and $5th$ generation fighters. Still, there is ongoing debate on this topic.

1.3 Hydrogen as Energy Source for Surveillance UAV

The usage of Hydrogen as an alternative to fossil fuels is of interest to the defense sector. Recently, a project named Resilience Hub Network in Europe with the objective of harvesting energy through sustainable ways and store it in H_2 was financed under the Consultation Forum for Sustainable Energy in the Defense and Security Sector (European Defense Agency, 2020). Moreover, H_2 is considered key in reaching the European Union goal of attaining carbon neutrality and sustainability (Edwards, Kuznetsov, David, & Brandon, 2008; European Union, 2020).

It then becomes useful to explore ways to use hydrogen as an energy source in the different sectors of society. One possible usage would be to power UAVs. Since 2009, with the PITVANT project (Morgado & Sousa, 2009), the Portuguese Air Force has developed and tested UAS, acquiring valuable know-how and experience in both the design and later in the operation of class I UAVs (Caetano & Morgado, 2019).

On smaller eletric UAVs, the trend is to use lithium batteries to power the motors. These systems are characterised by their low response time and precise power control. Some of their drawbacks are the low specific energy ($W.h/kg$) when compared to fossil fuels, high charging time and limited number of charging cycles. As a result, the weight of the batteries can be considerable with respect to the Maximum Take-Off Weight (MTOW) of the vehicle and their lifespan is short.

Hydrogen fuel cells can mitigate some of the mentioned disadvantages. This fuel has a higher specific energy than the majority of fuels (Edwards et al., 2008) and the fuel cells that convert it to electricity have a much greater lifespan than the lithium batteries. Depending on the temperature and pressure this molecule can be found in various forms. An overview of hydrogen storage methods is provided by Züttel (2004). The most common ones consist of high-pressure gas cylinders that must be designed according to international standards (Stetson, McWhorter, & Ahn, 2016), due to safety concerns related to the high flammable and explosive characteristics of H_2 when stored in this form. Typically, H_2 tanks can be used between -40^o and 85^o (de Miguel et al., 2016). The use of fuel cells and hydrogen gas tanks should be done in well ventilated areas and only certified technicians should carry the fitting of regulators and filling of cylinders. Despite safety concerns, the filling process is much faster than the charging of batteries with similar energy content.

With the intent of exploring the application of fuel cells on UAVs and acquire knowledge in the field of renewable energies, a project to design, build and later test an UAS prototype was defined by the Portuguese Air Force Research Centre (CIAFA). This prototype must be of class I, powered by a hydrogen fuell cell and suited to perform surveillance missions, such as fire detection and/or incident control in the case of forest fires. The UAS project contemplates the design of the UAV itself and other subsystems such as payloads, navigation and communications. The present work together with those of Coelho (2021), P. Silva (2021) and Sá (2021), focus on the design of the UAV, while the one of G. Silva (2021) focuses on the selection of suitable avionics for the navigation and communications subsystems. A representative flowchart of the design process is provided in Figure 1.1.

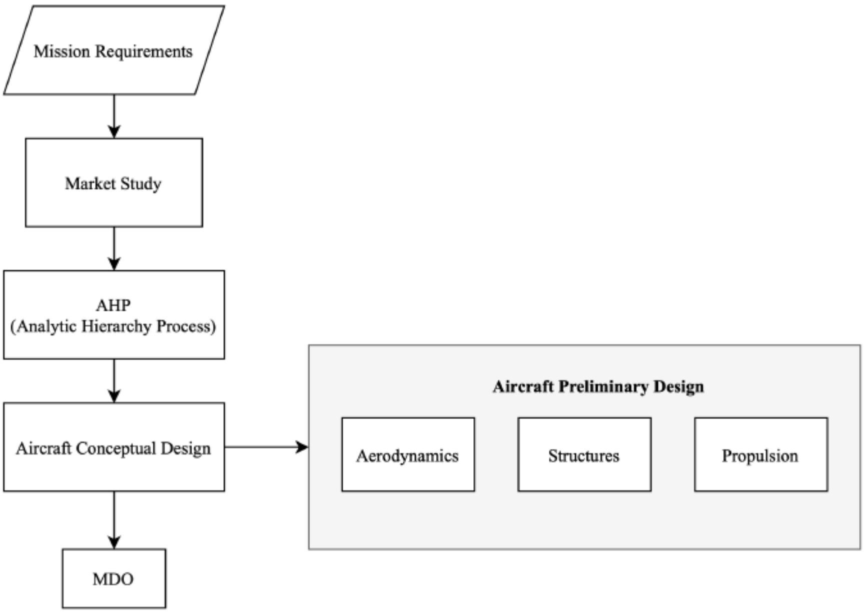


Figure 1.1: Project flowchart

The design of the UAV starts with the mission requirements that were previously defined by the Portuguese Air Force, Table 1.1 followed by a market study, an analytic hierarchy process and a conceptual design. All these were accomplished together with the other authors, a summary is provided in Alves et al. (2021).

Requirement	Value	Units	Description
MTOW	25	kg	-
Payload	2	kg	Include a gimbal
Endurance	> 2	h	Preferable over 3 hours
Cruise speed	35-45	kts	-
Stall speed	< 25	kts	Without flaps
Maximum speed	70	kts	Level flight
Ceiling	15 000	ft	
Takeoff & landing	-	-	VTOL, fully autonomous
Maximum take-off altitude	10 000	ft	Above mean sea Level
Propulsion type	-	-	Electric

Table 1.1: Design requirements by CIAFA

A market study was conducted to better estimate some initial design parameters and help the authors define some aircraft configurations that would be suitable to satisfy the given mission requirements.

After having several possible aircraft configurations, a analytic hierarchy process was done to decide which of these would be further designed and analysed.

With the general configuration selected, a simple computational tool was developed and used to perform the initial sizing of the vehicle and to perform some trade-off studies to help the project's decision makers understand the impact that some design parameters had on two important vehicle properties: its maximum take-off mass and its endurance.

After the conceptual design, an aircraft configuration was kept fixed for preliminary design where more detail in the different disciplines (aerodynamics, structures and propulsion) is achieved through the use of higher fidelity tools such as Computational Fluid Dynamics (CFD) to analyse the aerodynamics and propulsion disciplines and Computational Solid Mechanics (CSM) to analyse the structures discipline.

1.4 Objectives and Deliverables

The goal of the present work is to improve the conceptual design of an electric UAV powered by a fuel cell system using a multidisciplinary approach. The coupling between the different disciplines has to be considered and a suitable framework selected. Analysis of the complete system in this framework is meant to be performed to validate the results obtained during the conceptual phase of the project and later search for a better solution.

The deliverables of this work are the following:

1. Development of an *Open Source* software to aid in the initial sizing of aircraft, capable of performing trade-off studies[1];

2. UAV concept with an appropriate fuel cell and tank for H_2 storage [1];

3. Added functionality to OpenAeroStruct, an *Open Source* software developed by Jasa, Hwang, and Martins (2018), to handle fuel-cell systems and broader wing geometry definition;

4. New planform design that maximises the endurance of the baseline UAV, satisfying all operational constraints.

1.5 Thesis Outline

This document is organised in 2 parts, the first, which correspond to Chapters(2-3) are dedicated to multi-objective optimisation in the context of aircraft conceptual design. The second part, Chapters(4-5) deals with a multidisciplinary approach to perform optimisation of the baseline model with respect to a single objective.

The first chapter provided some insight on the importance of optimisation in aircraft design, the role of UAS in Defense and some background on the project of which the present work is part.

In the second chapter an overview of multi-objective optimisation is given. Two main approaches to multi-objective optimisation problems are presented followed by a comparison of different optimisation algorithms. Afterwards, some key concepts of Multi-objective optimisation are described and finally suitable evolutionary algorithms to solve multi-objective optimisation problems are discussed.

Moving forward to Chapter three, the connection between multi-objective optimisation and conceptual aircraft design is made. Also in this chapter, a framework to perform optimal trade studies in aircraft design is presented as well as the results of such trade-off studies and the final aircraft concept generated, which is the starting point for the second part of the work.

In the fourth chapter, the multidisciplinary approach is justified first and then an Open-source software developed to perform Multidisciplinary Design Optimisation (MDO) in aircraft design is discussed. After, some extra features added by the author that change the behaviour of the original ones are briefly explained.

The fifth chapter contains the formal optimisation problem, the mesh convergence study, the analysis of the baseline UAV, some parametric studies with respect to some parameters that have to be fixed and cannot be used as design variables in the optimisation problem, and finally the optimal solutions. These are presented from the simpler to the more complex. The chapter ends with a comparison of the most relevant solutions found.

Finally, in chapter six some concluding remarks are presented followed by suggestions for future work.

---

[1] this was accomplished together with P. Silva, Coelho and Sá

Chapter 2

Multi-Objective Optimisation

2.1 Theoretical Overview

'Optimisation is the process of maximising or minimising a desired objective function while satisfying the prevailing constraints'. (Belegundu & Chandrupatla, 2019).

If an optimisation problem involves more than one objective, it is classified as a multi-objective optimization problem (MOOP), whereas if it involves only one, it is classified as a single-objective optimisation problem (SOOP). A MOOP involving no conflicting objectives will have only one optimal solution and the problem can be treated as a SOOP. Aerospace design, however, is often a trade-off between several goals (i.e objectives), such as performance, cost and time-to-market (Marta, 2020). When conflict exists between different objectives, there are fundamental differences between single-objective and multi-objective optimisation.

In the case of multi-objective problems with conflicting objectives, a gain in one objective implies a loss in the other(s). As such, instead of one optimal solution, as in the case of single objective problems, a set of several optimal ones is found, with no solution being better than any other. The reason lies in the fact that there exists no solution that is simultaneously better in all objectives than any other solution of this set.

That is the fundamental difference between single and multi-objective optimisation problems: in single objective, one objective function is evaluated and improved as much as possible without violation of any of the problems constraints; in multi-objective, all of the objective functions are considered equally important and a set of different optimal solutions are found due to the trade-off of conflicting goals.

In MOOPs, the goal is not to find a single optimal solution but instead search for a set of multiple optimal solutions. Because these objective functions share some, if not all of the design variables, it is not possible to treat each objective function independently. Therefore, it is not possible to find a solution of design variables that attain the maximum or minimum value of each of them simultaneously.

Similarly to SOOPs, the problem usually is subjected to some constraints which have to be satisfied.

In its general form the Multi-Objective Optimisation Problem (MOOP) can be postulated as

$$\begin{aligned} \text{Minimise/Maximise} \quad & f_m(\boldsymbol{x}), & m = 1, 2, ..., M; \\ \text{subject to} \quad & g_j(\boldsymbol{x}) \leq 0, & j = 1, 2, ..., J; \\ & h_k(\boldsymbol{x}) = 0, & k = 1, 2, ..., K; \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, ..., n. \end{aligned} \tag{2.1}$$

Where $\boldsymbol{x}$ is the solution vector of n design variables, $\boldsymbol{x} = [x_1, x_2, ..., x_n]^T$, $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$ represent the inequality and equality constraint functions, respectively. The last collection of constraints are the design variable bounds which restrict each variable x_i to take values between the lower and upper values of x_i^L and x_i^U, respectively. The region delimited by these variable bounds is named the design space $\mathcal{D}$. A solution $\boldsymbol{x}$ that does not satisfy all the $J + K$ constraints and the $2N$ variable bounds is an infeasible solution. On the other hand, if $\boldsymbol{x}$ satisfies all the previous constraints, it is a feasible solution. The set of all the feasible solutions is denominated the feasible region $\mathcal{S}$.

In total there are M objective functions in the above formulation, where each of them can be either minimised or maximised. The series of objective functions can be assembled in a vector $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_M(\boldsymbol{x})]^T$. The different values of $\boldsymbol{f}(\boldsymbol{x})$ are represented in a multi-dimensional space named the objective space, $\mathcal{Z}$. The feasible objective space, obtained from the set $\mathcal{S}$ is denominated the Criterion Space $\mathcal{Z}_{\text{feasible}}$.

These two multi-dimensional spaces constitute another difference between single and multi-objective optimisation problems. In single objective problems, there is only one space, the n-dimensional design space $\mathcal{D}$, while in multi-objective problems, there is also the M-dimensional objective space, $\mathcal{Z}$.

If an optimisation problem has only linear objective functions and constraints, it is classified as a multi-objective linear program (MOLP). If any of the objective functions and or constraints are nonlinear it is named a nonlinear multi-objective problem. Linear optimisation problems have several properties that can be exploited. However, because in Aerospace Design it is almost impossible to formulate a meaningful problem with only linear objective functions and constraints, these properties do not possess any practical value to the present work and therefore will not be explored.

Whenever multiple conflicting objectives, all equally important, are present in an optimisation problem there is no single optimal solution. As a result of the conflicting nature of some objectives, a solution $\boldsymbol{x}$ cannot produce the best result for all objectives f_m and a set of optimal solutions arise. It becomes necessary to identify the solutions of that set so that the user can observe the best trade-offs. The Pareto-Front is the curve in the objective space $\mathcal{Z}$ that contains all the optimal trade solutions. Between any two solutions in that curve, none can be said to be better than the other. This curve is found by a component-wise comparison of the objective function vector $\boldsymbol{f}$ of all the found feasible solutions $\boldsymbol{x}$ in the design space $\mathcal{D}$.

An example of a two-function Pareto-Front can be observed in Figure 2.1. It is represented as a black line connecting solutions A, B, C and D while passing through several intermediate optimal solutions. All $\boldsymbol{x}_i$ in the Pareto-Front are named Pareto-optimal solutions. The Pareto-Front was obtained in a multi-objective problem which consisted in the minimisation of an objective function $\boldsymbol{f} = [f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})]^T$ where f_1 corresponds to the beam mass in kg and f_2 the beam end deflection measured in mm.

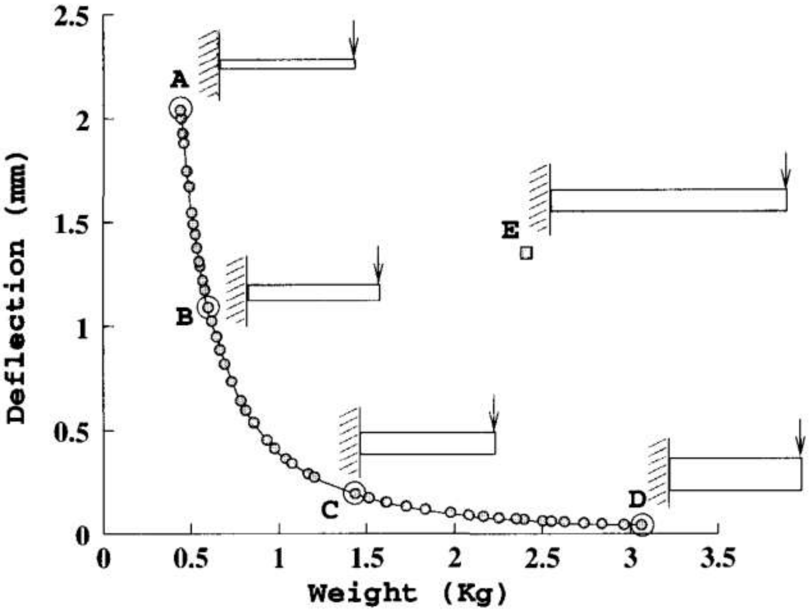


Figure 2.1: Pareto-front retrieved from Deb (2001)

Through comparison of solutions B and E, refer to Figure 2.1, it is possible to verify that solution B is better in both objectives (the beam it represents has a lower weight and a lower end-deflection) than solution E. Whenever this happens it is common to say that solution B dominates solution E. If a comparison is made between B and C (two solutions in the Pareto-front), nothing can be concluded with regard to which of the two is better. Solution B represents a beam with a lower weight than the beam solution C represents, but at the same time, the end deflection of B is higher than the end deflection of C and thus it is worse with respect to the second objective. Because in one objective, solution B is better and in the other is worse, both solutions are considered to be optimal without one dominating the other. Finally, if a comparison is made between solutions E and D again, no conclusion can be drawn. Solution D represents a beam with a lower end deflection (better) but with a higher weight (worse). Solution E, however, does not reside in the front because it is worse (in both objectives) than solutions B and C. This example illustrates an important fact: Just because two solutions are non-dominated with respect to each other it does not imply that they are Pareto-optimal.

The following characteristics of a Pareto-front can be deduced:

- Any solution in the Pareto-front is non-dominated by any other in the objective space $\mathcal{Z}$;
- Any solution outside the Pareto-Front is always dominated by at least one solution in the front.

In any MOOP, the goal is to find the Pareto-front with the most diverse possible set of non-dominated solutions to obtain valuable trade-offs with respect to all objectives.

Both the Pareto-front and the Pareto-optimal solutions provide relevant knowledge that can help designers compare and choose a good compromise between the several goals. As such, the Pareto-front is a tool that provides value to the decision-making process by showing the best compromises that can be accomplished. If the problem is transformed into a single-objective one, this information will be lost.

The dominance concept is important as it is used by most multi-objective optimisation methods to compare different solutions.

In case one objective function is to be minimised, then one solution being better (with respect to that objective) than another implies that $f_m(\boldsymbol{x}^{(1)}) < f_m(\boldsymbol{x}^{(2)})$. If the objective function is to be maximised then the previous relation is valid with a change in the operator from '$<$' to '$>$'. A solution $\boldsymbol{x}^{(1)}$ dominates a solution $\boldsymbol{x}^{(2)}$ if:

- Solution $\boldsymbol{x}^{(1)}$ is no worse than $\boldsymbol{x}^{(2)}$ in all objectives, $f_i(\boldsymbol{x}^{(1)}) \sim \triangleright f_i(\boldsymbol{x}^{(2)})$ for all $i = 1, 2, ..., M$;
- Solution $\boldsymbol{x}^{(1)}$ is strictly better than solution $\boldsymbol{x}^{(2)}$ in at least one objective, $f_k(\boldsymbol{x}^{(1)}) \triangleleft f_k(\boldsymbol{x}^{(2)})$ for at least one $k \in \{1, 2, ..., M\}$.

The dominance concept allows pair-wise comparison of any two solutions in the objective space $\mathcal{Z}$. For any set of solutions $\mathcal{P}$, it is possible to perform all pair-wise comparisons using the dominance criterion and find a subset of solutions $\mathcal{P}^{'}$ that are non-dominated by any solution $\boldsymbol{f}(\boldsymbol{x}^{(k)}) \in \mathcal{P}$. This set is the non-dominated set of $\mathcal{P}$. When the set of solutions $\mathcal{P}$ corresponds to the Criterion Space $\mathcal{Z}_{\text{feasible}}$, the non-dominated set $\mathcal{P}^{'}$ is the Pareto-optimal set or simply Pareto-front. Any solution $\boldsymbol{f}(\boldsymbol{x}^{(k)})$ in the Pareto-front is Pareto-optimal.

2.2 Approaches to Multi-Objective Optimisation

There are several ways of dealing with multiple criteria in an optimisation problem. Deb (2001) proposed two approaches: an ideal approach where the optimisation problem is treated as multi-objective and a preference-based approach where the problem is first converted into single objective. In the past, due to the lack of suitable computational methods and resources, the most common one was the preference-based approach. Both approaches are going to be briefly explained in the next paragraphs. A flowchart representing each procedure (ideal and preference-based approaches) can be found in Figure 2.2.

The ideal multi-objective optimisation procedure involves the formulation of an optimisation problem with several objectives (all equally important), the usage of a multi-objective optimisation algorithm to find multiple trade-off optimal solutions and, finally, the choice of one optimal solution from the optimal set based on higher-level information. The other approach also begins with the formulation of a multi-objective problem but, in contrast with the first approach, higher-level information is used to estimate a relative importance vector between the several objectives to transform the vector objective function into a scalar one. Afterwards, a single objective optimisation is used to find the optimal solution.

A third alternative, not mentioned in the previous reference, would be to convert all objective functions f_i except one, into constraints g_i and then use a single objective constrained optimisation algorithm to solve the problem and find the optimal solution, in the form

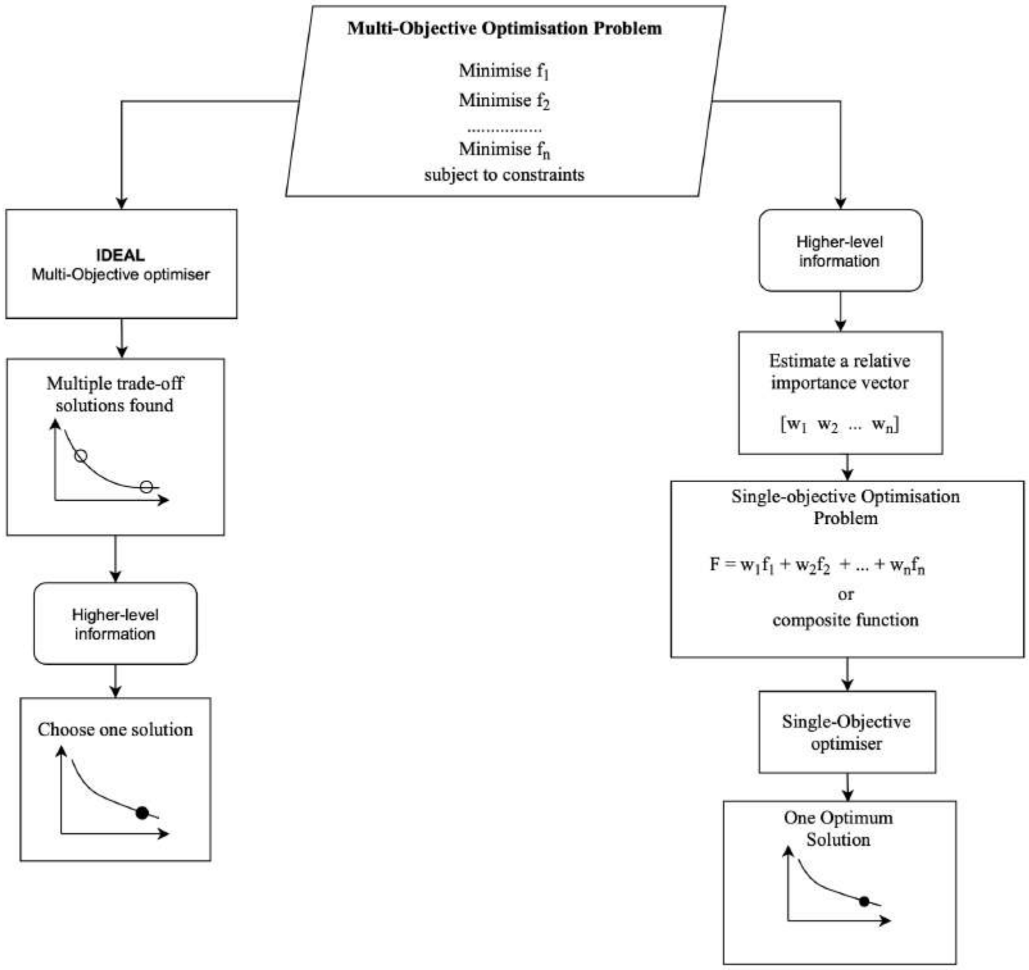


Figure 2.2: Two approaches for solving multi-objective optimisation problems, adapted from Deb (2001)

$$\text{Minimise } \boldsymbol{f} = \begin{Bmatrix} f_1 \\ f_2 \\ ... \\ f_n \end{Bmatrix} \longrightarrow \text{Minimise } f_i \quad \text{Subjected to } \begin{Bmatrix} g_1 \\ g_2 \\ ... \\ g_{n-1} \end{Bmatrix} \leq 0 \tag{2.2}$$

The main difference between the first and last two approaches is how higher-level information is used. In the last two approaches, it is used at the beginning of the formulation to transform the multi-objective problem into a single-objective, while in the first approach it is only used in the end after the different trade-offs are obtained.

The second approach is highly affected by the user-defined relative preference vector while the third approach is highly affected by what objectives are to be transformed into constraints and what values are specified as the *supremum* of each constraint. Both approaches require an analysis of non-technical, experienced-driven and qualitative data. Without knowledge of the different trade-off solutions, the task of analysing such information is harder because the user is defining the relative preference vector (in the

2nd approach) or transforming objective functions into constraints (in the 3rd approach) without knowing beforehand how those decisions will impact the resulting outcome.

The ideal approach requires usage of an optimiser capable of dealing with multiple objectives. Deterministic and trajectory-based algorithms perform optimisation according to a point-by-point search. In each iteration, one solution is evaluated and then modified to provide a better one for the next iteration. These algorithms are efficient for finding one optimal solution per simulation run but cannot be readily used to solve problems with respect to multiple objectives simultaneously where a set of optimal solutions is sought. In order to use this type of algorithms in MOOPs, it is necessary to first convert them into SOOPs.

In contrast with trajectory-based optimisation algorithms, population-based algorithms work with a set of solutions in each iteration instead of a single one. Whenever the problem contemplates only one objective, the initial population will converge to a single, optimal solution. If the optimisation problem has multiple objectives, population-based algorithms can be adapted to capture multiple trade-off solutions in the final iteration of a single simulation run. Well-known population-based algorithms are Evolutionary Algorithms (EA), that mimic nature's evolution to guide its search and optimisation process. As such, when conducting multi-objective optimisation following the ideal approach, Multi-Objective EAs (MOEAs) are employed. The main disadvantage with such algorithms is that they typically require many more function evaluations than gradient-based algorithms to converge to a solution. Depending on the complexity of the objective function to be evaluated and on the available computational power, the cost of using such algorithms can be prohibitive.

By using the proposed ideal approach it is possible to explore different trade-off solutions that will not only help decision makers to accept some losses in some areas to leverage other areas in early design stages but also help the present author to define a single objective optimisation problem to be solved later using higher fidelity models that couple different disciplines.

2.3 Comparison between Classical and Heuristic Algorithms

Independently of the approach chosen, Figure 2.2, there is always the need to choose and implement an optimiser. In this section an overview on the different types of algorithms available to solve optimisation problems is provided.

Deb (2001) distinguished between classical and non-classical algorithms. Classical algorithms are characterised by updating only one solution per iteration and using deterministic rules for transforming one solution into a new, better one. A comprehensive overview of such algorithms can be found in Nocedal and Wright (2006) and Ravindran, Ragsdell, and Reklaitis (2006).

Non-Classical algorithms are also named meta-heuristic or simply heuristic optimisation algorithms. Heuristic algorithms have the following characteristics (Gilli & Winker, 2008):

1. Provide high quality approximations to the global optimum;

2. Robust to changes in problem characteristics, i.e. they are suitable to solve a whole class of

problems and are not very sensitive to tuning parameters;

3. Easily implemented to many problem instances

Classical optimisation algorithms typically present some drawbacks:

- The optimal solution depends on the chosen initial one;
- The final solution is usually a local optimum;
- The algorithms were not built to handle discrete search spaces;
- The objective function has to be smooth and continuous;
- The algorithms cannot be directly used in parallel processing.

In real life problems, complex interactions between variables give rise to multiple local optimal solutions. In addition, non-linear constraints are also common. These two effects, together with the deterministic nature of classical algorithms, increase the probability of a solution to converge to a sub-optimal one. As such, unless there is already accumulated knowledge of the problem at hand that can guide the choice of the initial solution, the optimisation problem should be solved with several different initial guesses and the results compared. Such process is time-consuming.

Discrete variables are also common in real life applications. Even geometric parameters due to manufacturing limitations, might have discrete values. In addition, if a design variable x_i is related to a component which is going to be sourced from a supplier, the possible values that x_i can take are subjected to the available offer.

When discrete variables are present in an optimisation problem, it is possible to transform them into continuous variables and solve the problem with any classical optimiser. This process is simple but has some inconveniences: after solving the problem, for each discrete variable, the user has to replace the value obtained with either the higher or lower nearest discrete value and then re-evaluate both the objective function and the constraints. In a problem containing n discrete design variables, the user will have to evaluate 2^n possible solutions. Another disadvantage with this method is that checking the two nearest possible values for each design variable does not guarantee that an optimal real solution will be found. These two obstacles can be overcome by allowing only real discrete values of the design variables to be used in the optimisation problem.

Meta-heuristic algorithms, in particular Evolutionary algorithms, are relevant alternatives to the classical ones. Their application in problems does not rely on any set of strong assumption, as a consequence they are suitable to solve problems with non-smooth objective or constraint functions. They do not need the derivatives of neither the objective function nor of the constraints. At the same time, this type of algorithms have higher convergence rates than Classical Gradient-Free algorithms, they can be used to handle discrete design variables and they can easily take advantage of parallel processing computers.

2.4 Evolutionary Algorithms and Multi-Objective Evolutionary Algorithms

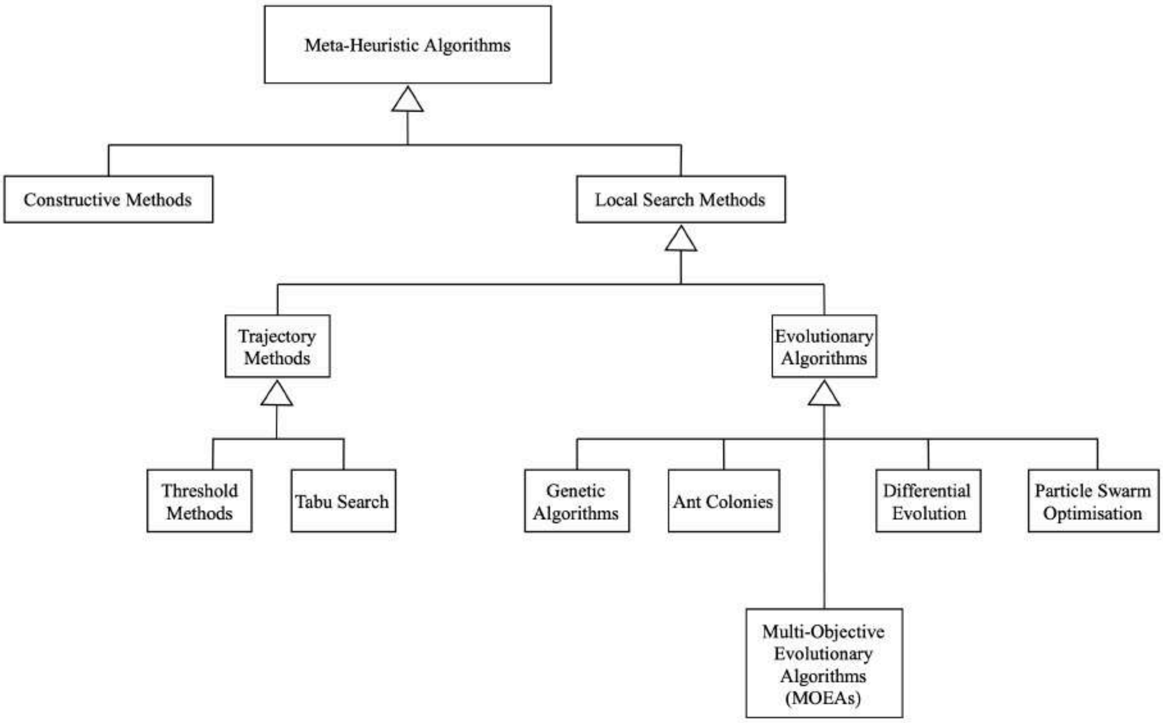


Figure 2.3: Meta-heuristic algorithms categorisation inspired by Gilli and Winker (2008)

Optimisation Heuristics can be divided in two main classes: constructive methods (also denominated greedy algorithms), which construct the solution in a sequence of locally optimum choices, (Cormen, Leiserson, Rivest, & Stein, 2009); and local search methods, that use information about the solutions $\boldsymbol{x}^{(k)}$ in a neighbour region.

Local search algorithms were not used in the past due to the lack of available computational resources. With its increase, more interest has been given to them. The classic local search algorithm can be found in Algorithm 1.

Algorithm 1 Local Search Algorithm, adapted from (Gilli & Winker, 2008)

```
Generate initial solution $\boldsymbol{x}^{(c)}$
while stopping criteria not met do
    Select $\boldsymbol{x}^{(n)} \in \mathcal{N}(\boldsymbol{x}^{(c)})$        ▷ neighbour to current solution
    if Acceptance criteria is met then $\boldsymbol{x}^{(c)} = \boldsymbol{x}^{(n)}$
    end if
end while
```

Local Search Methods can be further categorised in either Trajectory Methods or Evolutionary Algorithms (EAs). Trajectory Methods work with only one solution per generation/iteration while the Evolutionary work with a population of solutions in each iteration/generation. Due to this characteristic, they are also named population-based methods in the literature. The major difference between them resides in how the neighbours are chosen and how the acceptance criteria is defined, statements 2 and 3 in Algorithm 1.

EAs mimic nature's evolutionary principles to update a set of solutions in each iteration/generation. The set of all solutions in a given generation is denominated population. Since they work with several solutions simultaneously instead of a single one, they are expected to better search the design space $\mathcal{D}$ and consequently be more likely to converge towards the global optimal solution. Another advantage is that they can be easily adapted to solve MOOPs. One of the drawbacks of working with populations is that the computational power required to solve the problem in a reasonable time frame when using such algorithms is also high. In Aerospace Design, the actual function evaluation is performed by simulation software, which will have to be executed once for each solution to evaluate statement 4 of the Algorithm 1. This can lead to prohibitive optimisation time and the usage of parallel systems becomes essential.

When solving MOOPs with the ideal approach, a multi-objective optimiser should be employed. A great portion of the algorithms used for this type of applications are special modifications of single-objective EAs. Therefore the working principles of these single objective EAs are required to understand Multi-Objective Evolutionary Algorithms (MOEAs). In the present work, focus is given to Genetic Algorithms (GAs), which are subset of existent EAs since one particular GA was used to solve different optimisation problems and perform trade studies in the conceptual phase of the project. The interested reader can refer to Gilli and Winker (2008) for a review of different EAs such as Ant Colonies (AC), Differential Evolution (DE) and Particle Swarm Optimisation (PS).

Genetic Algorithms (GAs) are a type of Evolutionary Algorithms that use genetic operators to guide the search of the optimal solution. For a comprehensive description of GAs refer to Goldberg (1989) and Reeves and Rowe (2002). GAs can be broadly categorised in Binary-Coded or Real-Parameter GAs. The main difference between the two lies in how the algorithm represents the design variables. As the name suggests, Binary-Coded codify the design variables into binary strings, e.g, a design variable j of solution k: $x_j^{(k)} = [0\ 0\ 1\ 0]$, while Real-Parameter uses numerical representations of all its design variables.

The main advantage of coding the design variables in binary strings is related to the pseudo-chromosomal representation of a solution $\boldsymbol{x}^{(k)}$ that is achieved. Using this type of representation has the following advantages: straightforward implementation of genetic operators; permits different precision in the representation of different design variables, which in turn enables the use of integer variables; variable bounds are enforced as a direct consequence of the coding process.

Nevertheless, Binary-Coded GAs also have some disadvantages associated with the binary representation of the design variables. Some of them are:

- Hamming Cliffs, e.g, $[0\ 1\ 1\ 1\ 1]$ and $[1\ 0\ 0\ 0\ 0]$ represent neighbour solutions in the real space but require 5 bit changes in order to transition from one to the other. This phenomenon causes an artificial obstacle to gradual searches;

- Impossibility of achieving a different numerical precision in the optimal region. The precision used to represent each design variable in a solution $\boldsymbol{x}_j^{(k)}$ is constant during the execution of the algorithm. It is not possible to have lower precision when searching the entire search space and higher precision when searching in the neighbourhood of the optimal. Therefore, if higher precision is desired in the optimal solution, the user has to select a string representation with higher length before running the algorithm. The higher the string length that is defined, the greater the population size requirement will be (Goldberg, Deb, & Clark, 1991) and, as a consequence, the larger the computational cost will be.

Real-Parameter GAs can be used to overcome the mentioned obstacles. Another advantage of this type of algorithm is that performing the evaluation of the objective function $\boldsymbol{f}(\boldsymbol{x})$ and constraints, $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$, is easier than in Binary-Coded GAs because there is no need to decode the value of the different design variables since they were never codified in the first place.

In spite of the advantages, Real-Parameter GAs have some disadvantages related to the use of the genetic operators that were developed with binary string representation in mind, particularly, the crossover and mutation operators. Though there exist differences between the two, both have the same general principles and workflow. These differences will not be described in-depth since only a general overview is sought.

Genetic Algorithms use the principles of natural selection and genetics to guide its search towards the optimal solution. They begin by generating an initial population, then evaluate the objective function and constraints of each individual solution. Next, they assign a fitness value and finally verify if a termination criteria is met. In case the condition is verified, the algorithm stops and the best solution among the current population is returned to the user (in the case of single objective optimisation). In case the criteria is not met, the genetic-inspired operators namely Reproduction, Crossover and Mutation are applied to the current population and a new one is generated. The whole process repeats until the termination criteria is met. NSGA-II is an example of a GA, and although it has some unique operators namely the Non-Dominated Sorting and the Crowding Distance Sorting its general workflow is common to all GAs, refer to Figure 2.4.

Generating the initial population is similar to generating an initial solution $\boldsymbol{x}_0$ in classic optimisers: if there is already knowledge to guide the generation of the initial population it can be used, otherwise a random population is usually created. The size, i.e, the number of individual solutions in the population has a direct impact on the computational cost and convergence towards the optimal. It is therefore, an important parameter to take into consideration when initialising the algorithm.

After having several solutions, it becomes necessary to evaluate them to identify the best ones and guide the search towards the optimal. In the evaluation process the constraint and objective functions are computed for each individual in the population and a fitness value is assigned accordingly. In some cases, the value of the fitness function is simply equal to the value of the objective function.

The Genetic Operators are used to generate better solutions from generation to generation.

The Reproduction/Selection operator is applied to a population with the goal of making replicas of the good solutions and eliminating bad ones. The operator identifies the best solutions from the population

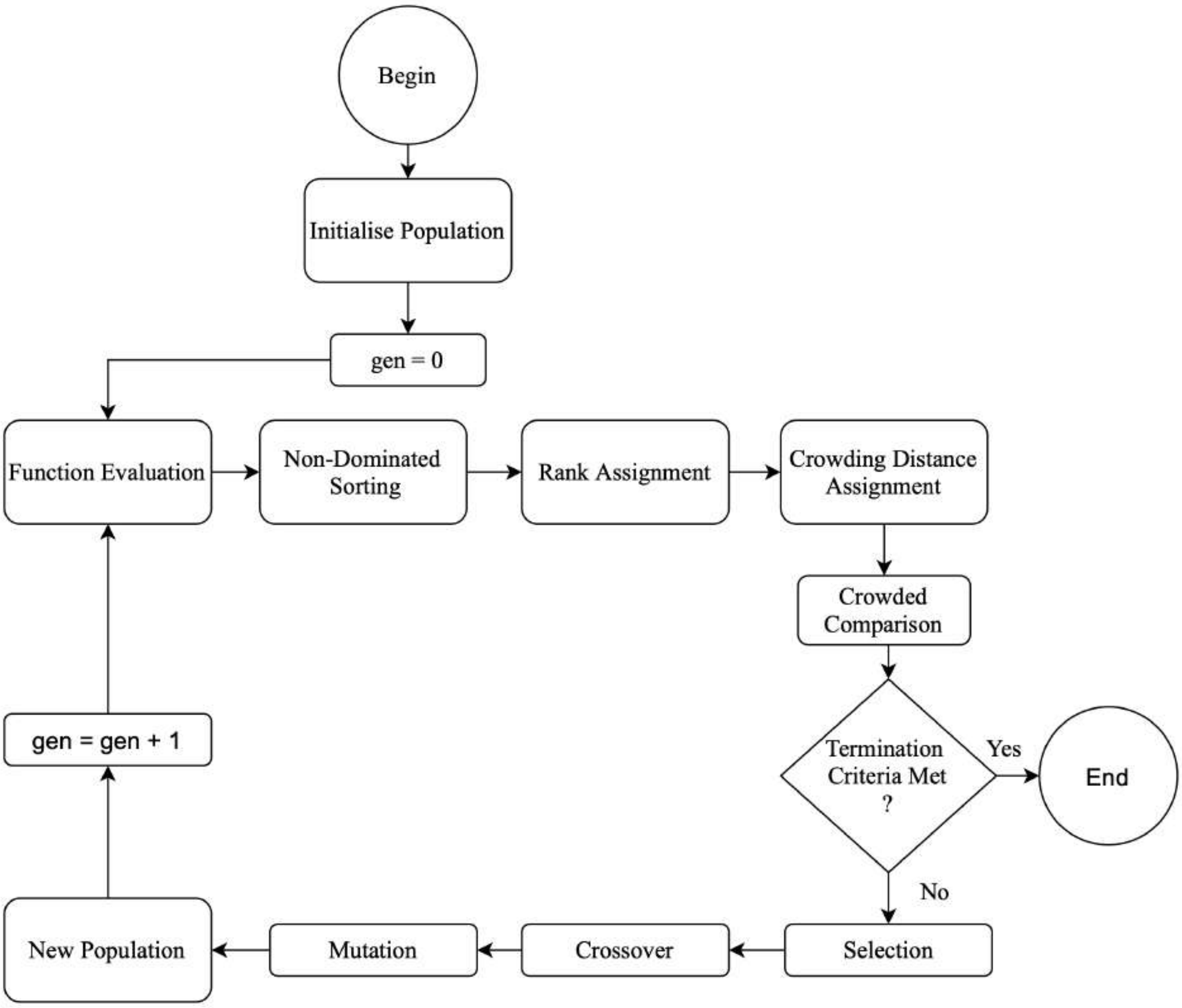


Figure 2.4: NSGA-ii flowchart inspired from: (Deb, 2001)

and then produces exact replicas of them. Subsequently, it eliminates the bad ones and establishes the replicas in their positions. The population formed during the Selection/Reproduction operator is named the mating pool, which is going to be used in the next genetic operator: crossover.

The Crossover operator is applied to the solutions in the mating pool with the intent of generating different ones. Two solutions are picked up from the mating pool and combined together to create a new one. The ones that were picked from the pool are named 'parents' and the resulting solution is named 'offspring'. Both the choice of the parents and the combination to generate the offspring has randomness associated, i.e, with the same mating pool and the same crossover operator it is possible to obtain different results. The selection of the individuals which are going to mate and the actual process that combines them together to generate the offspring are not deterministic.

After crossover, mutation can happen to maintain diversity in the population. Since the working principles of the ones used in Binary-Coded GAs are different from the ones used in Real-Parameter GAs, a brief description for each is provided next.

In Binary-Coded GAs, each individual solution $\boldsymbol{x} = [x_1, x_2, ..., x_n]^T$ is represented as a string of 0s and 1s. Due to this representation, the mutation operator is straightforward: it flips one bit. If it is 0 the bit-wise mutation operation will transform it into a 1; and vice-versa. Mutations do not always happen, the genetic operator selects a bit, randomly, in a string that represents solution $\boldsymbol{x}^{(k)}$ and then changes its value with a probability of p_m. Goldberg (1989) proposed a mutation clock operator that determines the location of the next bit to be mutated based on an exponential distribution.

In Real-Parameter GAs, each individual is represented as is, therefore the bit-wise mutation mentioned previously cannot be used. In order to mutate an individual, one common methodology consists in performing a perturbation in the current solution $\boldsymbol{x}^{(k)}$ to neighbour values using probability distributions. Some common distributions used are: Gaussian Mutation (Schwefel, 1987); Random mutation (Michalewicz, 2013); and Polynomial Mutation (Deb & Agrawal, 1999).

None of the mentioned genetic operators are deterministic; however, it is expected that they will eliminate the worse individuals $\boldsymbol{x}^{(k)}$ and combine the better ones throughout the generations to produce even better ones.

An optional operator that is often used to improve the performance of GAs is the Elite-Preserving Operator (Rudolph, 1999; Zitzler, Deb, & Thiele, 1999). As the name implies it is used to preserve and use the best solutions found in generation j in the following generation $j+1$. One simple implementation consists in copying the best $(\epsilon \cdot 100)\,\%$ of the current population into the next. The remaining $(1-\epsilon)\cdot 100\,\%$ is created through application of the described genetic operators (selection/reproduction, crossover and mutation) to the current population which includes the best $(\epsilon \cdot 100)\,\%$ individuals that were copied directly into it.

One GA suitable to handle MOOPs is the Elitist Non-Dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002). A flowchart representative of this algorithm was already provided in Figure 2.4. It has some unique characteristics that differentiate it from other GAs which are related to the existence of a Non-Dominated Sorting and Crowding Distance Assignment operators.

The Non-Dominated Sorting operator precedes the rank assignment. It assess which solutions are non-dominated and which are dominated, sorting them according to their dominance over the others. Based on this sorting, a rank is then attributed.

The Crowding Distance Assignment operator calculates a metric that assesses how close one individual is to their neighbours. With both the value of this metric and the rank assignment, the Crowded Comparison operator is employed to eliminate the undesired solutions. The result is a sorted population of the same size as the initial one where the rank 1 solutions are the non-dominated ones. If any stopping criterion is met, the algorithm stops and returns this population to the user, otherwise, the described Genetic Operators are used to generate a new offspring population. After generation 0, all the offsprings and parents are subjected to all the operators from the non-Dominated Sorting until the Crowded Comparison. Hence, elitism is ensured.

Chapter 3

Conceptual Aircraft Design Using Multi-Objective Optimisation

This chapter focuses on the trade-off studies that were performed, the tools used as well as their implementation in order to perform the analysis and ultimately guide the decision making at the conceptual stage of the UAV design.

3.1 Initial Sizing

Aircraft sizing determines the take-off gross weight for an aircraft to perform its design mission. Sizing is the single most important calculation for aircraft design (Raymer, 1992).

To conduct the sizing of the UAV, a computer program was developed (Alves et al., 2021, sec. 4). This numerical tool allows for quick estimations of the MTOW and Endurance, as well as the size of the motor, wing, rotor and batteries needed for it to be able to perform the mission defined by the user.

The program uses a simple iterative process to estimate the MTOW of the UAV which was based on (Gundlach, 2014). An absolute tolerance of 1 gram with respect to the weight was defined as stopping criterion and, as fallback, a maximum of 50 iterations were allowed.

One of the disadvantages of using the numerical tool developed is that it does not allow for optimisation or trade-off studies as is. By changing manually some of the program's inputs, it is possible to perform parametric studies, but if the combined effect of different inputs are to be studied then the process quickly becomes laborious and unpractical.

To automatise the process and search the design space for possible optimal solutions, a suitable framework to be used together with this tool was searched.

3.2 Pymoo, A Multi-Objective Optimisation Framework

Pymoo (Blank & Deb, 2020) is an open source multi-objective optimisation framework written in Python. It has several optimisation algorithms suitable for single- and multi-objective optimisation problems, that can easily be customised to suit specific needs. Furthermore, Pymoo is also well documented with several examples and code snippets to reproduce them, which contributes to a fast learning experience. The above characteristics of the Pymoo framework and the author's previous knowledge and experience with the Python programming language contributed to the choice of this framework.

From the possible algorithms available in Pymoo, the NSGA-ii algorithm, described previously in Section 2.4, was used due its capability to handle multi-objective optimisation problems. The optimisation algorithm will execute the numerical tool developed with a given set of design variables to evaluate the vector objective function $\boldsymbol{f}(\boldsymbol{x})$ as well as all the variables needed to assess the constraints, which corresponds to the Function Evaluation and Rank Assignment processes in the NSGA-ii flowchart, in Fig.2.4. In each generation, the numerical tool will be executed once for each individual in the population.

3.3 Problem Formulation

It is well established that Aircraft Design is an iterative process (Raymer, 1992), therefore several trade-off studies were performed with different design variables and constraints.

In this Section the author first presents the complete optimisation problem and during discussion, Section 3.5, the Design Variables as well as the constraints used to obtain each set of optimal solutions are clearly indicated. In the conceptual phase of the project, there were two main criteria that were desired to be explored: minimisation of MTOW; maximisation of total endurance or flight time. These two objectives conflict with each other since more fuel is needed in order for an aircraft to be airborne for longer periods of time. In the context of this particular project, the need for more fuel implies a larger hydrogen tank to store it. Therefore, as the endurance time is increased, it is expected that the UAV's MTOW increases as well. Being the project in its early stage, there was no knowledge about the possible trade-off solutions which contributed to a troublesome conversion of the two objectives to a single one. As a consequence, the problem was naturally posed as multi-objective and the trade-off between endurance and MTOW was studied not only to help decide upon which fuel cell and hydrogen tank to carry aboard but also to refine the mission profiling parameters that best suit the projects needs.

The MOOP in Eq.(2.1) has now the vector objective function $\boldsymbol{f}(\boldsymbol{x})$ composed by two objectives: the minimisation of MTOW and the maximisation of Endurance. Due to the way the algorithm chosen (NSGA-ii) works, the maximisation of the flight time was transformed to minimisation of $-$Endurance. As such, the objective function $\boldsymbol{f}$ can be written in vector for as

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{Bmatrix} MTOW(\boldsymbol{x}) \\ -Endurance(\boldsymbol{x}) \end{Bmatrix} \begin{bmatrix} \text{kg} \\ \text{Hours} \end{bmatrix} \tag{3.1}$$

with MTOW being the sum of structural weight, related to the airframe; propulsion system weight, which takes the motors, ESC and propeller/rotor weights into account; energy weight, which accounts for batteries and hydrogen needed to produce electric energy; other weights to account for cables, servos, avionics, payload and H_2 tank mass, expressed as

$$MTOW(\boldsymbol{x}) = m_{\text{s}} + m_{\text{prop. system}} + m_{\text{energy}} + m_{\text{other}} \quad \text{kg} \tag{3.2}$$

and Endurance calculated as the sum of the time needed to accomplish each mission segment,

$$Endurance(\boldsymbol{x}) = \sum_{i=1}^{n} t_i \quad \text{h},$$

where i represents each defined mission segment and N the total number of segments. For details about the mission profile and the procedure used to evaluate both objective functions consult (Alves et al., 2021, secs. 3 and 4).

To ensure that the design requirements were met and that the solutions are realistic, the following inequality constraint functions were defined:

$$\boldsymbol{g}(\boldsymbol{x}) = \left\{ \begin{matrix} MTOM(\boldsymbol{x}) - 25 \\ -E(\boldsymbol{x}) + 2.5 \\ b(\boldsymbol{x}) - 4.0 \\ P_{\text{Con. Mode}} - P_{\text{nominal}} \\ m_{\text{fuel}} - m_{\text{Tank}} \\ V_{\text{stall}} - V_{\text{Op}} + 8 \end{matrix} \right\} \begin{bmatrix} kg \\ h \\ m \\ W \\ g \\ kts \end{bmatrix} \leq 0\,, \tag{3.3}$$

where $g_1(\boldsymbol{x})$ sets the maximum allowable MTOW; $g_2(\boldsymbol{x})$ sets the minimum Endurance time; $g_3(\boldsymbol{x})$ limits the maximum allowable wingspan; $g_4(\boldsymbol{x})$ is used so that the maximum required power is less than the fuel cell nominal power when the aircraft is flying in forward flight mode; $g_5(\boldsymbol{x})$ ensures that the amount of fuel needed to perform the mission is within the tank capacity; and $g_6(\boldsymbol{x})$ establishes that the minimum operational speed is at least 8 kts above stall.

The optimisation problem inputs are six design variables that can be represented in vector format as $\boldsymbol{x} = [x_1, x_2, ..., x_6]^T \in \mathcal{D} \subseteq \mathbb{R}^6$. $\mathcal{D}$ represents the design space shown in Table 3.1.

The design variable x_1 sets the disk loading– ratio between the aircraft weight and total rotor area. This variable is crucial to determine the necessary power loading P/W for VTOL and Hover missions. The design variable x_2 represents the Wing Loading- ratio between the aircraft total weight and total wing area. This parameter is crucial for determining the power loading in the different forward flight mission segments. It cannot be arbitrarily chosen due to project requirements such as stall speed and maximum ceiling. The upper bound of this variable is given by the critical flight condition, i.e, the one that requires a smaller W/S. The determination of the allowable values for W/S is performed inside the Design Point routine of the developed numerical tool (Alves et al., 2021, fig. 7, sec. 4). The design variable x_3 represents the main wing aspect ratio, used to estimate the induced drag coefficient k and later to calculate the wing dimensions based on the MTOW and wing loading. The design variable x_4 is the loiter time. As this parameter is increased, both Endurance and MTOW are expected to increase. The design variable x_5 is used to calculate the W/S feasible region which impacts the obtained power-loading in forward flight mode whenever it is the critical condition. Finally, the design variable x_6 is the speed at which the UAV must travel in order to maximise its endurance.

The hydrogen tank mass was estimated based on a simple proportionality through a ratio that was assumed to be constant. This approach has some disadvantages: the value chosen depends on the size of tank being used; even with a correct ratio to estimate the required tank, since it will be procured instead of produced in-house, a study considering a set of available tanks will have to be carried out.

Table 3.1: Design variables for the multi-objective optimisation problem.

Design variable	Description	Lower bound	Upper bound	Units
x_1	Disk loading, W/A	100	350	N/m^2
x_2	Wing loading, W/S	100	250	N/m^2
x_3	Wing aspect ratio	5	12	-
x_4	Loiter time	2	∞	h
x_5	Stall speed	26	32	kts
x_6	Operational speed	30	45	kts

3.4 Convergence-Study and Parameter Tuning in the NSGA-ii Algorithm

As with all optimisation problems, it its necessary to initialise the algorithm, define termination criteria and customise some settings to better suit the particular set of problems that are being solved.

With the *Pymoo* framework, it is possible to initialise the algorithm with either a random population or with a user-defined population of solutions. Because problem knowledge is not yet known at this point, the author decided to use a random population with 200 individuals to initialise the algorithm.

The *Pymoo* framework offers several possible termination criteria: predefined number of function evaluations; predefined number of iterations/generations; change of performance metrics over time. In the first two mentioned criteria, the algorithm will stop whenever the user-defined number of either the function evaluations or the generations is reached. The last criterion uses a metric to determine the largest 'movement' between a solution and its closest neighbour across generations. When this 'movement' is below a certain threshold, the algorithm stops. This metric can be applied either to the design space $\mathcal{D}$ or to the objective space $\mathcal{Z}$, although the application in the latter is more challenging. For a detailed description of these metrics refer to Blank and Deb (2020).

The user is not required to use only one termination criterion, combinations of more than one are possible. In such cases, the algorithm will stop when at least one of them is met. The author defined a termination criteria composed of two criterion: a convergence criterion and a fallback to prevent the program from crashing when the algorithm is not able to fulfill the first (convergence criterion). The convergence criterion is defined with objective space $\mathcal{Z}$, design space $\mathcal{D}$ and constraint violation tolerances that must be satisfied simultaneously in order for it to be satisfied. When this is the case the Pareto-front will be said to have converged. As a fallback criterion, the maximum number of generations is also set.

With both criteria (convergence and fallback), the algorithm will be forced to stop either because it was able to find a converged Pareto-front or because the number of maximum iterations was reached.

In order to visualise the impact of the number of generations in the convergence of the Pareto-front, the optimisation problem was executed for a large number of generations (*n_gen* = 200), such that at each generation all the solutions $\boldsymbol{f}(\boldsymbol{x})$ in the Criterion Space $\mathcal{Z}_{\text{feasible}}$, i.e, the dominated and the non-dominated, are represented in a plot. Several plots corresponding to different generations are represented in Figure 3.1.

In each of the four plots, the blue circles represent non-dominated solutions and the red triangles correspond to dominated ones. All the plots have the same range, which was chosen based on the *ideal* and *nadir* vectors in the Criterion Space $\mathcal{Z}_{\text{feasible}}$, $\boldsymbol{\mathcal{Z}}^*$ and $\boldsymbol{\mathcal{Z}}^{\text{nad}}$ respectively.

In the 1st Generation, Figure 3.1(a), only two points can be observed even though the algorithm executed 100 function evaluations, one per individual in the population. Some of the solutions are not feasible and therefore are not represented, while others are feasible but they are outside range. At the 5th generation, Figure 3.1(b), it is already possible to see a greater number of solutions in the Criterion Space $\mathcal{Z}_{\text{feasible}}$ window. A few generations later, at generation 17, Figure 3.1(c), the Pareto-front can already be seen. However, the largest portion of points are still dominated ones. From Generation 70 onward, Figure 3.1(d), the final population has only non-dominated solutions (no red triangles are observed in the figures). Observing Figure 3.1(e), it is possible to notice that there are more solutions in the neighbouring region of the 21 kg mark. As such, with an increase in cost of $20 \cdot 200 = 4\,000$ function evaluations, the obtained diversity in the Pareto-front is higher. Finally, at generation 115, Figure 3.1(f), in the neighbouring region of the 21 kg there is a small increase in number of solutions compared to the previous figure. This accomplishment is at the expense of $25 \cdot 200 = 5\,000$ more function evaluations.

Obtaining Pareto-fronts with higher diversities is desirable, nevertheless, reducing the computational cost is also desirable. It is not possible to increase one without increasing the other. So, a compromise will be sought when defining the convergence criteria. Because the intention is to use the number of generations only when the algorithm cannot meet the convergence criteria, as a fallback, this number has to be greater than 90. The maximum number of generations is set to 150. With the fallback criteria defined, an adequate convergence criteria is now set so that the algorithm would stop between generations 70 and 90.

The author defined the tolerance in the design space $\mathcal{D}$, *x_tol*; the tolerance in the objective space $\mathcal{Z}$, *f_tol*; and the constraint violation tolerance, *cv_tol*. Other important parameters used to assess these tolerances such as *nth_gen* and *n_last* are used with their default values: $5,\ 30$, respectively. For details about the implementation and meaning of these parameters, refer to the *Pymoo* documentation (Blank, 2020). Several tolerances were tried when defining the convergence criteria. For each, the computational cost (Function Evaluations) is registered and compared. The results are summarised in the Table 3.2.

Through comparison between each criteria (column-wise), it is possible to assess that, of all the three parameters used to define the convergence criteria, the most sensible one is the *x_tol*, related to the 'movement' metric in the design space $\mathcal{D}$. A decrease in one order of magnitude will increase the

Table 3.2: Convergence Criteria, Cost Comparison

x_tol	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
f_tol	$1 \cdot 10^{-3}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
cv_tol	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
Last Generation	54	54	125	69
Cost	10 800	10 800	25 000	13 800

cost more than twofold (1st and 4th columns). On the other hand, the least sensible parameter was the *f_tol*, which is related to the 'movement' metric in the objective space $\mathcal{Z}$. A decrease in three orders of magnitude does not produce any increase in the computational cost (1st and 2nd columns). By using the last criteria, which is given in the last column in Figure 3.2, a compromise is possible: the computational cost is higher than the first two but the obtained Pareto-front is better; and when compared to the 3rd criteria the cost is significantly lower but the Pareto-front is not significantly worse.

3.5 Trade-Off Studies

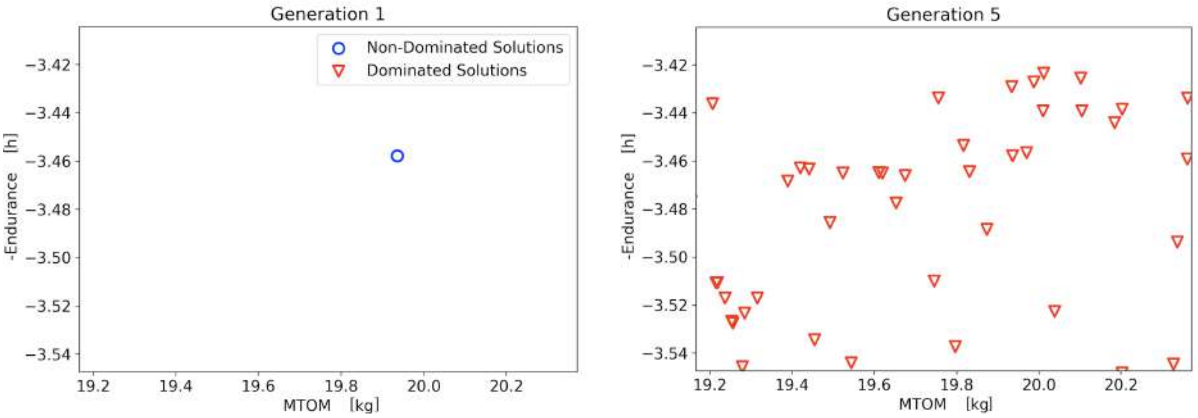
The results obtained in the conceptual phase were used to perform trade-off studies and help in the decision process. First, a brief description of the general UAV configuration is presented, later some trade-off studies that were performed are described, and finally, the configuration that is going to be used as baseline in the upcoming studies of Chapters 4 and 5 is briefly discussed.

3.5.1 UAV Concept and Estimated Parameters

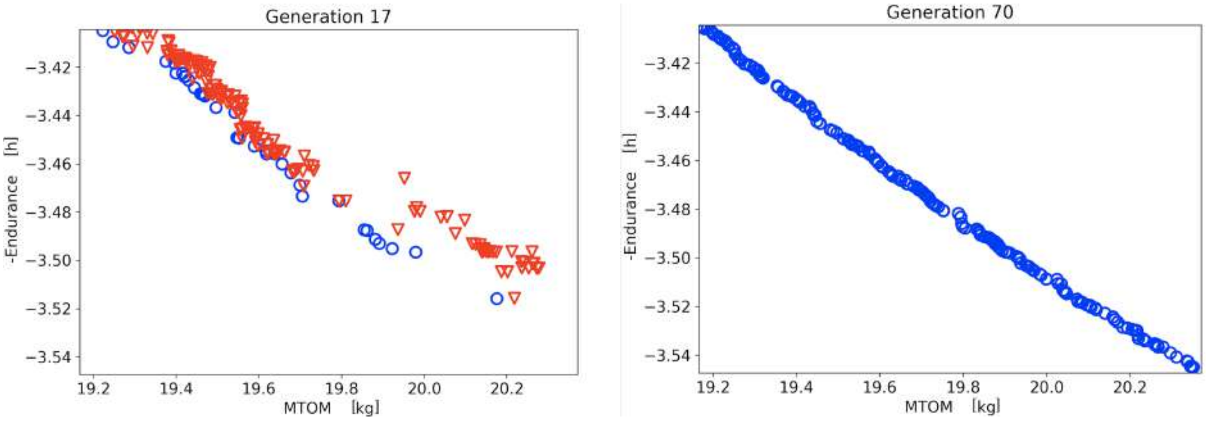
In the conceptual phase of the project, a lift+cruise configuration was chosen based on an Analytic Hierarchy Process (AHP) (Saaty & Vargas, 2012) with a set of criteria defined. Then, after some operational and aerodynamic considerations had been taken, a pusher configuration with an inverted V tail was selected.

The typical mission profile of the UAV starts with a vertical take-off. This is accomplished with the rotors which have an independent propulsive system powered by Li-Po batteries. Then, a transition between vertical and conventional fixed-wing flight occurs. The forward flight propeller is turned on and the vertical rotors will be shut down after sufficient forward speed is reached. This flight mode is powered by fuel cells which will convert the hydrogen stored in the gas tank into electric current. Once in fixed-wing mode, the UAV will perform two distinct climb segments with different gradients. The mission altitude will be reached and the vehicle will start the surveillance over the desired location. Afterwards, it will return back to the starting location where it will descend close to the ground. Then a landing circuit will be performed and the UAV will land vertically after the vertical rotors had been turned on.

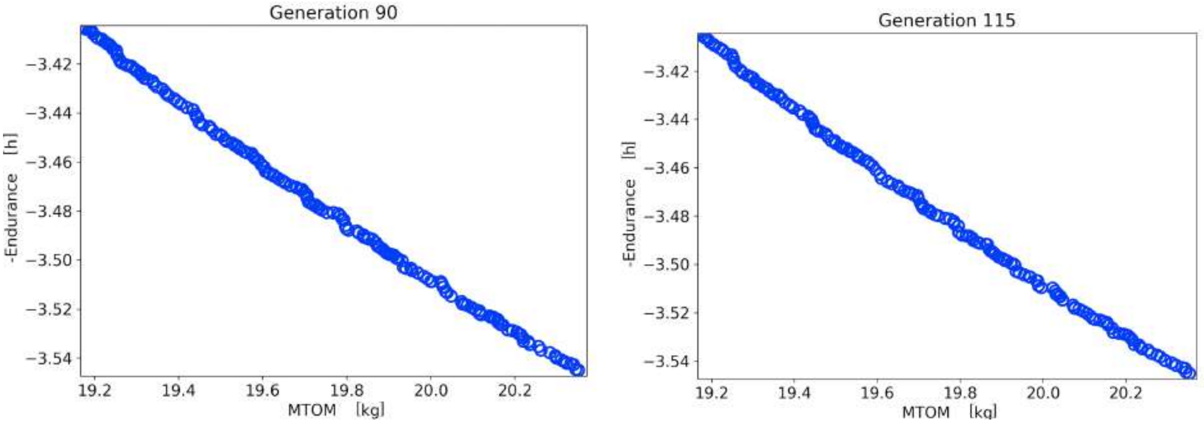
With the definition of the mission profile, together with a market search on similar UAVs and fuel cells, some important parameters to size the vehicle and predict its performance were estimated. Accordingly, a fuel cell with 800 W of nominal power was chosen. These parameters are listed in Table 3.3, for details refer to Alves et al. (2021, sec. 5).



(a) After gen. 1 (b) After gen. 5



(c) After gen. 17 (d) After gen. 70



(e) After gen. 90 (f) After gen. 115

Figure 3.1: Solutions obtained at different generations

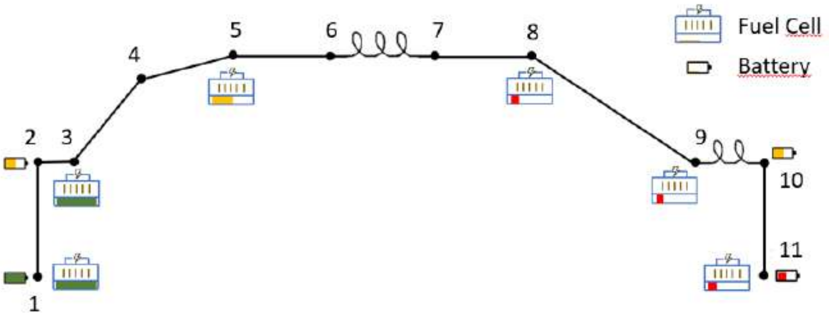


Figure 3.2: Typical mission profile, retrieved from Alves et al. (2021)

Table 3.3: UAV initial parameter values retrieved from Alves et al. (2021)

Parameter	Value	Parameter	Value
Airplane base drag coefficient - C_{D_0}	0.04	Forward motors power-to-mass ratio	3.5
Oswald efficiency factor - e	0.75	VTOL motors power-to-mass ratio	4.5
Maximum lift coefficient - $C_{L_{max}}$	1.3	Batteries safety factor	0.3
Structural factor	0.35	Batteries specific energy density	160 Wh/kg
Propeller efficiency - η_{pr}	0.65	Electric system efficiency	0.85
Induced power factor - k_i	1.2	Hydrogen-to-tank mass ratio	0.035
Rotor solidity - σ	0.10	Hydrogen low heating value	120 MJ/kg
Rotor profile drag coefficient - $C_{d_{0(rotor)}}$	0.012	Avionics power requirement	80 W
Fuel cell efficiency	0.4-0-5	Avionics and cabling mass	2.5 kg

3.5.2 Concept Optimisation

With the software developed in Section 3.1, the design variables with the respective upper and lower bounds defined in Table 3.1, the termination criteria set in Section 3.4 and the parameters defined in Table 3.3, it was possible to run the NSGA-II algorithm with the intent of solving the optimisation problem of Section 3.3. The initial sizing iterative solver, which is responsible to evaluate both the MTOW and constraint functions, was not able to satisfy the absolute tolerance defined within the 50 maximum iterations for any individual of the random initial population. As a result, an error occurred at generation 0 and the optimisation problem was aborted. It was not the NSGA-II algorithm that had failed to find the Pareto-front, instead, it was the actual analysis that failed.

To understand why the solver was having problems to converge the MTOW, some intermediate results were analysed in detail. It was found that in the Design Point calculations (Figure 3.3), the power-loading P/W obtained for the conventional flight segments of the mission profile was too high for the motors power-to-mass ratios estimated (Table 3.3). As a consequence, the solver was not able to find a solution as no amount of weight of the propulsive system was capable of providing enough power. Lower values for P/W in the different segments are desired since they will imply that lighter motors and lower H_2 consumption rates could be achieved. Minimising the fuel consumption for a given flight condition would be equivalent to maximising the amount of time that the vehicle can be airborne.

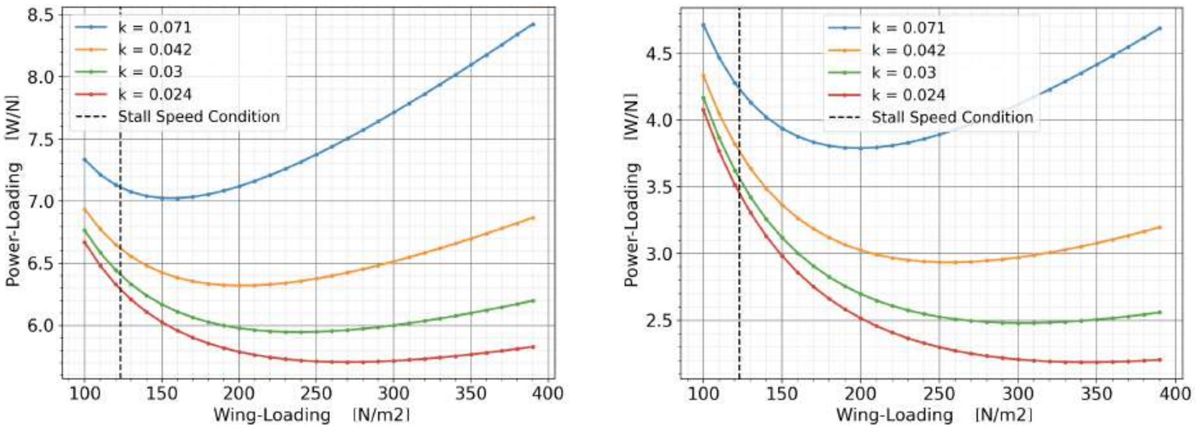
P/W curves as function of wing-loading W/S for the 1st climb and the cruise segments of the mission profile are represented in Figure 3.3(a) and 3.3(b), respectively. A vertical black dashed line represents the maximum allowable wing-loading value which is imposed by the stall speed constraint V_{stall} (Table 1.1).

It can be observed that the minimum value of every P/W curve is not attainable without violation of the stall speed condition. Moreover, it is also clear that the value of P/W obtained for one segment is also affected by the aerodynamic lift-induced drag coefficient k. As an example, for the climb segment with $W/S = 120\ N/m^2$ the value of P/W ranges from 6.3 (for lowest k) to $7.1\ W/N$ (for highest k). The lift-induced drag coefficient k is estimated according to the Prandtl's Classical Lifting-Line Theory (Anderson, 2017, sec. 5.3),

$$k = \frac{1}{\pi \cdot AR \cdot e}, \tag{3.4}$$

with e being the Oswald efficiency factor and AR being the wing aspect ratio. With $V_{\text{Stall}} = 25$ kts $\Rightarrow k \geq 0.071$ for given values of AR and e. Visually, it means that P/W curve of the UAV being designed corresponds to the blue ones on both plots (Figure 3.3). A low V_{Stall} implies that the highest allowable value for W/S will also be small. As a result, for two UAVs with the same weight W, the wing area S of the one with lower V_{Stall} has to be bigger. Since the wingspan $b(\boldsymbol{x})$ is constrained in the posed optimisation problem, the larger the area the lower the wing aspect ratio. Therefore, low stall speeds imply low wing aspect ratios AR, which in turn imply high values for the lift-induced drag coefficient k according to Eq.(3.4). From the power-loading plots in Figure 3.3, it can be seen that as k increases the corresponding P/W also increases.

At this stage a design decision had to be made: either the wingspan constraint $b(\boldsymbol{x}) - 4.0 \leq 0$ or the stall speed condition $V_{\text{Stall}} - 25 \leq 0$ had to be relaxed. The decision makers opted for the second option motivated by the fact that the UAV would have VTOL capability which allows it to fly with speeds V lower than the stall in conventional fixed wing mode, $V \leq V_{\text{Stall}}$ by using the vertical flight rotors to generate lift.



(a) Power-loading for the 1st climb segment

(b) Power-loading for the cruise segment

Figure 3.3: Power-loading for different mission segments

After solving the same optimisation problem with the relaxed stall speed, $V_{\text{Stall}} = 30\,$kts, a Pareto-front with the non-dominated solutions was obtained. A set of solutions was discussed and it was concluded that although all constraints are respected, g_4 included, refer to inequality (3.3), there was not much margin between the maximum power required to the fuel cell system and its actual nominal power (refer to Alves et al. (2021, sec. 6) for details). Due to the uncertainty in the estimation of the base drag coefficient, C_{D_0} and the consequences that a small increase might have had in the whole project, it was decided that new solutions with bigger power margins were to be sought. To increase the power margin one can either increase the total power that can be delivered by the source, i.e upgrade the fuel cell, or sacrifice some performance in the proposed concept and, thus, lower the maximum power required keeping the same nominal power.

One of the estimated parameters with higher uncertainty is the hydrogen tank which was estimated as

$$m_{tank} = \frac{1}{c} \cdot m_{H_2} \qquad (3.5)$$
$$\text{with} \quad 0 < c < 1,$$

where $c = \frac{m_{H_2}}{m_{tank}}$ is assumed constant and estimated according to a market search on hydrogen gas tanks (P. Silva, 2021); and m_{H_2} is the required fuel mass needed to fly the defined mission profile.

In an attempt to increase the accuracy of the results obtained, the hydrogen tank mass estimation is replaced with commercially available tanks. Based on previous results, two tanks are considered: a larger tank with dry mass of 4.3 kg and a capacity for 175 g of H_2; and a smaller one of 3.3 kg with 148 g of H_2 capacity.

Upgrading the fuel cell implies an upgrade to the hydrogen tank as well. A cell with higher nominal power is heavier and will have lower efficiency when operating at lower power outputs, thus it is expected that in cruise there will be a decrease in efficiency which in turn implies an increase in the amount of fuel needed to perform the same mission. The 800 W Fuel Cell is used together with the smaller tank while the 1300 W Fuel Cell is used with the larger.

On the other hand, with the same 800 W fuel cell, it is also possible to increase the power margin by lowering the maximum power consumption in level flight. As such, for the lightest configuration, two additional design variables were added to the optimisation problem, stall speed and operational speed, x_5 and x_6 in Table 3.1, respectively; and the constraint g_5 added to guarantee that the mass of hydrogen can be held in the tank, inequality (3.3). For the heavier configuration, only the constraint g_5 was added, with $m_{\text{tank}} = 175\,$g, but no additional design variables were considered since no performance sacrifice is intended.

Both problems have the same convergence criteria set previously in Section 3.4.

A superposition of the obtained sets of Pareto-optimal solutions, one for each problem, is represented in Figure 3.4, where it can be observed that both Pareto-fronts, represented in blue and orange lines, have a steep slope. This implies that small variations in MTOW imply large changes in the total Endurance time. With the lighter configuration (approximately 18.5 kg), it is possible to achieve between

$3.53 - 4.66\,$h of total flight time with an increase of $34\,$g of H_2. This represents an increase of 32% in Endurance with an increased cost of 0.18% in terms of weight.

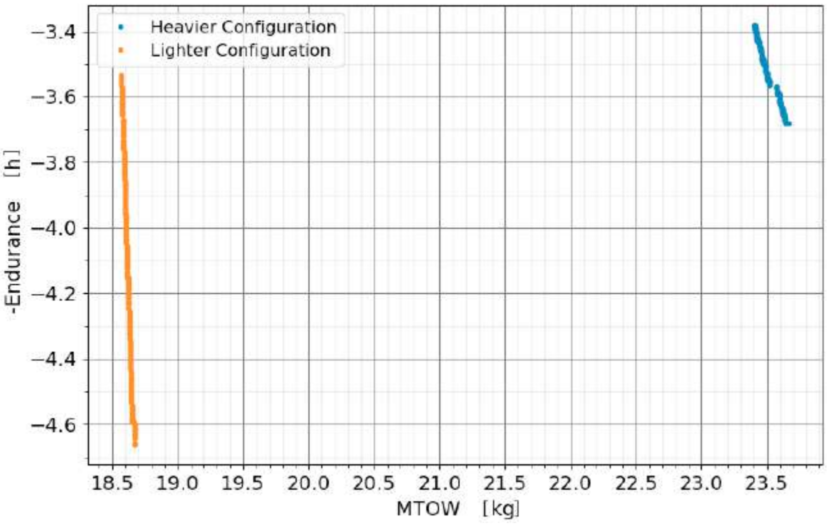


Figure 3.4: Comparison of the two different sets of Pareto-optimal solutions

With the heaviest configuration (approximately 23.5 kg) it is possible to fly $3.38 - 3.68\,$h with an increase of $16\,$g of H_2, which represents an increase of 8.9% of endurance at a cost of 0.07% more weight.

The reasons why the total endurance time is extremely sensitive to wieght is related to:

1. High specific energy of H_2;
2. Discretisation of the H_2 tank mass
3. Low power consumption during the loiter segment of the mission profile.

As stated in Section 1.3, H_2 is one of the fuels with higher specific energy with $e \approx 120\,$MJ/kg (Edwards et al., 2008). Therefore, small increases in the mass of H_2 to be carried imply high gains in energy that will be used to power the vehicle's electrical motors for longer periods of time. Secondly, due to the discretisation of the tank masses, there is no direct correlation between the amount of fuel needed to perform the mission and the weight of the tank. It is only ensured that the amount of H_2 can be stored inside the respective tank, as imposed by constraint g_5. As such, instead of one continuous Pareto-front, several different fronts are obtained, one for each tank where the only variable that affects the weight of the UAV is the mass of the fuel.

Lastly, because the UAV is being designed for high efficiency in the loiter segments, it will have low power consumption during the majority of the flight, which also contributes for improving the ratio between the total endurance time and the amount of required fuel.

Because there are considerable gains with respect to endurance at a small penalisation in the UAV's MTOW and there is no advantage, from the flight safety point of view, in saving between 16 and 34 g of H_2, it was decided that both tanks would be completely full at take-off.

To choose the individual solution between the two possible configurations a comparison between the two is made. Since the heavier configuration has a smaller endurance time when compared to the lighter configuration, and the goal of the trade-off studies was to maximise endurance and minimise MTOW simultaneously, the set of optimal solutions corresponding to the heavier configuration were discarded. Although the heavier configuration has more fuel, the increase in weight due to the upgrade in the H_2 tank will require a larger wing area S to obtain the same optimal wing-loading W/S found by the optimisation algorithm during the Design Point calculations. Due to the wingspan constraint g_3, inequality (3.3), this will imply a decrease of aspect ratio, as pointed previously when the stall speed effect was discussed, this will increase the estimated induced drag coefficient k according to Eq.(3.4). As a result, the power required in forward flight modes will increase: both the ratio P/W and W itself are higher. An increase in the required power will imply higher consumption of H_2 and less endurance time can be obtained. To sum up, it is concluded that if the wingspan constraint g_3 is to be respected then there is no benefit in scaling up the UAV.

By giving more freedom to the optimiser through the addition of the x_5 and x_6 design variables, it was possible to decrease the power required to the fuel cell to approximately 560 W without violating the constraints. As a result, the margin between the maximum power required in flight to the fuel cell and its nominal value was increased as desired.

3.6 Final Configuration of the Conceptual Phase

The Pareto-optimal solution corresponding to the lighter configuration was analysed and, with the estimated power required and propulsion system mass, the batteries, motors, propellers and rotors commercially available were selected (P. Silva, 2021). As expected, the total UAV weight increased when the mass estimates were replaced by the actual components since all had to be rounded up to the nearest commercially available, to guarantee that the propulsion system produces sufficient thrust to the aircraft.

The general parameters of the final configuration of the conceptual phase which is going to serve as baseline in subsequent studies are listed in Table 3.4[1]

Table 3.4: General characteristics of the baseline UAV, retrieved from (Alves et al., 2021, sec. 7)

Description	Value	Description	Value
Stall speed	28 kts	Operational Speed	38 kts
Wingspan	4 m	Propulsion system mass	2.60 kg
Wing area	1.372 m^2	Energy system mass	7.23 kg
Payload	0.9 kg	Structural mass	7.56 kg
Endurance	3h20min	MTOW	21.6 kg

With the UAV's MTOW and estimated wing area, a preliminary design of both wing and tail was accomplished with the use of low-fidelity analysis software (Coelho, 2021).

---

[1] The difference in payload was due to a change in the project requirements. This value corresponds to the mass of the Gimbal chosen

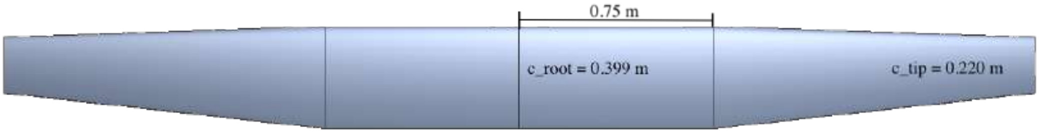


Figure 3.5: Wing Planform, retrieved from Coelho (2021)

The wing is composed of three panels: One at the centre and two at the tips. The centre portion of the wing is rectangular with a constant chord of 0.399 m and span of 1.5 m; while each of the outboard panels are tapered with $\lambda = 0.55$, thus the chord decreases linearly from the ends of the central panel until each tip. The total wingspan of the wing is 4.0 m. The SG6042 airfoil (Giguère & Selig, 1998) is used in the whole wing.

The tail is an inverted V with a dihedral angle of -45 deg, constant chord of 0.255 m and a projected span of 1.43 m. It is connected to the wing's central panel by a double boom. The selected airfoil was the symmetric NACA 0008.

Chapter 4

A Multidisciplinary Approach To Aircraft Design

Aircraft design involves knowledge and analysis of different disciplines, namely, aerodynamics, structures, controls and propulsion (Raymer, 1992). The analysis of each discipline cannot be done independently of one another due to the interactions that exist between them. For example, the aerodynamic loads that occur at the wing will deflect its shape. The deflected shape, different from the original one, will be subjected to different aerodynamic loads. In order to find the true shape of the wing and the resulting loads in flight it is necessary to couple these two disciplines.

This chapter begins with the reasons that led to the choice of using classical optimisation algorithms, followed by a brief overview of their working principles. Then, a description of both the multidisciplinary model that was used as well as the framework upon which it was built is provided. Afterwards, the wing and tail geometry implementation on the model is described. Then, motivated by some difficulties related to the geometry of the baseline model and the existent features in the software used, some code developments are presented.

4.1 Classical Optimisation Algorithms

In Section 2.3, the advantages and disadvantages of classical and heuristic methods was provided. The main disadvantages of the classical methods are that they are dependent upon the initial solution $\boldsymbol{x_0}$, they cannot handle discrete search spaces and they require a continuous objective function. However, if the mentioned requirements are satisfied, they are very powerful in the search for the optimal, exhibiting much greater efficiency than heuristic methods who tend to have poor convergence rates (Martins & Ning, 2022).

Classical optimisation algorithms can be divided into gradient-based and gradient-Free algorithms.

Gradient-based methods use the derivative information of the objective function f to guide the search, they can be further categorised in line search or trust-region algorithms.

Line-search methods use the derivative information at the current iterate $\boldsymbol{x}^{(k)}$ to compute a search direction and then find a step length α that approximately minimises f along that direction. After finding the new point $\boldsymbol{x}^{(k+1)}$ a new search direction and step length are calculated and the process repeats until a termination criteria is met (convergence achieved or number of max iterations reached, for example).

Trust-region Algorithms also use the derivative information of f at an iterate $\boldsymbol{x}^{(k)}$ but instead of finding a search direction these methods use it to construct a model m_k of f such that inside a region defined around $\boldsymbol{x}^{(k)}$, the model has a similar behaviour to f. Afterwards, the minimum of m_k is found and f is evaluated at that point. The process repeats until a termination criteria is met. In both class of algorithms the derivatives of f are needed. Gradient-free methods require only information about $f(\boldsymbol{x})$ and the problem constraint functions $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$, which make them useful when computing derivatives is not possible.

The optimiser used in the next chapter to solve the nonlinear optimisation problem, the Sequential Least Squares Program (SLSQP) (Kraft, 1988), falls under the category of Line-search methods.

4.2 OpenAeroStruct, A Lightweight Aerostructural Optimisation Tool

OpenAeroStruct[1] is an open-source low-fidelity aerostructural analysis and optimisation tool that was developed in NASA's OpenMDAO[2] framework (Jasa et al., 2018). It uses the Vortex-Lattice-Method (VLM) together with Finite-Element-Analysis (FEA) for the analysis of lifting surfaces.

OpenMDAO (Gray, Hwang, Martins, Moore, & Naylor, 2019) is a high performance open-source software, written in Python, which was developed by NASA for multidisciplinary design, analysis and optimisation (MDAO). It is primarily designed to use gradient-based techniques and allows efficient and accurate calculation of the coupled model derivatives. It provides a modular environment to facilitate the integration of the different discipline analysis into the multidisciplinary model. One of the characteristics that make this framework different from others is how it defines each discipline as an implicit function so that it can later use the unified derivatives equation to compute the derivatives of the coupled model. This approach is named Modular Analysis and Unified Derivatives (MAUD) Architecture, for a comprehensive description refer to Hwang and Martins (2018).

OpenMDAO has four fundamental classes to build models. These are the *Component*, *Group*, *Driver* and *Problem*. An example of a generic OpenMDAO Model can be found in Figure 4.1. Components in OpenMDAO are used to create the lowest-level of functionality. A component can represent a whole discipline or just part of it. Groups contain other components, other groups or a mix between the two. Their main purpose is to package components together, create better organised namespaces and facilitate the use of solvers. The top-level group which contains all the other groups and components hierarchically organised is the *model*. The *Driver* class is used to define the algorithms that will iteratively call the model, like optimisation algorithms. Finally, the *Problem* class is used to construct the container of both the model (with all the instances of the other groups, components and their hierarchy) and the driver.

---

[1] version used: 2.3.0

[2] version used: 3.6.0

Each instance of the *Problem* class contains one instance of the *Driver* class.

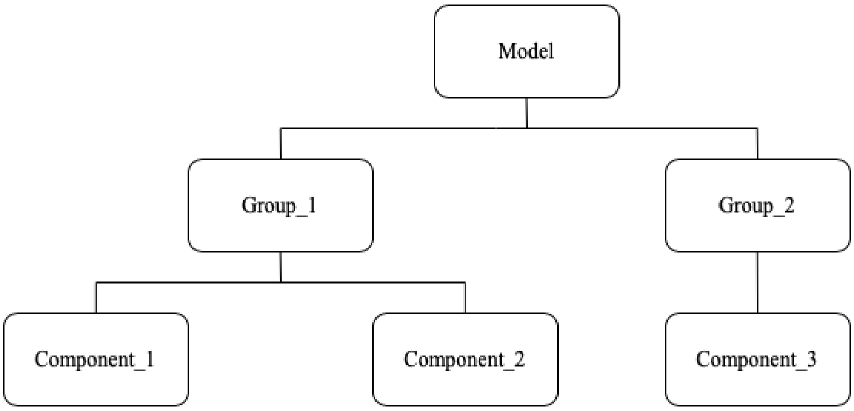


Figure 4.1: OpenMDAO generic model

OpenAeroStruct uses the mentioned classes to construct its models. Since it is built on top of OpenMDAO, the task of computing the total model derivatives is simplified. The tool defines the partial derivatives of each component output or residual with respect to its inputs and the NASA's framework solves the unified equation to find the coupled model derivatives. The partial derivatives computation can be performed using finite differences, complex-step, or analytic methods.

VLM is used to evaluate the aerodynamic loads acting on lifting surfaces. The implementation of the method in OpenAeroStruct was based on Anderson (2017).

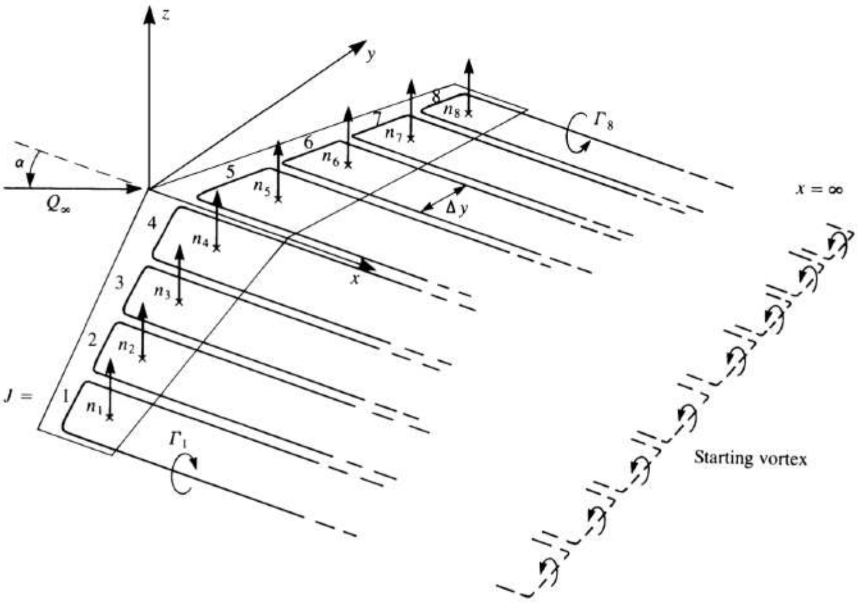


Figure 4.2: Illustrative VLM model with multiple horseshoe vortices along the span, retrieved from: (Katz & Plotkin, 2001)

Given a structured mesh, which defines a lifting surface like a wing or tail, a vortex system is constructed. The vortex system is comprised of horseshoe vortices, each consisting of two trailing edge vortices in the free-stream direction and a bound vortex in the spanwise direction. Several horseshoe vortices are imposed along the span and chordwise directions. An illustrative representation of the model is given in Figure 4.2, where multiple horseshoe vortices were imposed in the spanwise direction. The reference axis shown was used throughout the present work.

A vortex filament induces a flow field in its vicinity, the strength of it is its circulation, responsible for generating lift on that surface. The Biot-Savart Law relates the induced velocity on the flow field at an arbitrary point P, caused by a segment $d\boldsymbol{l}$ of a vortex filament with its circulation Γ,

$$d\boldsymbol{V} = \frac{\Gamma}{4\pi} \cdot \frac{d\boldsymbol{l} \times \boldsymbol{r}}{|\boldsymbol{r}|^3}, \tag{4.1}$$

where Γ represents the circulation strength and $\boldsymbol{r}$ represents the distance between the vortex and P where the flow field velocity is being measured. Integration over a semi-infinite straight vortex filament results in

$$V = \frac{\Gamma}{4\pi h}, \tag{4.2}$$

with h being the distance between P and the start point of the vortex filament.

In each panel, a control point is defined and a flow tangency condition is enforced so that the velocity component normal to the panel is 0. By imposing this flow condition at all the control points for each horseshoe vortex, the linear system is obtained,

$$\boldsymbol{A\Gamma} = -\boldsymbol{V}_\infty \cdot \boldsymbol{n}, \tag{4.3}$$

where $\boldsymbol{A}$ is the aerodynamic influence coefficients matrix, $\boldsymbol{V}_\infty$ is the velocity in the non-disturbed field and $\boldsymbol{n}$ is the normal to the panel. Solving the linear system allows the determination of the circulation strength vector $\boldsymbol{\Gamma}$. With it, the aerodynamic forces acting on each individual panel can be computed using

$$\boldsymbol{F}_i = \rho \Gamma_i \left(\boldsymbol{V}_\infty + \boldsymbol{v}_i\right) \times \boldsymbol{l}_i, \tag{4.4}$$

where $\boldsymbol{v}_i$ is the induced velocity at the center of the bound vortex, and $\boldsymbol{l}_i$ is the bound vortex vector. After computing the panel forces, the lift and drag are obtained by decomposing it in the freestream and the perpendicular to the freestream direction.

The structural model consists of spatial beam-bar elements with 6 degrees of freedom (DOFs) per node, which results in a total of 12 DOFs per element. The 6 DOFs per node are three translation displacements in the x, y and z directions and three rotations with respect to the x, y and z axes. The stiffness matrix for a single element is

$$[k]_e = \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 & 0 & -k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12k_2^z & 0 & 0 & 0 & 6k_2^z l & 0 & -12k_2^z & 0 & 0 & 0 & 6k_2^z l \\ 0 & 0 & 12k_2^y & 0 & -6k_2^y l & 0 & 0 & 0 & -12k_2^y & 0 & -6k_2^y l & 0 \\ 0 & 0 & 0 & k_3 & 0 & 0 & 0 & 0 & 0 & -k_3 & 0 & 0 \\ 0 & 0 & -6k_2^y l & 0 & 4k_2^y l^2 & 0 & 0 & 0 & 6k_2^y l & 0 & 2k_2^y l^2 & 0 \\ 0 & 6k_2^z l & 0 & 0 & 0 & 4k_2^z l^2 & 0 & -6k_2^z l & 0 & 0 & 0 & 2k_2^z l^2 \\ -k_1 & 0 & 0 & 0 & 0 & 0 & k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12k_2^z & 0 & 0 & 0 & -6k_2^z l & 0 & 12k_2^z & 0 & 0 & 0 & -6k_2^z l \\ 0 & 0 & -12k_2^y & 0 & 6k_2^y l & 0 & 0 & 0 & 12k_2^y & 0 & 6k_2^y l & 0 \\ 0 & 0 & 0 & -k_3 & 0 & 0 & 0 & 0 & 0 & k_3 & 0 & 0 \\ 0 & 0 & -6k_2^y l & 0 & 2k_2^y l^2 & 0 & 0 & 0 & 6k_2^y l & 0 & 4k_2^y l^2 & 0 \\ 0 & 6k_2^z l & 0 & 0 & 0 & 2k_2^z l^2 & 0 & -6k_2^z l & 0 & 0 & 0 & 4k_2^z l^2 \end{bmatrix} \tag{4.5}$$

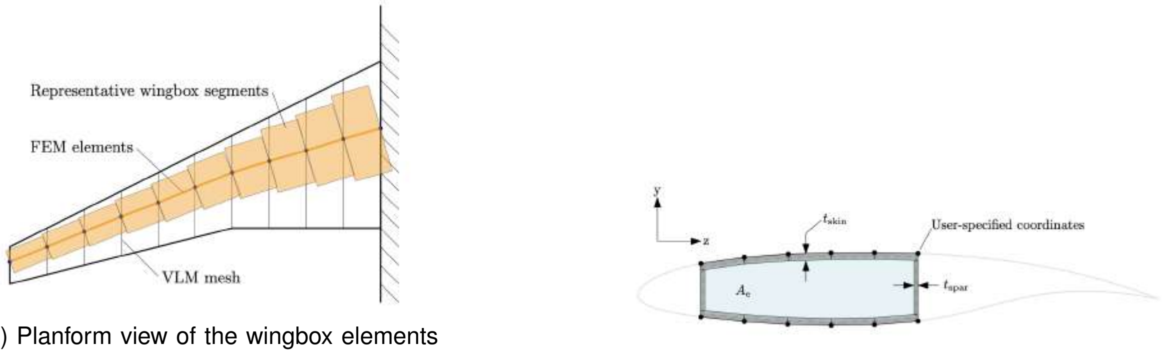
where $k_1 = \frac{EA}{L}$, $k_2^z = \frac{EI_z}{L^3}$, $k_2^y = \frac{EI_y}{L^3}$, $k_3 = \frac{GJ}{L}$ and E is the Young's modulus, A is the beam-bar cross sectional area, L is the length of the element, G is the shear modulus, J is the polar moment of inertia and I_s are the second moments of area about each of the axes. OpenAeroStruct first assembles the global stiffness matrix $\boldsymbol{K}$ using each element $[k]_e$ and then, solves the linear system

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{4.6}$$

where $\boldsymbol{u}$ is the vector of displacements at the nodes and $\boldsymbol{f}$ are the forces and moments acting on them.

In order for OpenAeroStruct to calculate the stiffness matrix for all the elements that the lifting surface is discretised into, some important geometric properties are required: A, I and J, the cross sectional area of the element, the second moment of area about two reference axes and the polar moment of inertia, respectively. These parameters are calculated in a plane perpendicular to the element. Two cross-sectional shapes can be used to calculate the mentioned parameters: a tubular one and a wingbox (Chauhan & Martins, 2019; Jasa et al., 2018).

In the present work the cross-section chosen was the wingbox for both lifting surfaces, i.e wing and tail. The wingbox structure consists of the upper and lower skin, which are responsible to resist bending loads and the front and rear spars which are responsible to resist shear loads. Both skins and spars also provide a closed loop for the torsional shear flow and, therefore, are effective in resisting torsional loads (Chauhan & Martins, 2019). A representation of this structural model is given in Figure 4.3.



(a) Planform view of the wingbox elements imposed on the VLM mesh

(b) Wingbox cross-section

Figure 4.3: Wingbox model, retrieved from Chauhan and Martins (2019)

The structure is modeled by providing the airfoil coordinates of the portion that is occupied by the wingbox and by specifying the thickness of both spars and both skins, t_{spar} and t_{skin} in Figure 4.3(b). The same skin thickness is used for both the upper and lower skins and the same spar thickness for both the front and rear spars. Although it is possible to vary these parameters along the wingspan, it is not possible to use different airfoil coordinates along the wingspan, neither it is possible to change this portion during the optimisation process. The larger the portion of the airfoil used, the greater will be the distance between the front and rear spars. E.g, in case the user decides to use the coordinates of $10-60\,\%$ of the chord, the front spar is positioned at 10% of the chord, from the leading edge, and the rear spar at 60%. The user-defined coordinates are scaled at each section with an average of the chord lengths of the two nodes that enclose the wingbox and with the thickness-to-chord-ratio t/c variable that must also be set by the user. The VLM mesh with the FEM elements and the wingbox segments are represented in Figure 4.3(a).

The load and displacement transfer scheme used is both consistent and conservative (Jasa et al., 2018; Martins, Alonso, & Reuther, 2005). In order to transfer the aerodynamic loads from the VLM mesh into the FEM elements, the location of the shear centre along the chord for each wingbox is needed. To maintain simplicity, this location is estimated as an average of the location of the spars weighted by their respective area. For a complete description of the simplified wingbox model refer to (Chauhan & Martins, 2019).

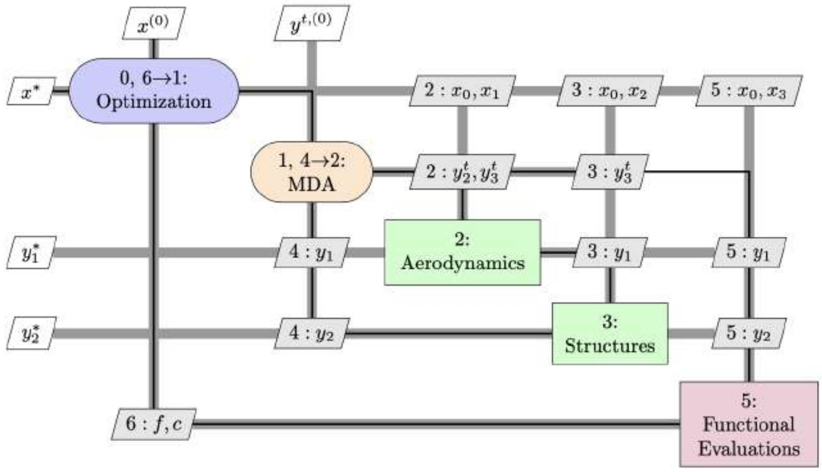


Figure 4.4: XDSM diagram of the default aerostructural optimisation problem in OpenAeroStruct, retrieved from Jasa et al. (2018)

To perform MDO, both disciplines (Aerodynamics and Structures) as well as their coupling have to be considered. The extended design structure matrix (XDSM) diagram (Lambe & Martins, 2012) representative of the aerostructural optimisation problem implemented in OpenAeroStruct is shown in Figure 4.4.

The aerodynamics group receives a mesh and outputs aerodynamic loads and the structures group receives the loads and outputs the structural displacements. Before the objective function and constraints can be solved, it is required that the solution of the coupled system is found so that the aerodynamic loads and the structural displacements in both groups are the same. A solution of the coupled system is found by solving

$$\boldsymbol{\mathcal{R}}_i(\boldsymbol{y}, \boldsymbol{y}^t) = 0, \tag{4.7}$$

where $\boldsymbol{y}$ and $\boldsymbol{y}^t$ represent the state and the target state variables, respectively. Different solvers can be used to converge the coupled system, such as Newton or fixed-point iterations.

OpenAeroStruct simplifies the process of transferring the aerodynamic and displacement loads by using the same spanwise discretisation for both the FEM elements and the VLM panels.

4.3 Wing and Tail Geometry Definition in OpenAeroStruct

OpenAeroStruct can model any number of given lifting surfaces. To do so, the user must define each surface as a set of key-value pairs. Each pair contains information about some wing-related parameter like aerodynamic twist, spar thickness and its shape, defined as a mesh of nodes. To define the mesh the user can use one of the default surfaces or define a custom mesh. There are two main default lifting surfaces: a rectangular and the Common Research Model (CRM) (Rivers, 2019). The mesh consists of a multidimensional array of dimension $i \times j \times k$, where the index i identifies the chordwise position of the node, j locates the node in the spanwise direction and k stores its 3 coordinates. It is initially defined by the user and stored inside an OpenMDAO *Group* called geometry which in turn is part of another *Group* with a user-defined name. Since the author intends to model both wing and tail, two *groups* are instantiated, each with a geometry *group* that contains a mesh. A graphical representation of this hierarchy can be observed in the N2 chart of Figure 4.5 where the wing *group* is expanded. Below it, is the tail which has a similar composition.

There are several built-in functions to manipulate the original geometry of the mesh that can also be used as design variables during optimisation, for a complete list refer to "OpenAeroStruct Documentation-Geometry Creation and Manipulation" (2018). Using the rectangular surface together with some of these geometry manipulation functions, it is possible to define simple trapezoidal planforms. Even though the baseline UAV has a trapezoidal shape, its inboard sections (which corresponds to 37.5% of the semispan) are rectangular which is hard to achieve using the available tools. To overcome this issue, a custom mesh is defined instead of using one of the default surfaces:

1. Create two rectangular meshes using OpenAeroStruct's built-in functions. One of them is going to be the inboard segment which will remain rectangular and the other one will be the outboard segment where taper is applied;
2. Apply taper to the mesh so that the leading edge remains perpendicular to the fuselage;
3. Add an offset to the outboard wing segment;
4. Join the two wing segments together.

First, two rectangular meshes were constructed using the helper function, *generate_mesh*, which receives as input a dictionary with user-defined options and outputs a rectangular mesh. One of the

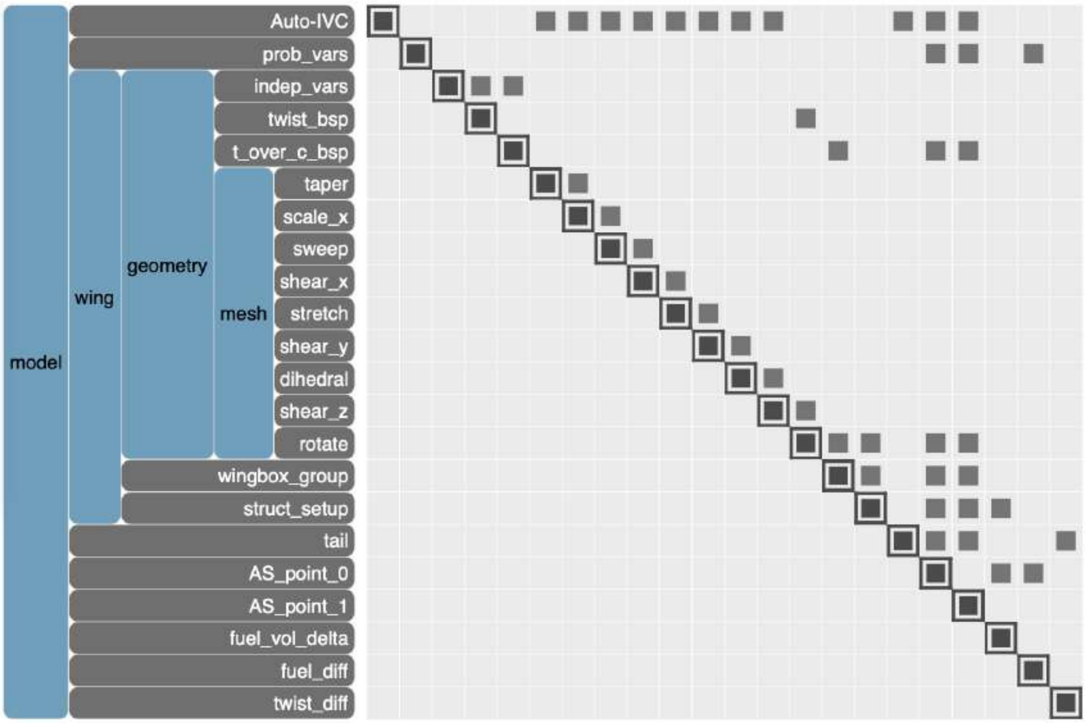


Figure 4.5: Aerostructural model with wing *group* expanded

required user-defined options is the number of chordwise and spanwise nodes to discretise the mesh into different panels. Different levels of fidelity can be obtained by varying these two parameters. In Chapter 5, an analysis is conducted to select the adequate number of chordwise and spanwise nodes for each of the wing segments.

The available *taper* function alters the chord linearly to produce a tapered wing. Because the taper is applied around the quarter-chord line, it could not be used as is to define our tapered segment whose leading edge remains perpendicular to the fuselage. To overcome this difficulty, a new function was implemented to apply the desired taper.

First, taper is applied to the quarter-chord line using the already built-in function, then the angle θ between the leading edge and the y direction is calculated. Finally, the built-in *sweep* function is executed with the tapered mesh and $-\theta$ given as inputs. The *sweep* function simply applies a shear deformation in the x direction to all points of a given mesh based on their distance to the $y = 0$ plane and the given angle of sweep θ. Because all the nodes at a distance $y = b$ will suffer the same x translation the chord of the wing at each section will not be altered. Therefore, the chord length distribution along the span defined by the *taper* function will be the same but the leading edge will remain perpendicular to the fuselage as desired.

At this stage the two wing segments are already defined but they both start at $y = 0$. As such, the outboard segment suffers an offset in the y direction.

Finally, the two meshes are joined together. Since the nodes that define the root of the outboard segment are the same as the ones that define the tip of the inboard segment, the duplicates are omitted.

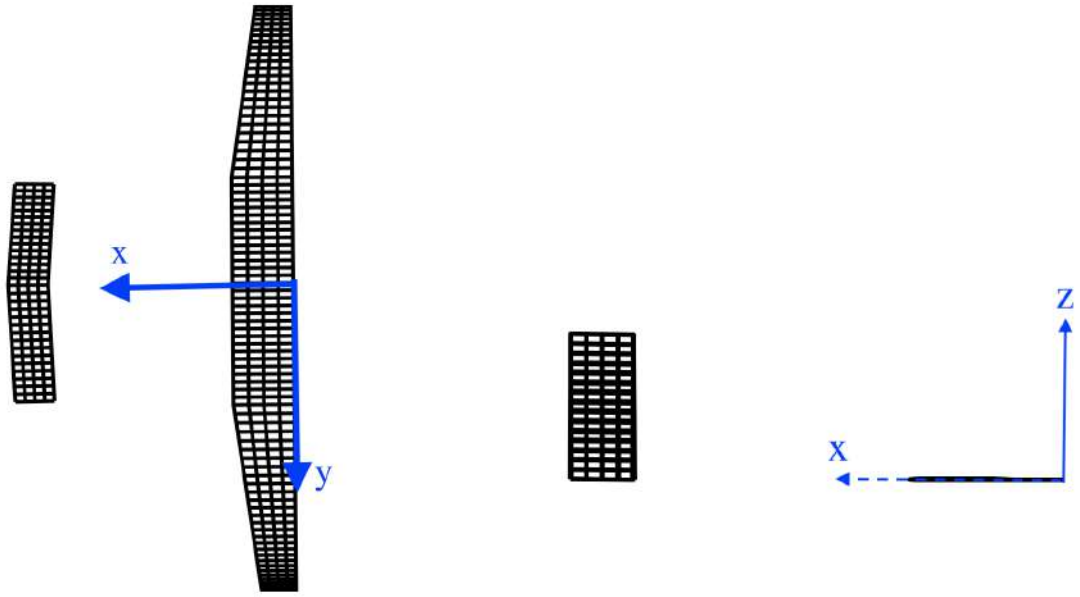
The tail geometry is simpler than the wing's thus, the built-in functions in OpenAeroStruct alone are suitable to define it. The methodology to define the tail mesh is as follows:

1. Create a rectangular mesh using OAS built-in function;

2. Apply dihedral;

3. Offset the mesh in the x and z directions

To model the tail, a rectangular mesh is generated, similarly to the wing, and then the *dihedral* function is executed with -45^o as input angle. This function performs a deformation in the z coordinates of all nodes in the mesh according to their distance to the $y = 0$ plane and the user-provided dihedral angle. With the tail geometry defined, an offset in the x and z directions are used such that the tail is placed downstream and above the main wing, respectively. A representation of the modelled UAV lifting surfaces in OpenAeroStruct is given in Figure 4.6.



(a) Top View

(b) Side view

Figure 4.6: Geometry of the modelled lifting surfaces in OpenAeroStruct

4.4 Features Added to OpenAeroStruct

In the upcoming subsections, some important features developed by the author and added to OpenAeroStruct are discussed. Subsection 4.4.1 describes how a suitable endurance estimation for an aircraft with approximately constant weight is implemented. Because this calculation requires both aerodynamic and structural parameters that are already calculated by the program, the relevant connections between outputs and inputs from different components are illustrated as well.

In Subsection 4.4.2, a description of a new *taper* class that extends the functionality of the original one is given.

4.4.1 Endurance for a H_2 Powered Aircraft

After converging the coupled aerostructural system, some relevant parameters– Functional Evaluations in Figure 4.4– like the lift L, drag D and weight W of the vehicle; with both L, D and $W = W_0 + W_s$ the fuel burnt is estimated based on the Breguet range equation. With the same reasoning, other relevant performance metrics such as the endurance time could also be added to the Functional Evaluations.

The endurance function implemented is (Raymer, 1992),

$$E = \frac{L}{D} \frac{E_{sb}\ \eta_{b2s}\ \eta_p}{(W_0 + W_s)\ V}\ m_b. \tag{4.8}$$

In Section 5.1, this function will be further discussed. Here focus is given only on how it is implemented.

In Eq.(4.8), L, D, and $W_0 + W_s$ are already calculated, the total energy content $E_{sb} \cdot m_b$ and the flight speed V are constants defined in the conceptual phase of the project. The only missing variables are the propeller efficiency η_p and the system's efficiency between the battery and the shaft that drives the propeller, η_{b2s}. In reality, η_p is dependent of several factors: propeller shape; RPMs (revolutions per minute) needed to fly a certain flight condition; and position on the UAV (pusher or tractor configuration) to name a few. η_{b2s} depends on the batteries, cables, ESCs and motors selected. A detailed study using high fidelity software of a propulsion system for the UAV designed in the conceptual phase, including the propeller performance can be found in P. Silva (2021). Because the scope of the present work is to perform aerostructural optimisation, the flight conditions are kept fixed. Hence, for a selected propeller, and electric system both efficiencies can be considered constant during cruise. The values used for η_p and η_{b2s} were also estimated in P. Silva (2021).

To implement the endurance calculation, a new OpenMDAO Explicit Component is defined. It consists of a Python class that inherits from OpenMDAO's *ExplicitComponent*. It receives as inputs all the variables mentioned above and it outputs the endurance in hours as given by Eq.(4.8). The analytic partial derivatives of the output (endurance) with respect to each input: $\frac{\partial E}{\partial L}, \frac{\partial E}{\partial D}, \frac{\partial E}{\partial W_0}, ...$ were declared and defined in this component.

Once defined, it is instantiated with the name *endurance* and added to an existing OpenMDAO *Group* that contains all the other components needed to evaluate performance parameters, the later is named *total_perf*.

A representative N2 chart with the relevant inputs and outputs of this new *endurance* component is provided in Figure 4.7. In the chart, *AS_point_0* is a *Group* which contains the aerostructural model and Functional Evaluations for the cruise flight point. Inside it, the *total_perf* group is expanded with its components. The added *endurance* component is represented in light blue, with its inputs and single output (also named endurance). These variables are identified on the diagonal. The origin of the inputs is represented in orange and their destination is represented in green. I.e, C_L and C_D are both calculated in *CL_CD* group and are both used to estimate the endurance (output). All the parameters in Eq.(4.8) assumed constant are stored inside *prob_vars* component and must be defined by the user.

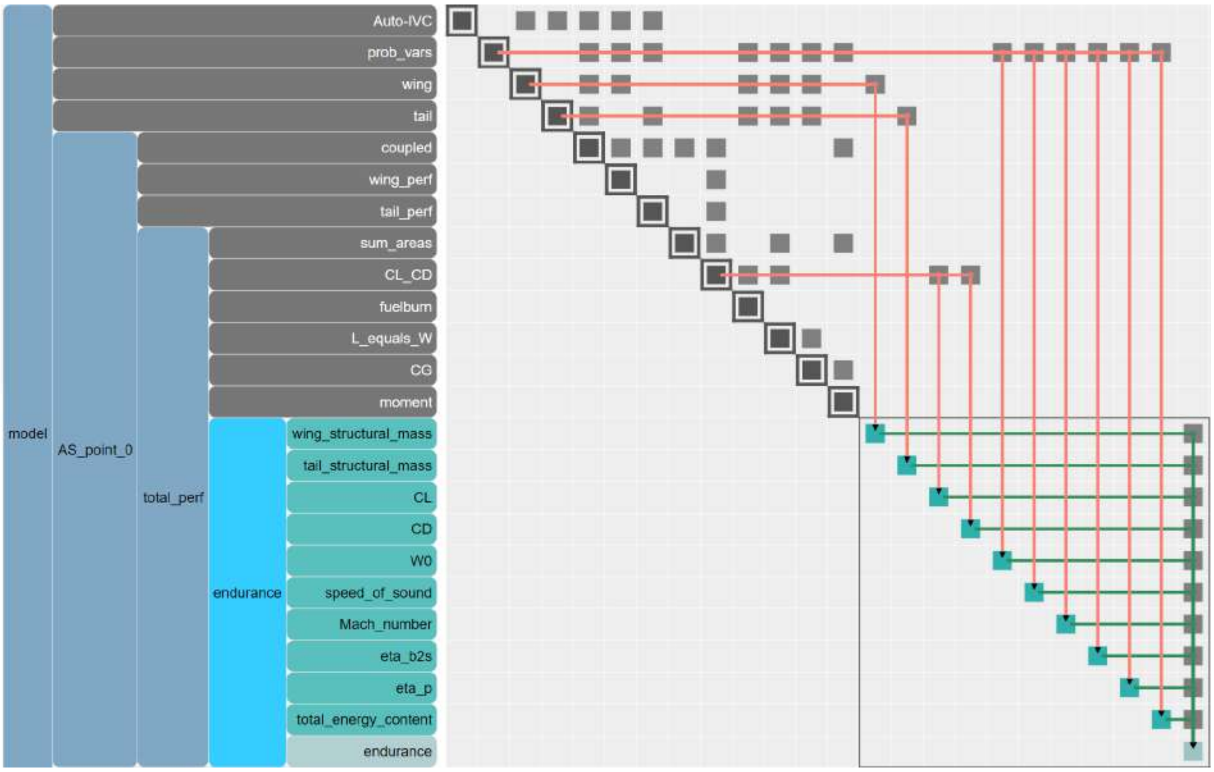


Figure 4.7: N2 chart of the model with *endurance* inputs/outputs highlighted

4.4.2 Generalisation Of The Taper Class

As stated previously, OpenAeroStruct has several geometry manipulation functions already built-in, being *taper* one of them. To use it, the user must add the pair *'taper': value* to the lifting surface being defined. The value defined is the desired $\lambda = c_{tip}/c_{root}$ ratio. With this value, the chord will decrease linearly from the lifting surface root until its tip such that the ratio $\lambda = c_{tip}/c_{root}$ is satisfied. By default, this transformation happens around the quarter-chord line, i.e, all the nodes in the original mesh that are at the quarter-chord line remain unchanged. This implies that after the transformation the leading edge of the surface is modified.

Two changes were sought:

1. Apply taper around the leading edge of the lifting surface, maintaining it perpendicular to the centreline;
2. Enable linear variations of the chord between any arbitrary position on the lifting surface and its tip.

A bottom-up approach was followed to program this new taper functionality. First, a new OpenMDAO *Explicit Component* is created. The inputs of this new component are the ratio c_{tip}/c_{root} and an offset. This offset must be given as a percentage of the semi-span from which the taper will be applied. So, for example, if the user sets this parameter to zero then taper will start at the root of the lifting surface and the behaviour of this new *taper* function does not change. If the user sets this offset to 20%, then taper will be applied between 20% of the wing semi-span (starting from the root) and the tip.

Similar to the built-in *taper*, the output of this new one is the altered mesh. When computing the new tapered mesh the reference used is the leading edge instead of the quarter-chord line.

In order to use taper as a design variable, the component partial derivatives of the output (mesh) with respect to the inputs (λ and the offset) are required. To avoid the increase in computational cost associated with the usage of approximate derivatives given by finite-differences or complex-step, analytic derivatives are provided. Here a simplification was made: the derivative of each mesh entry with respect to the offset is set to zero. This is only true in case the offset does not change in the optimisation problem. After the new class defined, unit tests were conducted to ensure that both the geometric transformation of the mesh was as intended and the analytic partial derivatives were correct. After testing the new *Explicit Component* individually, it is then implemented in OpenAeroStruct, inside the OpenMDAO *Group* that contains all the geometric manipulation components. It is instantiated from the *GeometryMesh* class and corresponds to the *mesh* dark-blue box in the N2 chart represented in Figure 4.8. In it, the existence of the string 'taper_with_offset' is checked in the surface dictionary keys. If it exists then the new *taper_with_offset* will be instantiated with the corresponding value. Then, in the parent *Group*, instantiated from *Geometry* class with the name *geometry* in the chart, the surface dictionary is again checked for the existence of the key *taper_with_offset*, if this key is found then the variable 'taper_with_offset' is added to an existing OpenMDAO *IndepVarComp* and connected to the *taper_with_offset* so that it can be used as a design variable in the optimisation problem. This *IndepVarComp* is represented as indep_vars in light blue in Figure 4.8. The taper_with_offset output can be seen being connected from *indep_vars* to the taper_with_offset component. The other input needed to apply taper to the wing is the offset, which is assumed constant. This value is stored inside the *prob_vars* component and connected to the input *offset* of the *taper_with_offset* component.

This new OpenAeroStruct development facilitates the creation of lifting surfaces with general trapezoidal shapes while also allows the use of the taper ratio λ as a design variable in the optimisation problem without affecting the rectangular portion of the wing.

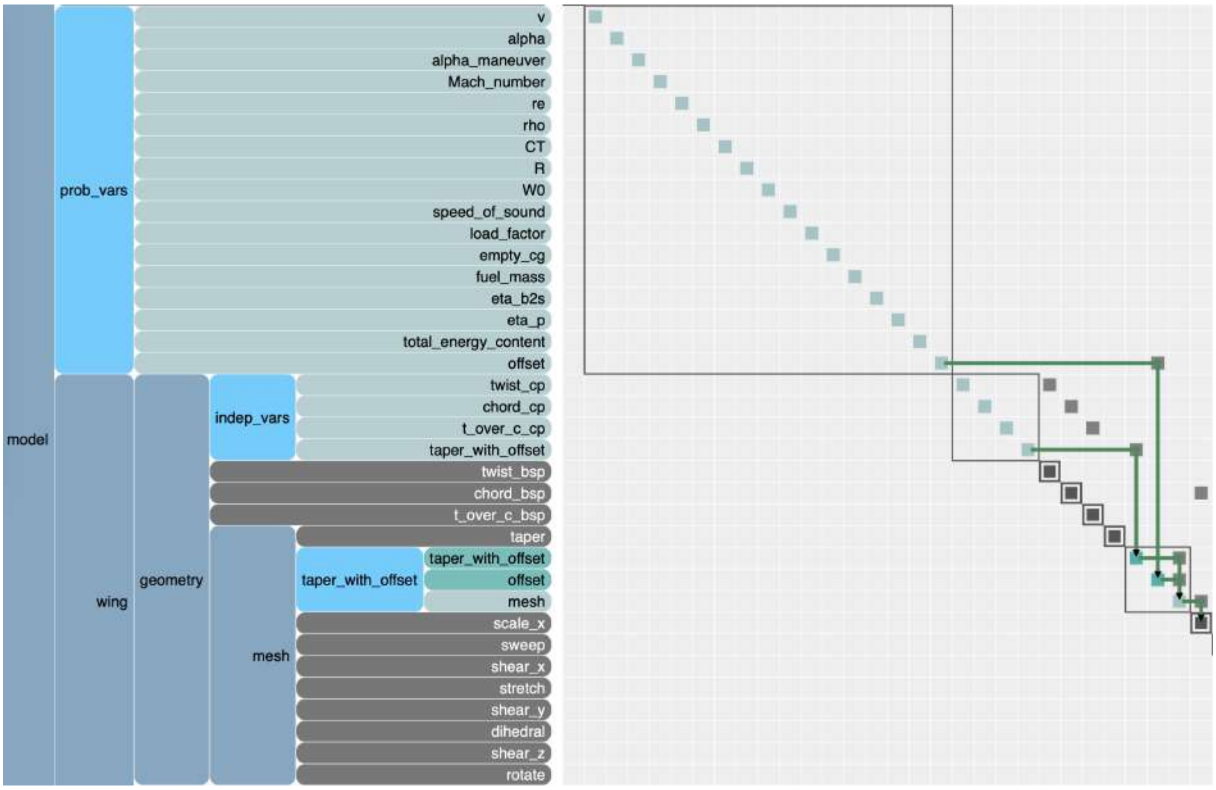


Figure 4.8: N2 chart of the model with *taper_with_offset* inputs/outputs evidenced

Chapter 5

Next Generation UAV Design

In this chapter, the optimisation problem is formally posed, the baseline solution is analysed and an optimisation is executed to improve the current solution. The results presented in this chapter were accomplished with a 2,6 GHz 6-Core Intel Core i7 processor and 16 GB of RAM running at 2667 MHz.

5.1 Optimisation Problem Formulation

As mentioned in Chapter 1, the UAV developed is for use in surveillance operations, thus aerostructual optimisation is performed with Endurance as the objective function to be maximised subject to a set of constraints.

The parameters of the UAV to be optimised are listed in Table 5.1. The total energy available for the main surveillance mission is calculated from

$$\begin{aligned} E_{\text{total}} &= m_{\text{available}} \cdot e \quad W.h \\ \text{with } e &= 33.3 \quad \frac{kW \cdot h}{kg} \\ \text{and } m_{\text{available}} &= 0.9 \cdot m_{\text{Total H2}} - m_{\text{climb}} - m_{\text{recharge}} - m_{\text{descent}} \quad g, \end{aligned} \tag{5.1}$$

considering a 10% reserve with respect to the total H_2 available in the tank and taking into account the estimated required energy to perform both climb segments of the mission profile, to recharge the battery coupled to the fuel cell and to descend.

Two flight conditions are considered for the optimisation problem: one for the main surveillance mission in steady level flight and another for a 6.0 g load case. The objective function will be maximised for the first condition and the wing structure will be sized for the second. This load case is defined according to the flight envelope of the baseline UAV, for details refer to Sá (2021).

Both wing and tail structures are assumed to be made of the same material, a composite made from a mixture of epoxy and bidirectional carbon fabric. A complete description on how the material can be manufactured and its properties can be found in J. Silva (2017). The structures of both lifting surfaces are modelled as simplified wingboxes where the front spar is placed at 10% of the chord and the rear at

Table 5.1: Parameters and specifications of the baseline UAV

Parameter	Value	Notes
Tank capacity	148 g of H_2	Maximum hydrogen stored
Energy	3479.85 W.h	Total energy available for main mission
MTOW	21.6 kg	
Main mission Mach	0.05647	
Main mission altitude	5000 ft	ISA + 20° C
6.0 g manoeuvre Mach.	0.07331	According to UAV's flight envelope
6.0 g manoeuvre altitude	0 ft	ISA + 20° C
Drag counts for VTOL rotors	350	
Aircraft weight without wing and tail structures	17.118 kg	
Structural material density	1300 kg/m^3	Based on epoxy + bi carbon fabric
Structural Young's modulus	48.99 MPa	Based on epoxy + bi carbon fabric
Structural shear modulus	5 GPa	Based on epoxy + bi carbon fabric
Structural yield strength	567.79 MPa	Based on epoxy + bi carbon fabric

60%. The airfoil coordinates of 10% to 60% of the chord of SG6042 and NACA 0008 are used for the cross-section shape of the wing and tail, respectively. These are scaled with the chord at each cross-section. The cross-sectional properties along the wingspan, namely the area and the area moment of inertia, are calculated and then used in the stiffness matrix of each FEM element Eq.(4.5). For details on the wingbox model used and its implementation in OpenAeroStruct consult Chauhan and Martins (2019).

To account for components not modeled with the wingbox structure such as overlaps, fasteners, ribs and bulkheads, the weight of the wing structure obtained with the FEM model is multiplied by a factor of 1.25.

Two possibilities to formulate the aircraft endurance are discussed. The UAV carries H_2 as fuel, stored in a tank, to use as fuel. H_2 will flow from the tank to the fuel cell where it will be converted into water and electric current. This current is then going to power an electric motor which is responsible to drive a propeller to provide the thrust required for the vehicle to fly. The endurance equation for propeller-driven aircraft at level flight is given by (Raymer, 1992)

$$E = \int_{W_i}^{W_f} \frac{1}{-CW} \left(\frac{L}{D}\right) dW \tag{5.2}$$

$$\text{with } C = C_{\text{power}} \frac{V}{\eta_{\text{p}}},$$

where C_{power} is the power-specific fuel consumption [kg/J], V is the aircraft speed [m/s], η_p is the propeller efficiency, W_i and W_f represent the UAV weight at the beginning and at the end of the segment [N], respectively. Since for the surveillance mission the UAV is not expected to drop payload, the difference in weight will be due to the H_2 consumption, $W_f - W_i = W_{\text{H2 consumed}}$.

The baseline solution of the UAV configuration has a tank capable of storing $148\,g$ of H_2. Assuming all of it is consumed, then the maximum weight variation of the UAV would be 0.685% of its total weight. Assuming this variation to be negligible, the Endurance equation for steady level flight of electric propeller-driven vehicles, adapted from Raymer (1992), can be expressed by

$$E = \frac{L}{D} \frac{E_{sb}\,\eta_{b2s}\,\eta_p}{(W_0 + W_s)\;V}\,m_b \qquad \text{(4.8 revisited)}$$

and used as an alternative, with E_{sb} being the battery specific energy [Wh/kg], η_p the propeller efficiency, η_{b2s} the system efficiency from battery to motor output shaft, m_b the battery mass [kg] and W_0 and W_s the empty and structural weight of the aircraft, respectively. For the baseline, $W = 21.6\,kg$ during the whole mission.

Although the designed UAV has a battery coupled to the fuel cell for delivering extra power when required, it does not store the total energy required to fly the main mission. The energy is stored in the chemical bounds of the H_2 which in turn is stored inside the tank. As such, the total energy available in Eq.(4.8) given by $E_{sb} \cdot m_b$ is replaced by the right hand side of Eq.(5.1).

In the previous equation (4.8), some of the parameters are going to be assumed constant, namely the propeller efficiency η_p, the total system efficiency between the fuel cell and the motor output shaft, and the total energy content. The H_2 capacity of the tank was chosen based on previous studies, hence in this optimisation problem the total energy content stored in the H_2 chemical bounds is fixed as well. By varying parameters like wing geometric twist, and structural parameters such as spar thickness, it is possible to increase the Endurance time by either increasing aerodynamic efficiency (L/D) or decreasing the UAV's structural weight W_s, respectively.

A summary of the optimisation problem is provided in Table 5.2.

Table 5.2: Optimisation problem

	Function/variable	Note	Quantity
maximise	endurance	computed using Eq.(4.8)	
wrt	wing twist	b-spline parametrised using 5 control points	5
	spar thickness	b-spline parametrised using 6 control points	6
	skin thickness	b-spline parametrised using 6 control points	6
	α	for the main mission	1
	α_i	for the main mission	1
	$\alpha_{\text{6.0 g}}$	for the 6.0 g load case	1
		Total design variables	**20**
subject to	$L = W$	for the main mission	1
	$C_M = 0$	in the main mission to ensure trimmed flight	1
	$L_{\text{6.0 g}} = W_{\text{6.0 g}}$	for the 6.0 g manoeuvre flight point	1
	$\sigma_{\text{von Mises}} \leq \frac{\sigma_{\text{yield}}}{2}$	von Mises stresses aggregated using a KS function	1
	$TA_{\text{monotonic}} < 0$	monotonic twist constraint	1
		Total constraint functions	**5**

For structural sizing, a 6.0 g manoeuvre case is considered. With the corresponding loads and the wingbox geometry, the von Mises stresses of the most critical points in the upper and lower surfaces of the wing are calculated through a Kreisselmeier–Steinhauser (KS) function (Kreisselmeier & Steinhauser, 1979). In the spars, the maximum bending stresses, the maximum transverse shear stresses and the maximum torsional shear stresses are combined together and the von Mises stresses on the spars are calculated. Finally, using a KS function, all the von Mises stresses are aggregated into a single stress constraint.

To ensure a realistic design space, it is also enforced equilibrium by setting $L = W$ and trimmed conditions in level flight by setting C_M = 0. For the load case, only the former is imposed. Because the configuration to be optimised has a tail, the trim condition in steady level flight is achieved with the tail incidence angle, α_i.

To ensure that the optimiser produces a monotonically decreasing twist distribution from wing root to tip (wash-out), the constraint: $TA_{\text{monotonic}} < 0$ is added. This constraint function calculates the geometric twist difference between two adjacent points in the VLM mesh, from tip to root.

The design variables of this optimisation problem are the wing geometric twist, thickness-to-chord ratio, spar and skin thicknesses controlled by b-splines that define the parameter distributions from a number of given control points and the angle of attack and the tail incidence angle, which are scalars. The same thickness distribution is used for both front and rear spars, and the same thickness distribution is used for both the upper and lower skins. The lower and upper bounds for the wing geometric twist, spar and skin thicknesses are -15° and 15°, 0.6 and 3 mm, respectively. The bounds of the variables are summarised in Table 5.3.

Table 5.3: Bounds of the design variables for the MDO problem.

Design variable	Lower bound	Upper bound	Units
Wing twist	-15	15	deg
Spar thickness	0.6	3	mm
Skin thickness	0.6	3	mm
Manoeuvre angle of attack	-20	25	deg
Angle of attack	-10	10	deg
Tail incidence angle	-15	15	deg

The optimisation problem is solved with the gradient-based algorithm SLSQP (Sequential Least Squares Programming) (Kraft, 1988) available in the Python library *SciPy* (Virtanen et al., 2020) with the tolerance set to 1×10^{-7} and 200 maximum iterations as fallback.

Both during and after the optimisation process, some relevant parameters like the design variables, objective, constraints and other intermediate parameters are saved to a *sql* database so that they can be used for post-processing analysis. The post-processing (data manipulation and visualisation) is done with the Open Source *NumPy* (Harris et al., 2020) and *Matplotlib* (Hunter, 2007) Python libraries.

5.2 Aerodynamic Mesh Convergence Study

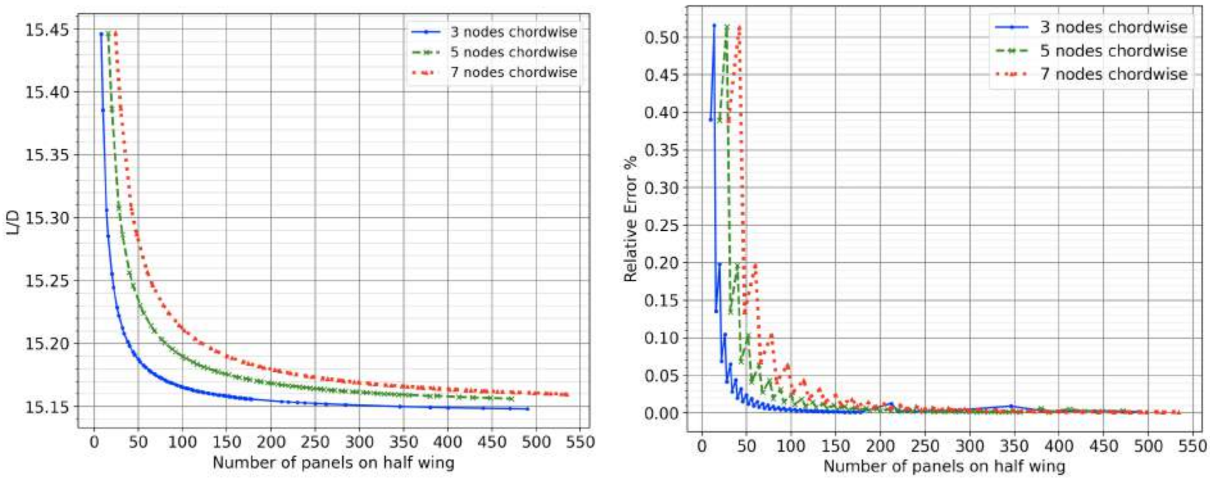
In order to define an adequate mesh, a convergence study is performed. A purely aerodynamic model is set up and analysis is executed in OpenAeroStruct, i.e, no *driver* is called to iteratively search for the optimal solution. For each analysis, the number of panels changes but all other parameters (both flow conditions as well as wing geometry) remain constant. The output variable aerodynamic efficiency $\frac{L}{D}$ is observed against the number of panels to select a mesh that satisfies a convergent trend with the lowest number of panels possible as it affects directly the computational time of the analysis.

In total, three different chord-wise discretisations were considered: 3, 5 and 7 nodes. For each of these discretisations, L/D was calculated for a given number of span-wise nodes. The aerodynamic efficiency as a function of the number of panels on the half wingspan is plotted in Figure 5.1(a). It can be observed from the mentioned figure that all the curves have a similar behaviour: as the number of panels increases, L/D decreases with a steep gradient, after a certain number of panels (between 50 and 150), it becomes practically constant.

The relative difference between consecutive points, the error, was determined (Weisstein, 2021) as

$$\delta = \left| \frac{(L/D)_i - (L/D)^{\text{ref}}}{(L/D)^{\text{ref}}} \right| \cdot 100 \quad \%. \tag{5.3}$$

The relative error δ as a function of the number of panels on half wingspan is plotted in Figure 5.1(b). As observed with L/D, the behaviour of δ with respect to the number of panels is similar for all three chord-wise panel discretisations. After approximately 200 panels, the error is negligible ($< 0.01\,\%$). Increasing further the number of panels will increase the computational effort without significant reduction of the relative difference δ, therefore a total of 200 panels will be used in subsequent analysis and optimisations runs. Since the value obtained for L/D with 5 chord-wise nodes (green dashed line on both mentioned Figures) is always lower than the value obtained with 7 nodes and greater than the value obtained with 3 nodes for the same number of panels, this panel discretisation was chosen.



(a) L/D vs number of panels

(b) L/D relative error

Figure 5.1: VLM mesh convergence analysis

5.3 Analysis of Baseline Solution

The baseline solution is analysed at two distinct flight conditions: steady level flight where the objective function endurance is evaluated, i.e, subject to $L = W$ and $C_m = 0$; and the manoeuvre case specified previously also subject to $L = W$ where the von Mises stresses on the wing structure are computed. To ensure that the constraints for each flight point are respected, freedom is given to the angle of attack and tail incidence angle, α and α_i respectively. Because no freedom is given to structural design variables, the failure constraint is not imposed.

The problem was successfully solved (Exit mode: 0) and the constraints were respected. The results for the baseline UAV are summarised in Table 5.4 and Figure 5.2.

Table 5.4: Performance of the baseline UAV

Parameter	Value	Unit
Endurance	03:29	h:min
C_L	0.601	-
C_D	0.042	-
L/D	14.24	
MTOW	21.6	kg
α	3.28	deg
$\alpha_{4.0g}$	11	deg
α_i	-1.13	deg
S_{wing}	1.373	m^2
V_{cruise}	38	kts

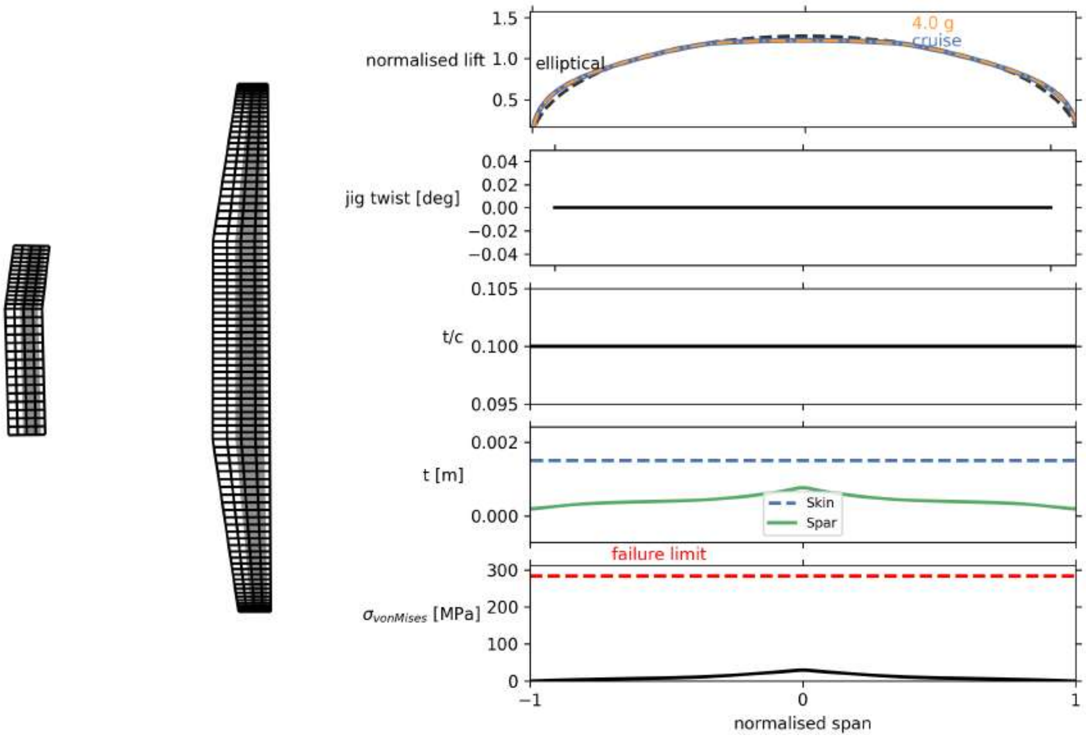
The computed Endurance of $03h29$ is slightly higher than the value obtained in the conceptual phase, $03h20$. This small difference is due to the higher aerodynamic efficiency that is obtained using OpenAeroStruct and the one estimated in the conceptual phase. In the conceptual phase, the Oswald efficiency factor (e) was estimated to calculate the induced drag,

$$C_D = C_{D0} + kC_L^2$$

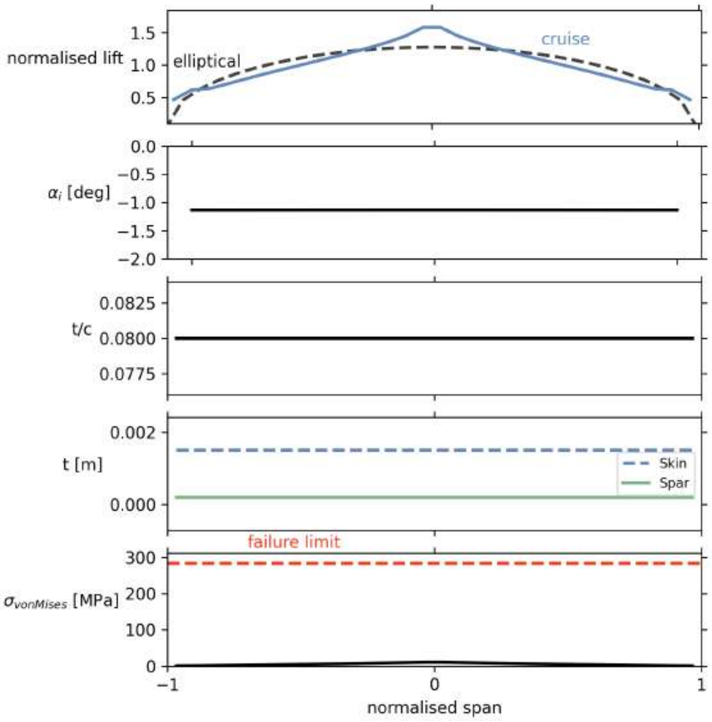
$$\text{With } k = \frac{1}{\pi ARe}.$$

In OpenAeroStruct the induced drag is the streamwise component of lift. In Figure 5.2(a) both the normalised lift along the wingspan (in blue) and the elliptical lift distribution (black dashed lines) are plotted. The two are almost identical and, as a consequence, $e \approx 1$. As such, it is likely that the factor used for the whole aircraft in the conceptual phase, which was estimated to 0.75 might be overestimated. Thus, the induced drag coefficient k was higher which led to a smaller endurance time as given by Eq.(4.8).

Other important results are the von Mises stresses computed for the load case for both the wing and tail structures. Both structures are assumed to be made of the same epoxy and carbon fabric material with a yield strength of 567.79 [MPa]. The allowable stress limit ($567.79/2$ MPa) is represented in red in both Figures 5.2(a) and 5.2(b). As it can be observed, the aggregated von Mises stresses along the



(a) Wing



(b) Tail

Figure 5.2: Baseline parameter distribution along wingspan

wingspan are smaller than the limit for all the wingspan. At the root, where the stresses are higher, the value obtained is on the order of magnitude of 10^6 while the allowable stress is in the order of magnitude of 10^8. On one hand, it is good that there is a margin between the maximum stresses in the structure for the worst load case that the UAV has to withstand and the allowable stress. On the other hand, in this case, this margin is too large, which implies that both wing and tail structures are oversized. As a consequence, the weight of the structure is higher than needed which penalises the endurance of the UAV. The weight of the structural components can be reduced without risking failure to increase endurance, according to Eq.(4.8).

5.4 Parametric Studies, WingBox Size Influence

To perform the analysis of the baseline solution, the portion of the airfoil that is occupied by the wingbox structure was defined as $10 - 60\,\%$ of the chord from the leading edge for both wing and tail. Because OpenAeroStruct does not allow the variation of these coordinates during optimisation, the influence of both the distance between spars and location of the structure in the airfoil is subject of parametric studies in this section.

The model analysed is kept fixed with the exception of the airfoil portion that is used to model the wingbox. In total, six cases are studied: three different portions are used for the wing structure (tail is fixed) and, later, three different portions of the tail airfoil are tested for a fixed wing structure, as summarised in Table 5.5. For all the test cases only the location of the front and rear spars vary, the distance between them is kept fixed: $0.10 \times c_{\text{wing}}$ for the wingbox structure and $0.15 \times c_{\text{tail}}$ for the tailbox structure.

Table 5.5: WingBox parametric cases

	Wing			Tail		
Case	a)	b)	c)	a)	b)	c)
Location	30-40%	35-45%	25-35%	30-45%	35-50%	15-30%

The results for test case a) of the wing and tail are shown in Figures 5.3 and 5.4, respectively. In both figures, the portion of the lifting surface that is used to construct the box model is represented as a dark grey area inside the VLM mesh. The twist, thickness-to-chord-ratio, skin and spar thicknesses were kept constant for each lifting surface, as such, those three subplots in Figures 5.5 and 5.6 were the same for the other two cases that are not shown.

The von Mises stresses that were obtained for wing and tail cases a), b) and c), although not exactly equal, were very similar and, hence, the resultant von Mises stresses distributions for the other cases are also omitted.

In the wing, the $\sigma_{vonMises}$ are significantly higher near the root of the wing where the maximum value almost reached $200\,[MPa]$, never reaching the allowable yield stress, $\sigma_{yield}/2$ (a factor of two is added for safety concerns since the material used is a composite), of $567.79/2\,[MPa]$.

In the tail structure, the von Mises stresses are also higher near the root, however, for this lifting surface, the magnitude of such stresses are much lower (at the root of the tail the aggregated von Mises stresses do not reach 100 MPa). For all wing and tail test cases, the structure of the lifting surfaces are

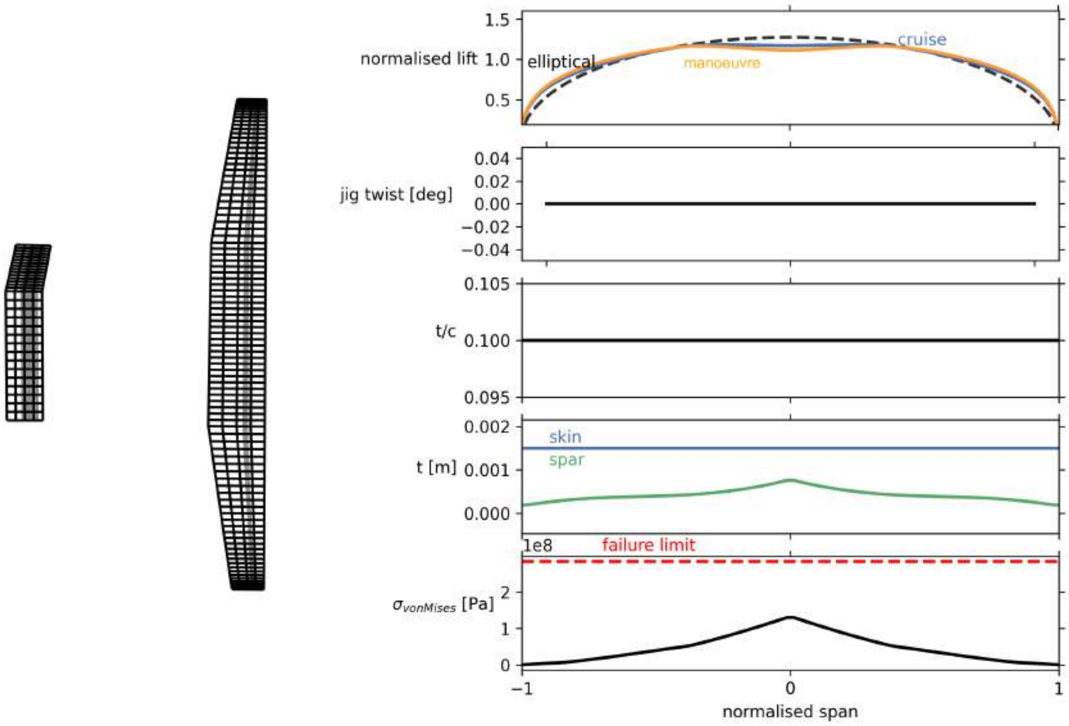


Figure 5.3: Parameter distribution along wingspan for wing test case a)

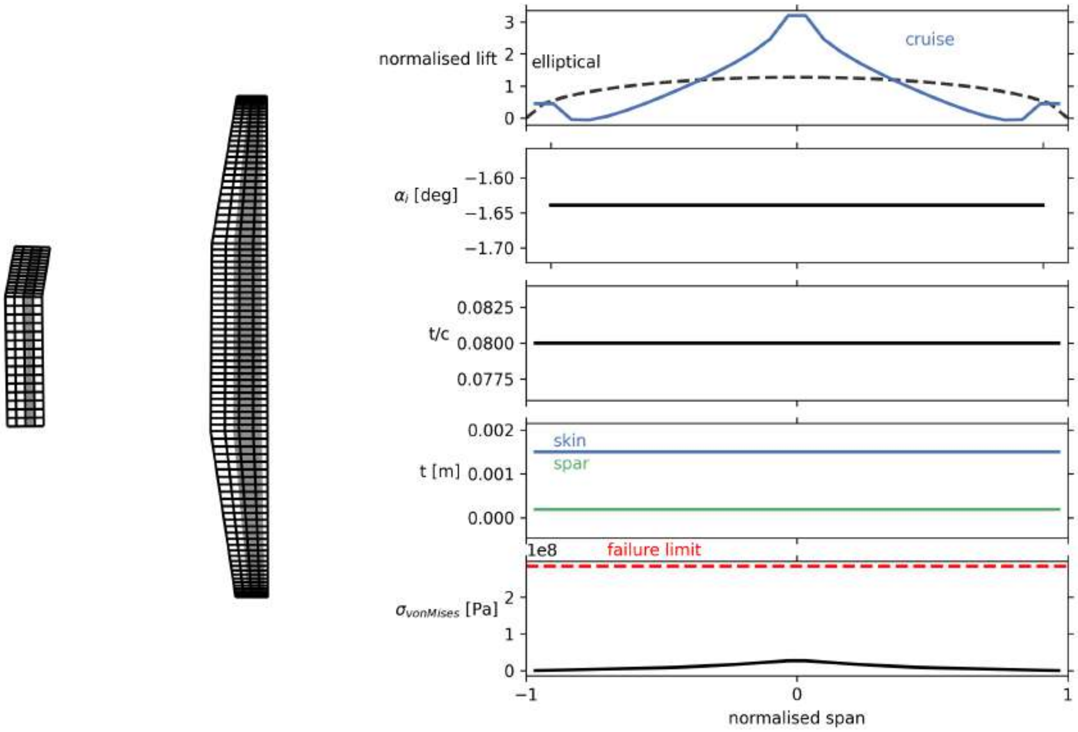


Figure 5.4: Parameter distribution along tailspan for tail test case a)

expected to endure the loads without failing.

One common trend to all three wing test cases is that near the tip of the wing, the lift distribution curve is greater than the elliptic. As it gets closer to the root, it inflects downwards and becomes lower than the elliptic. This trend is captured on the first subplot of Figure 5.3.

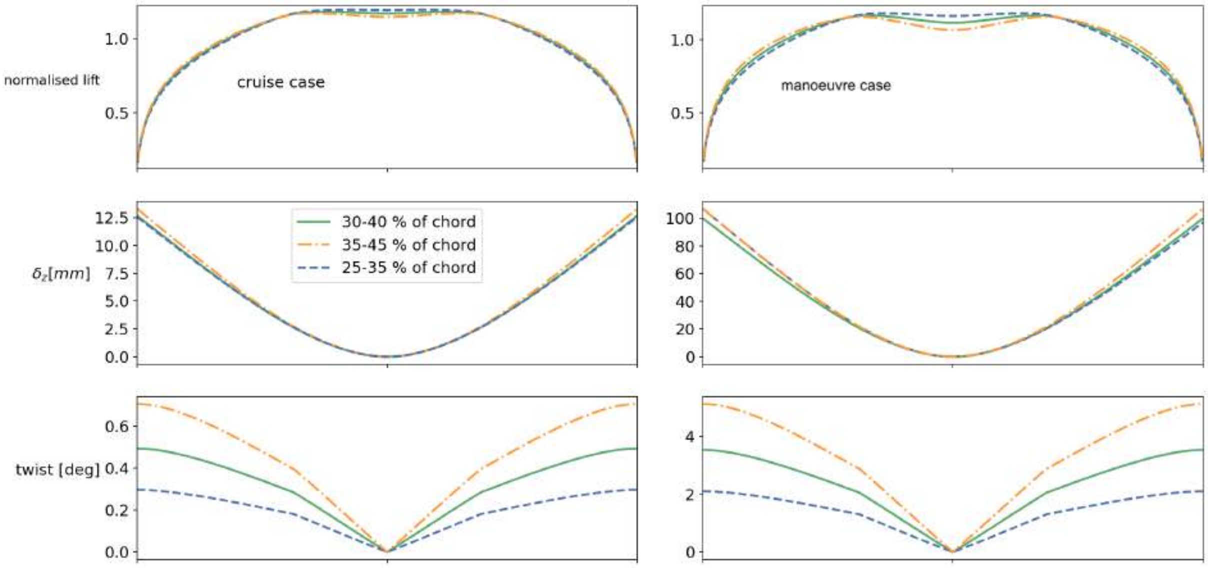


Figure 5.5: Results of the wingbox parametric studies

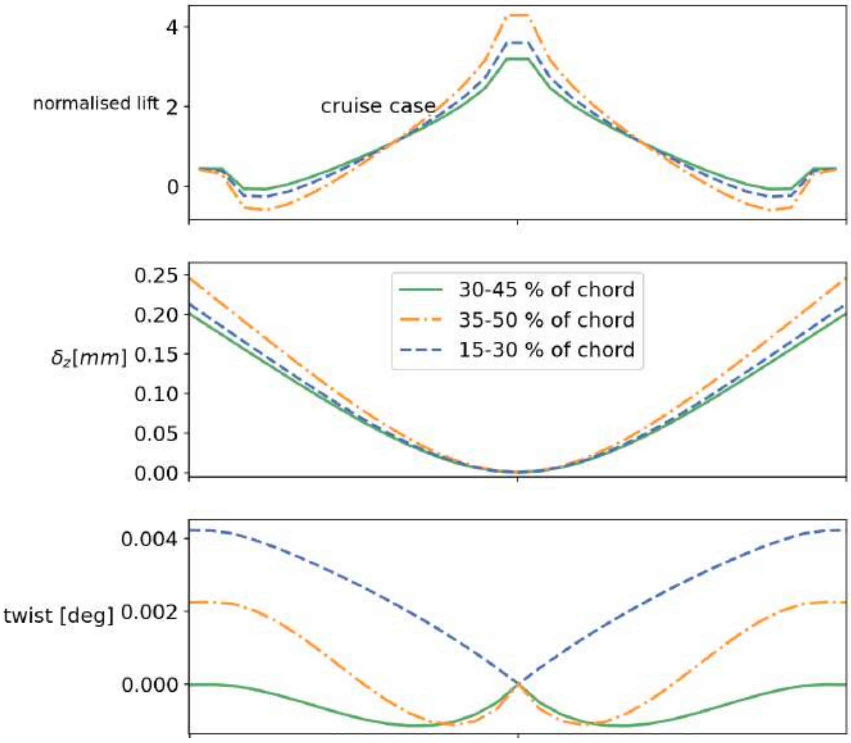


Figure 5.6: Results of the tailbox parametric studies

As mentioned previously, the only variation was the placement of the wing or tail structure with the relative distance between spars kept constant. Even though the aerodynamic parameters and flight conditions were kept fixed for all cases, the small changes made to the wingbox position have notable impact on the obtained lift distributions for both flight points. This is due to the coupling between aerodynamics and structures disciplines: small changes in the structures will impact the aerodynamics (lift distributions affected) and the opposite is also true.

The results obtained for the different portions of the wing airfoil used are plotted in Figure 5.5 and the ones obtained with different tail airfoil portions are in Figure 5.6. Both figures have 6 subplots in them: three on the left side which correspond to the cruise flight point; and three on the right which correspond to the manoeuvre case. In each subplot three curves can be observed: one for each test case as indicted in the legend.

By changing the location of the front and rear spars, the location of the shear centre of the wingbox changes. The further away the shear centre is from the centre of pressure of the lifting surface, the higher the twisting moment will be. The surface true shape in flight has some twist. This twist deformation alters the local effective angle of attack, α_{eff}, and thus, changes lift on that particular section of the surface.

Through observation of Figure 5.5 it can be concluded that in the cruise flight point there is no significant difference on the lift distribution over the wing as all three lines are virtually coincident. In the manoeuvre case that is no longer true as the three lift lines are clearly distinguishable on the right subplot of the figure.

As expected, the vertical displacements δ_z (displacements on the global z direction) are much higher at the manoeuvre case than at cruise. No significant difference can be observed between the three test cases. The twist deformation is similar for both flight conditions and the overall magnitude is small, $< 1^o$ for cruise.

Overall, placing the wingbox structure closer to the leading edge (between 25-35 % of the wing airfoil chord), case c), seems to be the best option: the twist deformation and the displacement δ_z at each spanwise location are the lowest of all three and the lift distribution is closer to elliptic, which in turn implies higher L/D. The worst performing wingbox is the one located at 35-45 % of the wing airfoil chord as it has the largest displacements δ_z and twist deformations for each spanwise position of all three and the lowest L/D.

Tail Parametric Studies A comparison between the three tail test cases can be done by observation of Figure 5.6. The three lift distribution curves, in contrast with the wing test cases, can be easily identified in the cruise flight point.

The overall best position to place the tailbox structure is between 30-45 % of the tail airfoil chord from the leading edge as this case was the one with lowest vertical displacements and lowest twist deformations when compared to the other two cases and, hence, it is the most robust for the aerodynamic loads.

Wing Wake Influence on Tail For all previous tail test cases, the tail lifting surface was in the wake of the wing. During the conceptual phase of this project, one of the reasons that motivated the choice of the inverted V tail was to minimise this influence.

To understand that influence, the wing was completely removed from the model and a new analysis was performed. To compare the effect of the wake, the wingbox size and location is kept fixed as well as the flight conditions (speed, angle of attack, incidence of the tail...). This analysis was conducted for tail test case a), i.e, for a tail structure placed between 30-45 % of the airfoil chord from the leading edge. The lift, vertical displacement and twist deformation distributions along the tail-span are plotted in Figure

5.7.

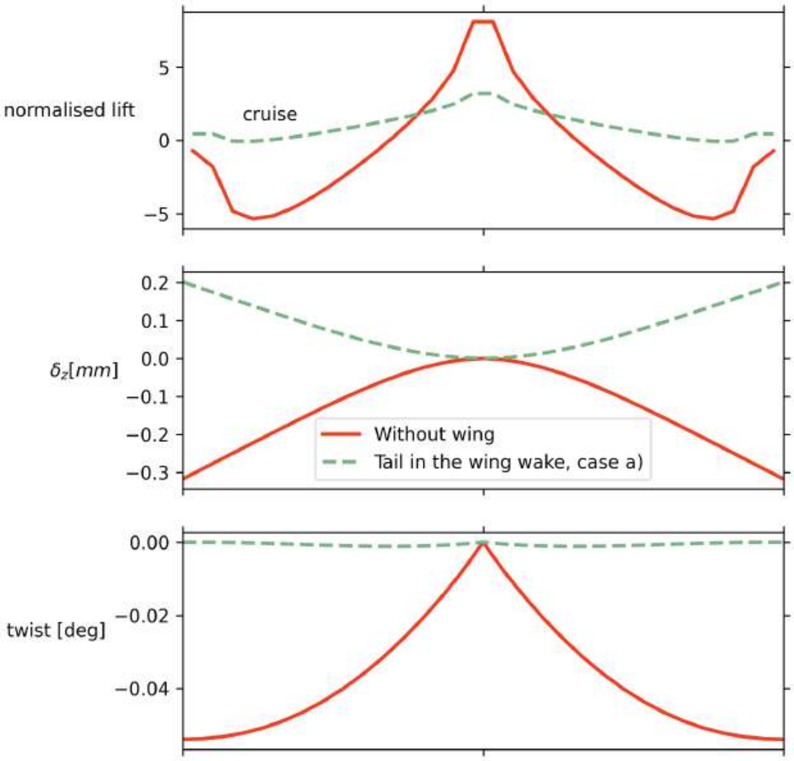


Figure 5.7: Wing wake influence on tail

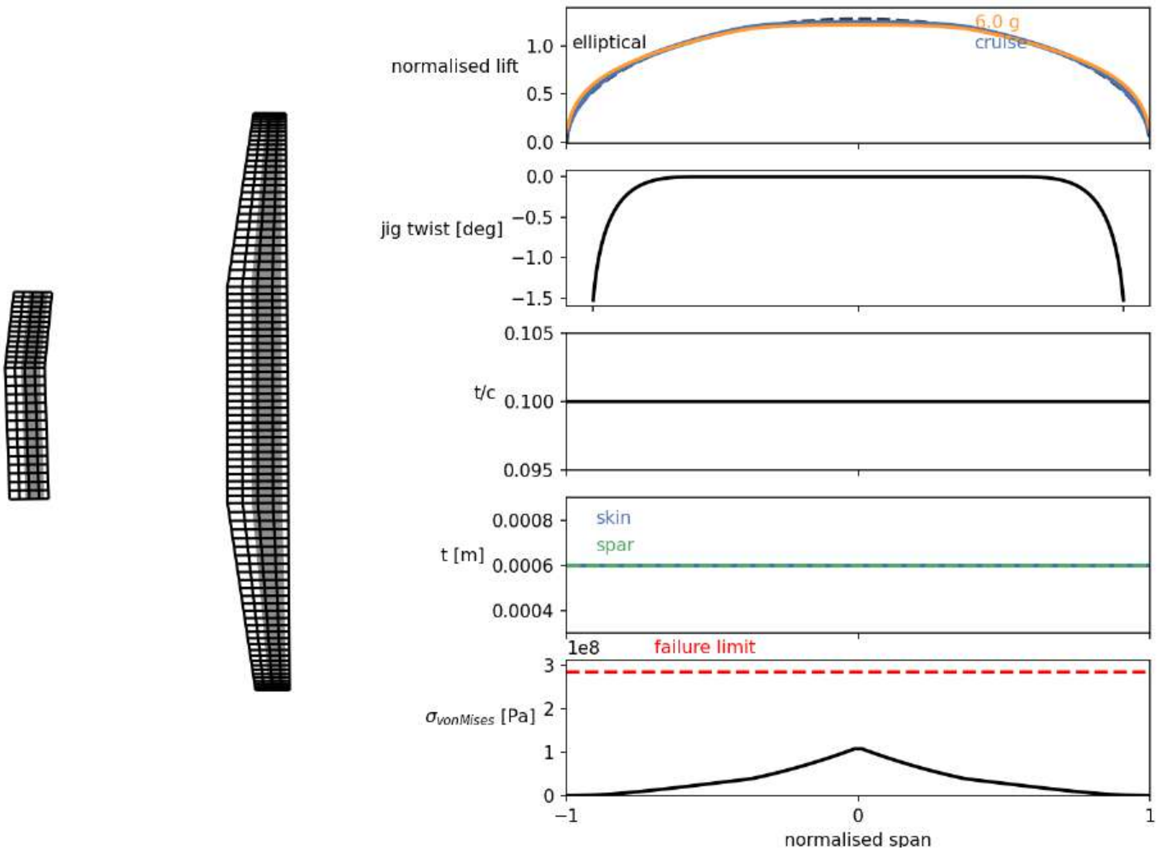
It is clear from the plot that the presence of the wing has a strong impact on the aerostructural performance of the tail. In OpenAeroStruct, the wake of the lifting surfaces is modelled by two trailing edge vortices aligned with the free stream. Without the wing, the lift distribution on the tail is positive near the root and negative closer to the tips. The deflection, although small (0.3 mm max.), is now upwards and the twist is an order of magnitude higher, nevertheless it is still very small ($<< 1^o$).

5.5 Improved Solutions

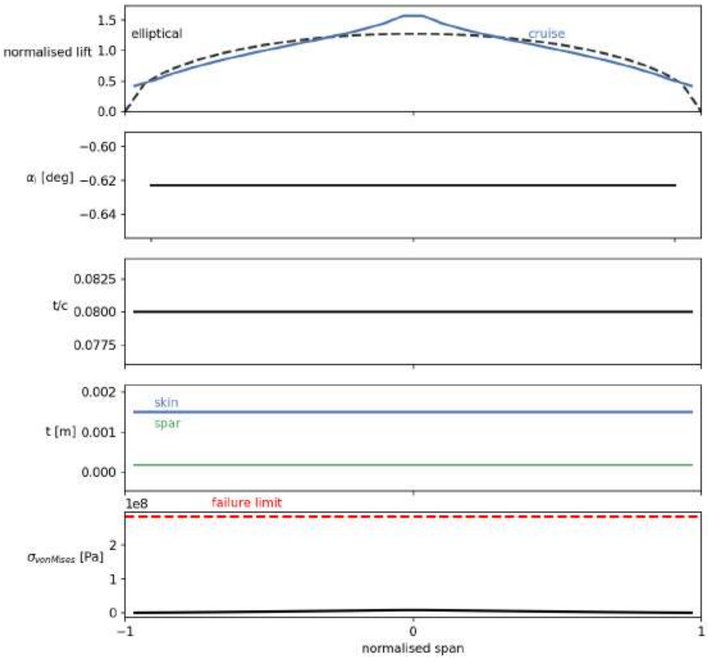
In the present section the improved solutions will be presented. To study the effect that some design variables have in the solution and explore the optimiser trends, the optimisation problem gradually grew in complexity. In all of them, the same number of structural design variables are used.

First, only geometric twist is allowed to vary. Then, this variable is substituted by taper with the use of the *taper_with_offset* functionality explained in Subsection 4.4.2. Afterwards, the chord at the root of the wing is added and, finally, the wingspan.

Later, with all the planform design variables introduced in the optimisation problem, the effect of the addition of a stall constraint was studied. In the last case, Subsection 5.5.6, a stall flight point is introduced to ensure the stall speed requirement of 28 kts.



(a) Wing



(b) Tail

Figure 5.8: Optimised parameter distribution along wingspan

5.5.1 Using Geometric Twist

With the optimisation problem defined in Table 5.2, the program was executed. After 16 iterations, the algorithm was able to converge to an optimal solution without violation of the constraint tolerances. The relevant parameter distributions along the span of both lifting surfaces were obtained and are plotted in Figure 5.8.

On the wing surface, Figure 5.8(a), the normalised lift distribution for the load case and for the cruise case is almost identical, both curves are close to elliptical. The major difference is near the wing root where both are below the elliptical (dashed line).

By introducing twist in the optimisation problem, it was expected that the optimiser introduced wash-out to reduce the lift-induced drag and, as a consequence, maximise L/D for the given flight conditions. Increasing this parameter would increase the Endurance time, as given by Eq.(4.8), which is the goal. The application of wash-out is visible in the second subplot of Figure 5.8(a). Near the root of the surface, the angle is null; as the wing progresses towards the tip, the geometric twist decreases, reaching the minimum value of -1.5 deg. The obtained lift distribution in the first subplot of the same figure corroborates the obtained twist distribution. Near the wing root, the lift (blue and orange lines) is lower than the elliptic (dashed black line), which is expected since the wing planform is rectangular and no twist is introduced. As the wing progresses to the tip, the planform is no longer rectangular (it has a taper λ) and geometric twist is applied, hence the obtained lift distribution is practically coincident with the elliptic.

The von Mises stresses on the structure of the wing are also lower than the allowable stress, represented in red. Hence, failure is not expected to happen. It is worth mention that all skin and spar thicknesses control points of the wing structure correspond precisely to the lower bound of both design variables: $0.6\,mm$.

In Figure 5.8(b), the relevant parameter distributions along the tail span are represented. Although no failure constraint was imposed on the tail structure, it is clear that the von Mises stresses are away from the allowable stress, which again is an indicator that failure on the tail structure is not expected.

Table 5.6: Optimisation results using geometric twist

	Optimised		Baseline	
Parameter	Value	Unit	Value	Unit
Endurance	03:35	h:min	03:29	h:min
C_L	0.536	-	0.601	-
C_D	0.041	-	0.042	-
L/D	13.1	-	14.24	-
MTOW	19.42	kg	21.6	kg
W_s	2.80	kg	4.98	kg
α	2.43	deg	3.28	deg
$\alpha_{6.0g}$	10.64	deg	11	deg
α_i	-0.82	deg	-1.13	deg

Other important results such as the objective function value, the MTOW and some aerodynamic properties are summarised in Table 5.6. The results obtained for the baseline solution, Table 5.4, are repeated here to make the comparison between the two easier.

When performing optimisation, the number of free parameters increases due to the addition of the design variables to the problem formulation, hence the design space is larger. In addition, a constraint (the wing failure constraint) and an objective function (the endurance) were also added. Due to the increase in complexity of the problem, the increase of the needed computational time (33 min to 04h52) from the analysis case to the optimisation was expected.

The optimised solution has a lower C_L than the baseline, which is also expected: there is a reduction of approximately 10.1% in the UAV weight. As such, the required C_L for level flight is lower. This reduction is obtained with a decrease of α: for the baseline $\alpha = 3.28^o$ and for the optimised solution $\alpha = 2.43^o$.

Since freedom was given to both structural and aerodynamic design variables, it is possible to improve endurance through a combination of the two effects: increase in L/D and reduction of W_s. A reduction of W_s would decrease the total weight of the UAV ($MTOW = W_0 + W_s$), therefore the amount of lift necessary to have equilibrium $L = W$ would also be lower. Because the baseline structure of the wing was oversized, there was possibility of improvement with respect to the weight of the structures. Therefore, the optimiser reduced W_s until (in this case) the structural design variables reached the lower bound.

Achieving less lift is possible with a reduction of the UAV speed or α (or a reduction of both). In the optimisation problem posed, the speed is kept fixed at 38 kts so, in case W_s decreases, the only possible solution would be to decrease α. With a decrease in α, both lift and drag are expected to decrease, nonetheless the ratio between the two L/D is expected to change: $\frac{\partial(C_L/C_D)}{\partial\alpha} \neq 0$. Even though C_L/C_D decreased, the reduction of W_s is such that the overall Endurance time is higher. This results suggests that it could be possible to further improve the endurance of the optimised UAV through a decrease in the flight speed or wing area: a reduction of each would imply an increase in α such that the equilibrium condition $L = W$ is satisfied. Which in turn would imply an increase in L/D and, consequently in endurance.

If freedom is not given to any structural design variables, the optimisation algorithm will not be able to find solutions with lower weight $W_0 + W_s$. Therefore, the same amount of lift as the baseline would be necessary to achieve equilibrium. As reduction of weight is not possible, the optimiser will try to increase L/D, which is the only parameter that is free in Eq.(4.8). To test this hypothesis, the same optimisation problem was solved again without structural design variables. After 1h29 the results were obtained, these are shown in Figure 5.9 and Table 5.7.

From Figure 5.9, it is observable that both lift distributions (cruise and 6.0 g load case) are almost coincident with the optimal elliptic lift distribution for the given flight conditions. The jig twist distribution found by the optimiser is similar to the previous one, though the minimum value obtained at the tips is lower than before (-2.0 deg without structural weight design variables and -1.5 deg in the previous optimal solution with structural design variables).

From Table 5.7, it is possible to verify that, with a higher α of 3.34 deg for the case where the structures are kept fixed, the aerodynamic efficiency L/D is higher than the previous optimal solution

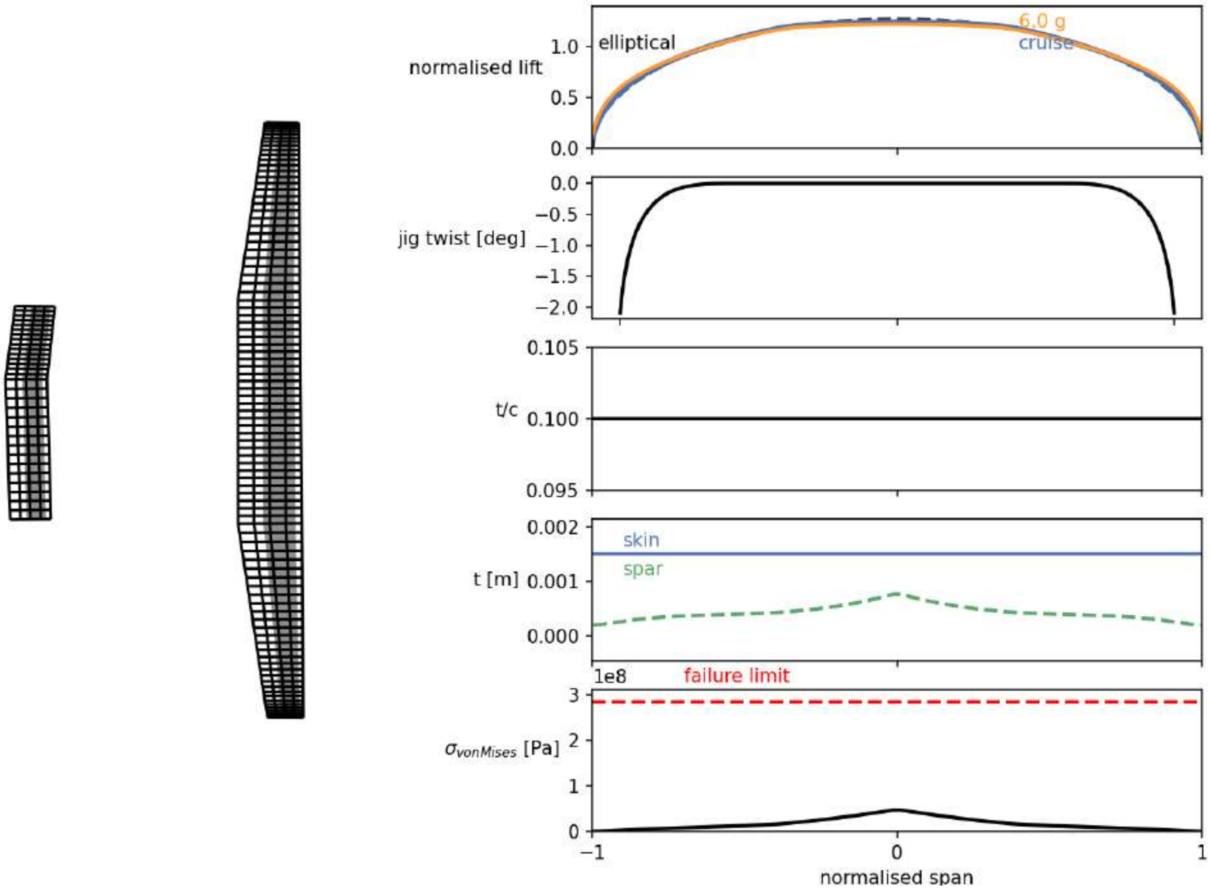


Figure 5.9: Wing parameter distribution along wingspan with fixed structural variables

Table 5.7: Comparison between optimal solutions and baseline design

	Optimised		W. Fixed Structures		Baseline	
Parameter	Value	Unit	Value	Unit	Value	Unit
Endurance	03:35	h:min	03:30	h:min	03:29	h:min
C_L	0.536	-	0.596	-	0.596	-
C_D	0.041	-	0.042	-	0.042	-
L/D	13.1	-	14.25	-	14.24	-
MTOW	19.42	kg	21.6	kg	21.6	kg
W_s	2.80	kg	4.98	kg	4.98	kg
α	2.43	deg	3.34	deg	3.28	deg
α_i	-0.82	deg	-1.16	deg	-1.13	deg

$(14.25 > 13.1)$. Nonetheless, the total endurance time is lower than before $(03h30 < 03h35)$.

A comparison between the values in the second and last columns show that, from the aerodynamic point of view, the wing designed in the conceptual phase has a very good performance. For the same flight conditions (speed, altitude and AoA), the introduction of twist did not improve the aerodynamic efficiency L/D of the wing: $\frac{14.25-14.24}{14.24} \times 100 \approx 0.07\%$ (which is negligible).

In the present chapter, the search for the optimal solution is performed with aid of gradient-based optimisation algorithms. One of their disadvantages is that they are likely to get 'trapped' at local optimal solutions as discussed in Section 2.3. With only one optimisation run, it is not possible to be certain that the solution found is, in fact, a global optimum. It could be possible that there exists a solution with higher weight $W_0 + W_s$ and also a higher L/D such that the endurance is higher. Solutions with lower weights than the one found are not possible since the structural design variables are already the lower bounds of the design space $\mathcal{D}$. To improve the likely-hood of the found optimal solution being a global one, more optimisation runs were performed with different initial skin and spar thicknesses as well as with different twist distributions. There were no significant differences in the results visualised, therefore they are omitted in the document.

5.5.2 Using Taper

In the previous problem, the geometric twist was used as design variable in order to maximise the endurance time. From a manufacturing point of view, it would be easier to construct a wing with no twist and an adequate taper than a straight wing with variable twist along the wingspan. Furthermore, until the current section, the baseline wing planform was kept fixed. The wing still has two distinct segments: one rectangular with a chord of 0.399 m and another tapered with $\lambda = 0.55$. To optimise the aerodynamic performance of the UAV, the only parameter susceptible of change was the geometric twist of each spanwise wing section.

In the present subsection, and in the following ones, the baseline UAV is improved by giving freedom to geometric planform variables. In this particular one, freedom is given to the taper ratio λ, which is going to change the chord linearly from 'root' to tip. The wing twist angle is kept fixed at 0^o for the entire surface. The rest of the optimisation problem is the same as before with the same tolerances and solver settings.

Although the planform shape is sought to be improved, the overall shape of the wing is to remain similar. I.e, it should have one straight rectangular section followed by a tapered one where the chord decreases linearly from the end of the inboard segment until the wing tip. The most recent OpenAeroStruct version (2.3.1 at the time of writing) does not have any capability that allows the user to modify the chord linearly from one desired wing spanwise position until its tip:

- Using control points for the chord to produce a smooth distribution is useful when defining arbitrary chord distributions, however, it is not possible to define precisely where these control points are placed in the aerodynamic mesh nodes. If, for example, two control points are used to manipulate the chord then the result would be a linear variation of the chord between root and tip (1st control point would be placed at the tip and the 2nd at the root). Therefore, this approach is not useful to modify wings composed of two distinct segments;
- Using the already available taper function to manipulate the lifting surface would change the chord from the root until the tip, which would be appropriate if the baseline wing was rectangular or simply tapered, which is not the case.

As such, the new *taper* function capable of tapering the wing from any arbitrary position until the tip is used, as described in Sub-section 4.4.2.

In order to define the wing taper, two parameters are needed: the taper ratio, λ which is simply the ratio between the chord at the tip, c_{tip} and the chord at the root, c_{root} and the offset, which indicates the position at which the taper will start. This offset must be given as a percentage with respect to the wing semi-span. Although the location of the starting point for the taper is an user-given input, it cannot be used as design variable in the optimisation problem. To overcome the limitation related to the taper offset, and to gain insight on what is the impact that the starting location of the taper would have on the aero-structural performance of the wing, instead of running only one optimisation with the taper ratio as design variable, a total of three cases are assessed, each with a different offset:

a) Taper applied from 37.5 % of the semi-span until the tip;

b) Taper applied from 30.0 % of the semi-span until the tip;

c) Taper applied from 45.0 % of the semi-span until the tip.

In each case, λ is free to change from iteration to iteration in the optimisation problem. It has lower and upper bounds of 0.1 and 1.0 respectively, for all three cases. The first case, a) with $\lambda = 0.55$ corresponds to the baseline solution's wing planform.

For all three cases the optimiser found an optimal solution that respects all constraints imposed (Scipy Exit Mode: 0). For both case a) and b) 5 iterations and 16 min were required, and for case c) only 4 iterations and 13 min were necessary.

The normalised lift distribution and the aggregated von Mises stresses along the wingspan for all three cases are represented in Figure 5.10. The wing skin and spar thicknesses are not shown because they are equal and constant for all cases: $t = 0.6\,mm$, which corresponds to the lower bound defined for both of these design variables. The optimal results found with the different offsets are presented in Table 5.8.

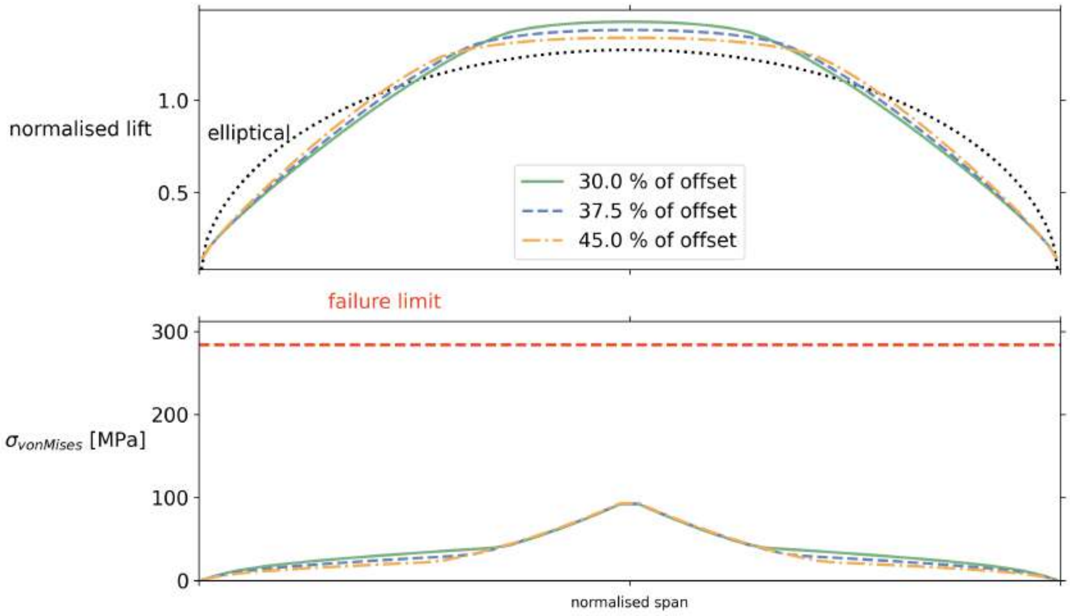


Figure 5.10: Optimisation results for taper with different offsets

Table 5.8: Comparison between optimal solutions using taper with different offsets

	30.0 % Offset		37.5 % Offset		45.0 % Offset	
Parameter	Value	Unit	Value	Unit	Value	Unit
Endurance	04:07	h:min	04:00	h:min	03:54	h:min
C_L	0.617	-	0.599	-	0.583	-
C_D	0.042	-	0.042	-	0.041	-
L/D	14,77	-	14.41	-	14.07	-
MTOW	19.07	kg	19.14	kg	19.20	kg
W_s	2.45	kg	2.52	kg	2.58	kg
α	4.01	deg	3.64	deg	3.29	deg
$\alpha_{6.0g}$	13.58	deg	12.89	deg	12.26	deg
α_i	-1.26	deg	-1.17	deg	-1.08	deg
λ	0.1	-	0.1	-	0.1	-
S_{wing}	1.09	m^2	1.15	m^2	1.20	m^2

From Table 5.8, it is clear that the best solution in terms of the objective function alone is the one with the offset set to 30% which has an endurance of 04h07. This solution allows the greatest reduction of W_s as the chord is reduced for a larger portion of the semi-span; and it is also the one with the highest L/D.

After analysing the results, it was found that λ was at the lower bound, $\lambda = 0.1$, independently of the value given to the taper offset.

In Figure 5.10, four distinct lift distribution curves are clearly identified, one curve for each taper offset value and one corresponding to the theoretical elliptic lift distribution. All three curves have a similar shape: lift is higher than the elliptic near the wing root; it decreases monotonically towards the tips, becoming lower than the elliptic at (approximately) half semi-span. In the second plot of Figure 5.10, it is observed the impact that the taper offset has on the stresses along the wing. All three curves are coincident near the wing root which is expected because all wings are the same until 30% of the semi-span. All have a change in the slope $\frac{\partial \sigma}{\partial y}$ due to the chord reduction along the wingspan. It changes exactly at the offset specified for each case. For all three cases, the wing is never near the failure limit (the failure constraint is inactive). Solutions with lift distributions other than the elliptical are expected, again the goal is to maximise endurance, which in our model can be accomplished by either increasing L/D, decreasing W_s or a combination of both.

Other common trend, observable in Table 5.8, is that the lower the wing area, the higher the required α and the higher the efficiency L/D (14.77 for the case where the offset is 30% and 14.07 for the case where offset is 45.0%). If these new results are compared to the ones obtained previously with the wing twist only as design variable (Subsection 5.5.1), the same trend is again evident. With an offset of 30% $\alpha = 4.01^o$, which is higher than the obtained $\alpha = 2.43^0$ for the optimal solution found with twist, and L/D is now 12.7% higher, even though the lift distribution of Figure 5.10 is not as close to the elliptic as the previous one (refer to Figure 5.8). This suggests that the angle of attack α and wing area S_{wing} have a stronger impact on the objective function than the aerodynamic design variables.

Endurance is being evaluated at the cruise flight point where $L = W$ for the UAV to be in level flight. So, the numerator in $\frac{L}{D}$ is fixed by the total weight of the vehicle. The total lift coefficient C_L is not necessarily the same. Writing lift as a function of the dynamic pressure and lift coefficient C_L the previous equilibrium condition is re-written as

$$\frac{1}{2}\rho V_\infty^2 S_{ref} C_L = W. \tag{5.4}$$

The chord of the wing is allowed to change, hence now the area of reference S_{ref} is not fixed. This explains why the obtained C_L values for solutions with almost the same total weight ($\approx 19.1\,kg$) are different. Total drag, on the other hand, can be altered. The closer the lift distribution curve is to the elliptic, the lower the induced-drag component is. Different lift distributions $L'(y)$ over the wingspan will contribute differently to the total drag D even if the total lift force $L = \int_{-b/2}^{b/2} L'(y)dy$ being generated is the same. The total drag for a lifting surface is calculated as

$$D = \frac{1}{2} \cdot \rho \cdot V_\infty^2 \cdot S_{ref} \cdot \left(C_{D_0} + C_{D_i} + C_{D_v} + C_{D_w}\right), \tag{5.5}$$

where C_{D_0} is used to account for the drag contribution of other components of the aircraft that are not modelled in the analysis, such as the vertical rotors, fuselage, double boom and landing gear; C_{D_i} is the induced drag component (due to the lift) that is calculated through the sum of the free-stream component of the aerodynamic forces acting on the panels on the VLM mesh; C_{D_v} is the viscous drag which is calculated using flat-plate-based empirical formulations (Raymer, 1992, sec. 12.5.3) and then adjusted using a form factor to account for the lifting surface's shape; finally, C_{D_w} is the wave drag based on the Korn equation (Malone & Mason, 1995).

The designed UAV has a cruise Mach of approximately 0.06, which implies that the wave drag contribution is null, $C_{D_w} = 0$. In the above formulation, of all three non-null contributions, only C_{D_i} and C_{D_v} are susceptible to change. But, since C_{D_v} is mainly affected by the thickness-to-chord ratio of the airfoil which is kept fixed and the local Re, significant changes to this variable are not expected; C_{D_0} is a user-given input, also kept fixed.

The designed UAV is a lift+rotor configuration where the vertical mode rotors are fixed and cannot be 'hidden' from the incoming airflow. Hence, the base drag component which does not change with respect to α is high, thus $C_{D_i} << (C_{D_0} + C_{D_v})$. The result is that variations of this drag component will have little effect on the overall total drag. That explains why α has more impact on the efficiency L/D than the lift distribution $L'(y)$ over the wingspan. Higher α implies higher lift. In turn, higher lift implies higher C_{D_i}. However, because $C_{D_0} >> C_{D_i}$ the increase in lift (or C_L) is higher than the increase in C_D. Therefore, higher α implies higher L/D.

Hence, the optimiser will try to reduce both the weight of the structure as much as possible as well as the wing area S_{ref} without violating any of the constraints. Because now it is also possible to change the chord of each section through taper (which decreases not only the width of the wingbox structure and therefore its weight, but also S_{ref}), the optimiser reduces λ to the lowest possible value, which corresponds to $\lambda = 0.1$ (the lower bound defined for the design variable) since the maximum stresses are small compared to the yield limit.

To confirm this hypothesis, several optimisations were performed for each of the three mentioned cases, with different lower bounds for λ. For the lower bounds tested ($\lambda_{\text{lower}} \in [0.1, 0.6]$), the optimal results were always obtained with them. Not in a single case did the optimiser choose a different value for λ other than the lower bound.

Although the best solution is the one that corresponds to the lowest λ, it does not imply that is a viable one. The reduction of wing area could be such that stall might occur. (stall speed was fixed at 28 kts in the conceptual phase). As mentioned, C_L is not fixed: as it can be seen in Table 5.8, the reduction of wing area S_{wing} is such that even though W reduces, C_L increases for the equilibrium condition given by Eq.(5.4) to be respected. This is usually accomplished with increases in angle of attack α. However, C_L cannot increase indefinitely, it has a maximum value $C_{L_{max}}$ which is achieved with $\alpha = \alpha_{stall}$. If α further increases, the wing will eventually stall. $C_{L_{max}}$ is dictated by the wing airfoil characteristics and by its overall geometry. Consequently, if this trend persists, then there is no assurance that the obtained lift coefficient is in fact reasonable, i.e, $C_L < C_{L_{max}}$. Stall is further discussed in a later subsection (5.5.5). Moreover, for $\lambda = 0.1$, the resulting wing tip chord is very small, $c_{tip} = 0.1 \cdot 0.399 \approx 4\, cm$ which is not viable from the manufacturing point of view.

5.5.3 Using Taper and Chord at Root

It was showed that it was possible to maintain the geometry of the baseline wing and to change its taper ratio from a given offset until the tip. It was also showed that λ could be given as a design variable and that the optimiser would push this variable to the user-set lower bound.

In this present subsection, more freedom is given to the optimiser by adding another design variable: the wing root chord. The intent is to allow the optimiser to vary the chord of the rectangular section and to use λ to define each section chord from the end of the inboard segment until the tip of the wing.

As mentioned in Sub-section 5.5.2, it is possible in OpenAeroStruct to produce arbitrarily chord distributions through the usage of control points. By changing the number of control points and their values, the chord distribution along the span changes. The control points influence a bspline interpolation that will produce an array with y values, y being the number of span-wise nodes in the aerodynamic mesh. Each value is a scaling factor that will increase or decrease the chord at each section of the wing, i.e, if the numerical value is one then the chord does not change, if it is two then the chord will double in length.

If one chord control point is used, the resulting interpolation is an array with the same scaling factor for all sections which would maintain $\lambda = 0.55$ (as defined in the baseline). If taper is used together with one chord control point, it is possible to change the chord of the inboard section of the wing and use λ to reduce the chord linearly until the tip.

The optimisation problem, as defined in Table 5.2, with taper and one chord control point instead of wing twist is solved.

The optimiser was not able to find a solution that satisfied the convergence criterion set previously

(1×10^7) in the 200 iterations, Scipy's exit mode: 9. Despite convergence not being achieved, the problem constraints were respected. A total of 22:42 h:min were needed to solve the problem.

Table 5.9 contains the results of the optimisation.

Table 5.9: Optimisation results obtained with taper and chord at root

	Optimised		Baseline	
Parameter	Value	Unit	Value	Unit
Endurance	05:29	h:min	03:29	h:min
C_L	0.868	-	0.601	-
C_D	0.045	-	0.042	-
L/D	19.1	-	14.24	-
MTOW	18.57	kg	21.6	kg
W_s	1.95	kg	4.98	kg
α	10.0	deg	3.28	deg
$\alpha_{6.0g}$	23.66	deg	11	deg
α_i	-4.66	deg	-1.13	deg
C_{root}	0.196	m	0.399	m
λ	0.23	-	0.55	-
S_{wing}	0.596	m^2	1.373	m^2

Even though the optimiser could not verify the convergence criterion set, the objective function improved 33.2% from last solution: the best obtained endurance in Subsection 5.5.2 was 04h07 and is now 05h29.

The wing area is now even smaller than before, which forces α to further increase (in Subsection 5.5.2 was as high as 4.0° and now is 10.0°). The result is a higher L/D and lower W_s.

Because no stall condition was imposed so far, the parameter that is limiting how much wing area can be reduced by the optimiser is α. It is not possible to have $S_{wing} < 0.596\,\text{m}^2$ because in order to have equilibrium α would have to increase. But it is already at the upper bound defined, $\alpha = 10.0^o$.

5.5.4 Using Taper, Chord at Root and Span

In this present section, the wing span is added as a design variable, allowing it to change the length of the span while keeping the relative position of each node in the aerodynamic mesh constant, e.g, if the third spanwise node is located at 50% of the semi-span, it will remain at exactly 50% of the semi-span after the transformation. This implies that the number of nodes remains exactly the same. As such, if the span is increased, there is the possibility that the number of panels being used is no longer enough to respect the convergence criteria defined in Section 5.2. For small variations in span, though, it is not expected that the numerical error changes significantly.

The span lower and upper bounds were set to 3.0 and 5.0 m, respectively. Similar to the previous section, the bounds of the other design variables added so far remained the same.

The problem converged in 34 iterations to the optimal solution without violation of the constraints (Scipy's exit mode: 0). The results obtained are shown in Table 5.10 together with the baseline for ease of comparison.

Table 5.10: Optimisation results using taper, chord at root, and span

	Optimised		Baseline	
Parameter	Value	Unit	Value	Unit
Endurance	05:41	h:min	03:29	h:min
C_L	0.890	-	0.601	-
C_D	0.044	-	0.042	-
L/D	20.4	-	14.24	-
MTOW	19.1	kg	21.6	kg
W_s	2.43	kg	4.98	kg
α	10.0	deg	3.28	deg
$\alpha_{6.0g}$	23.9	deg	11	deg
α_i	-4.87	deg	-1.13	deg
C_{root}	0.166	m	0.399	m
λ	0.1	-	0.55	-
Span	5.0	m	4.0	m
S_{wing}	0.598	m^2	1.373	m^2

With the increase of the problem complexity comes the increase in computational cost, now 17h32. Nevertheless, at the same time, this additional complexity allowed the optimiser to explore a larger design space, and, as consequence find a better solution with a total endurance time of 05h41.

The same trends with respect to the design variables are verified: reduction of the chord so that both W_s and L/D increase; reduction of both skin and spar thicknesses to the set lower bound of 0.6 mm. The span of the wing increased from 4.0 to 5.0 m. The result of the geometric transformations is again a wing with a smaller area which in turn requires a high α to ensure equilibrium. Similarly to the previous solution, α is at the upper bound. A downside from the reduction of the chord is the increase of the required α to a point where it could be higher than α_{stall}. If that is the case then the wing will stall, a result which is not acceptable.

5.5.5 Stall Considerations

Up until this point, no stall conditions were verified. It was only imposed $L = W$ in both cruise and load flight conditions. Although for the later condition this is enough as it is only an extreme case used to safely size the structure of the wing, in the former condition it is necessary that the designed wing can generate that amount of lift without stalling.

The stall phenomenon can be verified through several ways: α_{stall} could be verified experimentally for the cruise conditions and then the α upper bound in cruise could be set to that value; or the maximum $C_{l_{2D}}$ for the wing airfoil (SG-6042) could be determined and then a new constraint in the optimisation problem could be added such that

$$C_l < C_{l_{2D}}, \tag{5.6}$$

C_l being the lift coefficient for each span-wise section along the wingspan. Due to the additional constraint that the second approach implies, it will be computationally more expensive than the first. However, the time invested in the building and testing process of the first is significantly higher than the time that would be saved in the optimisation problem by not including an additional constraint. As such, although the second approach is computationally more expensive, it is overall cheaper in terms of both cost and total time.

To use the second approach to verify if stall occurs, $C_{l_{2D}}$ of the SG-6042 airfoil is necessary. To calculate it, XFOIL (Drela, 1989), a software that uses high-order panel methods to conduct analysis and design of subsonic isolated airfoils is used. The airfoil polar $C_l - \alpha$ is obtained for different Re numbers ranging between 200 × 10^3 and 500 × 10^3. Note that in cruise: $Re = \frac{\rho V \bar{c}}{\mu} \approx 412.5 \cdot 10^3$ at the wing root. Ncrit was set to 5 and later to 9 (this parameter is used to predict transition from laminar to turbulent flow). From the aerodynamic polars, the maximum $C_{l_{2D}}$ is observed to be between 1.4 and 1.5 (depending on the Re number and Ncrit used).

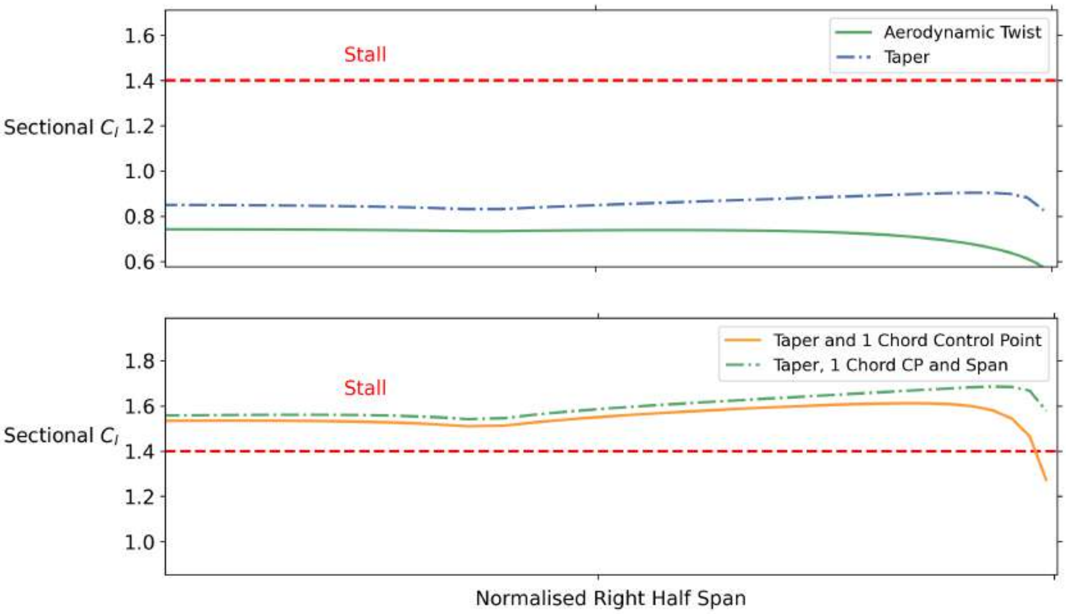
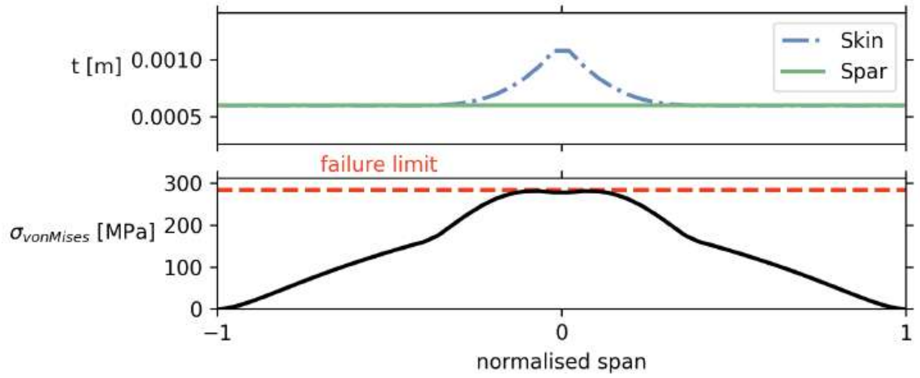


Figure 5.11: Sectional lift coefficients of different optimal solutions at cruise conditions

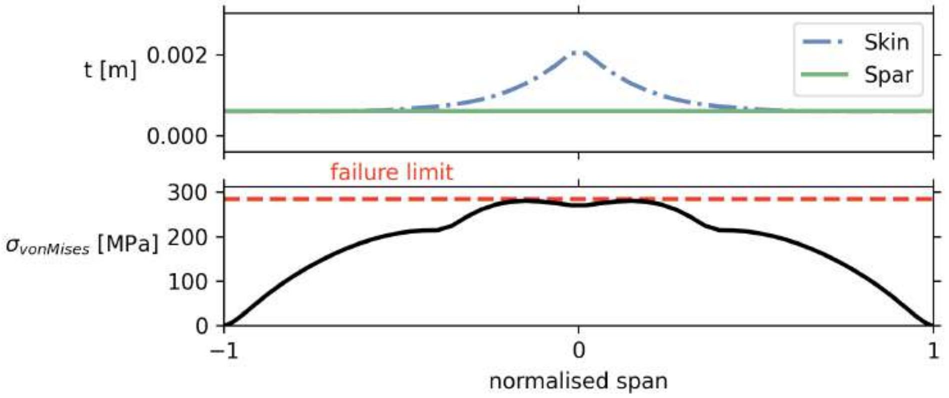
Figure 5.11 shows the lift coefficient C_l for each span-wise section of the right half wing at cruise conditions. Each curve was obtained from the solutions found in the previous sections 5.5.1-5.10 with a taper offset of 37.5 %. the maximum $C_{l_{2D}}$ is identified in red, which is taken as 1.4 for safety concerns. Above this limit lies the stall region.

As it can be seen in the first subplot of Figure 5.11, neither solution stalls at cruise. In the second subplot though, stall occurs for every section along the wingspan of each solution. Since the last two solutions are not valid as they are, the stall constraint given by Eq.(5.6) is added to the cruise flight point and both optimisation problems are solved again.

For both optimisation problems with the added C_l constraint the optimiser was able to find optimal solutions that respected both the objective function and constraint tolerances (Scipy exit mode: 0). With taper and one chord control point, it needed 12 iterations and 0h38 to converge. With taper, one chord control point and span a total of 28 iterations and 01h28 were needed.



(a) Optimised parameter distribution along wingspan, obtained with taper, chord and C_l constraint



(b) Optimised parameter distribution along wingspan, obtained with taper, chord, span and C_l constraint

Figure 5.12: Optimisation results with the addition of C_l constraint

The spar and skin thicknesses, and the von Mises stress distributions along the wingspan are shown in Figure 5.12. It is observed that the stress limit is respected and that the skin thickness has a maximum of 1.0 mm at the wing root.

Figure 5.13 shows the sectional $C_l s$ of both solutions. It is observed that these values do not exceed the maximum allowed, $C_{l_{2D}} = 1.4$, and that the maximum is reached near 75% of the semi-span where $C_l = C_{2D}$, which means that the cruise speed set at 38 kts is equal to the stall speed.

A comparison between the optimal solutions found with and without the C_l constraint is provided in Table 5.11.

It is verified that the addition of the constraint is now limiting the reduction of the wing area S_{wing}: not only S_{wing} is higher when the constraint is present but also α is no longer at the upper bound. S_{wing} increases as result of an increase of the planform design variables: c_{root} and λ. The presence of the constraint also decreased the total endurance in both cases.

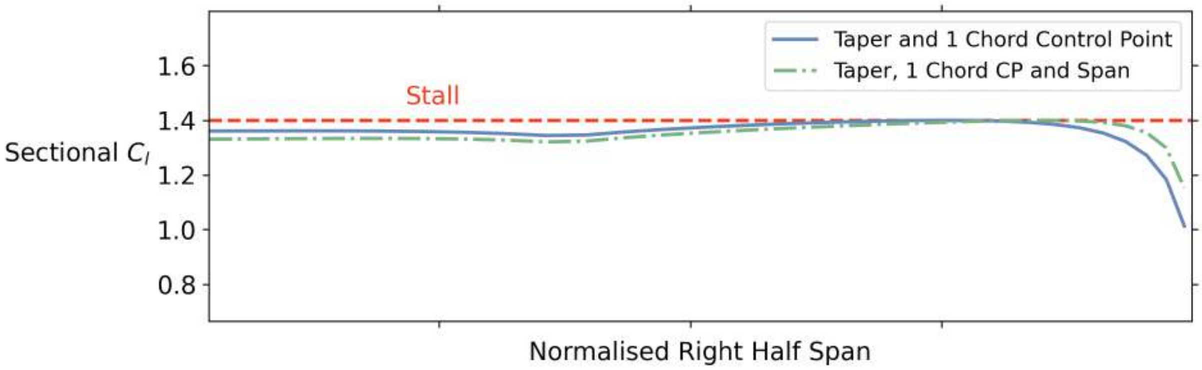


Figure 5.13: Sectional lift coefficients at cruise with C_l constraint

5.5.6 Final Optimal Solution

In Subsection 5.5.5, a constraint on the maximum allowable sectional lift coefficients of the wing was imposed. As a result, the planform dimensions of the wings found in Subsections 5.5.3 and 5.5.4 increased until that constraint was satisfied. Still, the wing area S_{wing} of both is approximately half of the baseline.

The baseline wing was defined in the conceptual phase of the project where the stall constraint

$$W/S_{wing} \leq \frac{1}{2}\rho V_{stall}^2 C_{L_{max}} \tag{5.7}$$

was respected with $V_{stall} = 28\,kts$. $C_{L_{max}}$ could not be calculated at the time and so an estimate was used.

The optimisation problem modeled in OpenAeroStruct, refer to Table 5.2, uses two flight points: one for the cruise where the endurance is to be maximised; and another one for a 6.0 g manoeuvre to safely size the structure of the wing. With the addition of the cruise constraint given by Eq.(5.6), it was ensured that at cruise the wing would not stall. However, if the flight speed of the UAV reduces to 28 kts there is no assurance that the wings produced by the optimiser in Subsections 5.5.1 to 5.5.4 can sustain level flight. In fact, for the cases where the C_l constraint is active, they probably cannot provide enough lift as the sectional $C_l s$ are already close to the maximum $C_{L_{2D}}$ of the airfoil. The lack of a 3rd flight point to check for stall at the lower speed of 28 kts explains why it was possible for the optimiser to reduce S_{wing} by 50%.

To ensure the optimal UAV is capable of sustaining level flight at the defined stall conditions, a third AeroStructPoint, i.e, flight condition, is created. Similarly to cruise, no loads are applied and the constraint $L = W$ must be verified. In addition, the constraint given by Eq.(5.6) must also be respected to ensure that the obtained sectional lift coefficients are not higher than the maximum 2D lift coefficient of the wing's airfoil.

The problem was solved using the same tolerances and solver settings as defined in Section 3.3. It took 25 iterations and 02h00 for the optimiser to find a converged solution without violation of the constraints (Exit mode: 0).

Table 5.12 shows the results obtained alongside the baseline for comparison. The total endurance

Table 5.11: Effect of the addition of an active C_l constraint on the previous solutions

a Solution obtained with taper and chord

	With C_l constraint		Without C_l constraint	
Parameter	Value	Unit	Value	Unit
Endurance	05:13	h:min	05:29	h:min
C_L	0.813	-	0.868	-
C_D	0.044	-	0.045	-
L/D	18.3	-	19.1	-
MTOW	18.63	kg	18.57	kg
W_s	2.02	kg	1.95	kg
α	8.40	deg	10.0	deg
$\alpha_{6.0g}$	20.8	deg	23.7	deg
α_i	-3.8	deg	-4.7	deg
C_{root}	0.210	m	0.196	m
λ	0.37	-	0.23	-
S_{wing}	0.677	m^2	0.596	m^2

b Solution obtained with taper, chord and span

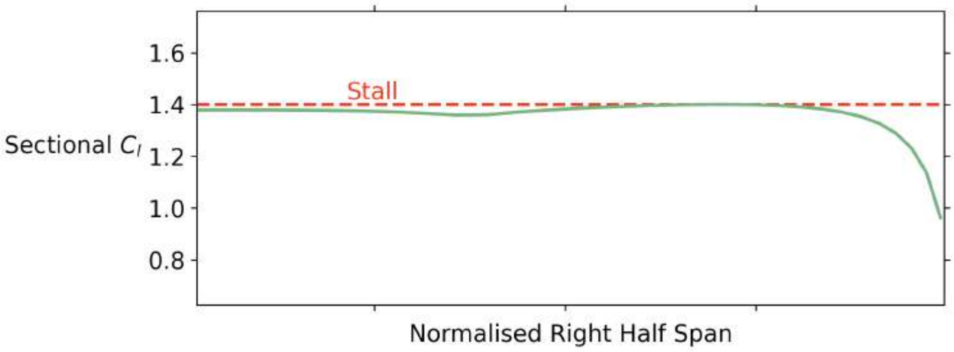
	With C_l constraint		Without C_l constraint	
Parameter	Value	Unit	Value	Unit
Endurance	05:20	h:min	05:41	h:min
C_L	0.814	-	0.890	-
C_D	0.043	-	0.044	-
L/D	19.0	-	20.4	-
MTOW	19.0	kg	19.1	kg
W_s	2.36	kg	2.43	kg
α	8.0	deg	10.0	deg
$\alpha_{6.0g}$	20.1	deg	23.9	deg
α_i	-3.6	deg	-4.9	deg
C_{root}	0.186	m	0.166	m
λ	0.25	-	0.1	-
Span	4.9	m	5.0	m
S_{wing}	0.696	m^2	0.598	m^2

time increased to 04h14, which represents a 21% increase when compared to the baseline. This increase in endurance is accomplished with both an increase in total aircraft C_L and reduction of C_D. Since α also increased it can be concluded that this wing is aerodynamically more efficient for the drag formulation used. The weight of the structures decreased which also contributed to the increase of the endurance. With regard to the planform design variables c_{root}, λ and b, the trend is again to decrease the wing area S_{wing} and to increase the wing's AR, which contributes to the reduction of the induced drag component in Eq.(5.5).

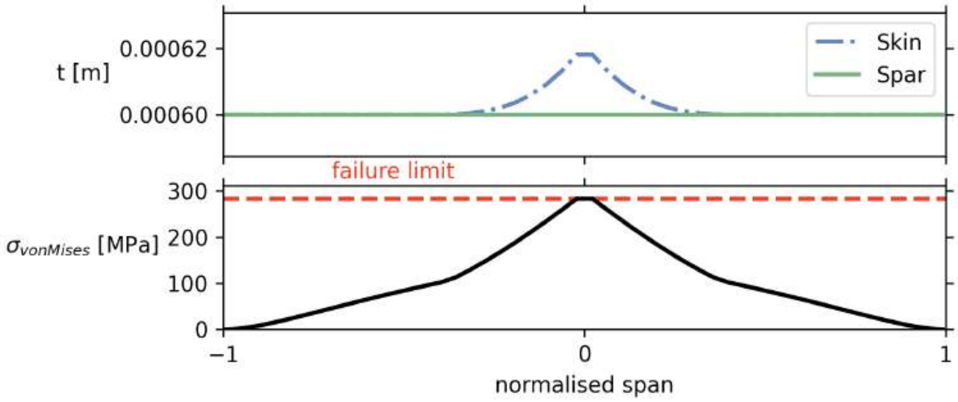
The sectional lift coefficients for the third flight point, stall, are given in Figure 5.14(a). The maximum

Table 5.12: Optimisation results obtained with 3 flight points

	Optimised		Baseline	
Parameter	Value	Unit	Value	Unit
Endurance	04:14	h:min	03:29	h:min
C_L	0.630	-	0.601	-
C_D	0.041	-	0.042	-
L/D	15.2	-	14.24	-
MTOW	19.0	kg	21.6	kg
W_s	2.41	kg	4.98	kg
α	3.9	deg	3.28	deg
$\alpha_{6.0g}$	12.6	deg	11	deg
α_i	-1.5	deg	-1.13	deg
C_{root}	0.272	m	0.399	m
λ	0.44	-	0.55	-
Span	4.7	m	4.0	m
S_{wing}	1.057	m^2	1.373	m^2



(a) Sectional lift coefficients at V_{stall}



(b) Thickness distribution and von Mises stresses

Figure 5.14: Parameter distribution along the wingspan using three flight points

C_l is reached at 50-75% of the semi-span with a value of 1.4 which corresponds to the value of $C_{l_{2D}}$ in Eq.(5.6). Therefore, stall will first occur at this portion of the wing, then at the inboard portion (between the root and 50% of the semi-span) and lastly at the tip, as desired to keep aileron control.

Finally, Figure 5.14(b) shows the thickness distribution of both wing skin and spar as well as the aggregated von Mises stresses along the wingspan. The spar thickness is constant and equal to the lower bound of 0.6 mm defined but the skin is slightly thicker at the root of the wing, $t = 0.62mm$. The failure limit was never reached and the maximum occured once more at the wing root.

5.6 Comparison of Different Optimal Solutions

Over the course of this chapter, the optimisation problem has been updated and refined to become not only more realistic but also more complex with the addition of design variables and constraints. Here, a comparison between the most relevant optimal solutions will be made.

Table 5.13: Comparison between optimal solutions

	W. Twist		W. Planform Design Variables		W. Three Flight Cases	
Parameter	Value	Unit	Value	Unit	Value	Unit
Endurance	03:35	h:min	05:20	h:min	04:14	h:min
C_L	0.536	-	0.814	-	0.630	-
C_D	0.041	-	0.043	-	0.041	-
L/D	13.1	-	19.0	-	15.2	-
MTOW	19.4	kg	19.0	kg	19.0	kg
W_s	2.8	kg	2.36	kg	2.41	kg
α	2.4	deg	8.0	deg	3.9	deg
$\alpha_{6.0g}$	10.6	deg	20.1	deg	12.6	deg
α_i	-0.8	deg	-3.6	deg	-1.5	deg
C_{root}	0.399	m	0.186	m	0.272	m
λ	0.55	-	0.25	-	0.44	-
Span	4.0	m	4.9	m	4.7	m
S_{wing}	1.373	m^2	0.696	m^2	1.057	m^2

Some of the results found in Subsections 5.5.1 to 5.5.6 are summarised in Table 5.13. The first results were found in Subsection 5.5.1 with only one aerodynamic design variable: twist. Doing so, the original planform of the wing was kept fixed. Through the introduction of wash-out by the optimiser, it was possible to obtain a lift distribution curve $L'(y)$ almost elliptical, which is an indicator of good aerodynamic performance as the induced-lift drag component is minimum.

Next are the optimisation results found in Subsection 5.5.5 with all planform design variables (λ, C_{root} and b) together with the addition of a stall constraint. This condition was imposed at the cruise flight point to ensure that the wing would not stall.

The last results presented were obtained with the addition of a third flight point to verify that at the defined stall speed V_{stall}, the wing would be able to sustain level flight, these were described in detail in Subsection 5.5.6.

Of all three, the first one achieved the lowest endurance. This solution has the highest weight MTOW and lowest L/D, therefore it is the least efficient from both the aerodynamic and structural point of view. It is also the one with largest wing area S_{wing}.

The highest value of endurance was obtained with all three planform design variables and the C_l constraint. The results of this optimisation correspond to the 2nd set of results in Table 5.13. The aerodynamic efficiency L/D obtained is the highest and the value of S_{wing} is the lowest. Although the endurance was the highest for this particular case, the wing produced in this solution cannot sustain level flight at the desired V_{stall}. In fact, the UAV with this optimal wing cannot fly if $V < V_{cruise}$.

Lastly, the 3rd set of results of Table 5.13 correspond to the best solution found when the stall speed is respected. Since the S_{wing} is lower than the original planform's, it is concluded that the original wing was slightly oversized.

Chapter 6

Conclusions

The purpose of this final chapter is twofold: the first is to provide a summary of how the objective and deliverables were achieved and the second is to make some recommendations for future work.

6.1 Achievements

The first achievement of this work was the the sizing of a small UAV for surveillance operations in the Portuguese Air Force, refer to Chapter 3. This was accomplished in the early conceptual phase of the project by using an in-house developed software to estimate the vehicle's MTOW and Endurance, together with a suitable *Open Source* framework to solve multi-objective optimisation problems. Doing so in the early stages of the project allowed to estimate the impact of some general design decisions on both objectives (weight and endurance), which contributed to a more informed decision process and to the adjustment of the project requirements.

Next, some developments with OpenAeroStruct were made to allow a definition of planform shapes and to enable the estimation of the endurance for an electric propeller driven aircraft. With the intent of using the endurance as an objective function and the added *taper_with_offset* as design variable in the optimisation problem, analytic partial derivatives were provided to the OpenMDAO model, to improve the computational efficiency for the gradient-based optimiser.

Afterwards, with the mentioned code developments it was possible to search for improvements to the current baseline model which had been designed during the conceptual phase of the project. The optimisation problem gradually grew in complexity. First, only twist was used as aerodynamic design variable. No significant improvements were obtained. The reduction in weight and the low AoA suggested that flying with speeds lower than the one defined, 38 kts, might increase the endurance of the vehicle. Because the speed was to be fixed, planform design variables were introduced in the problem.

The twist was set to 0 deg and instead, taper was used with different offsets. After the trends had been analysed, the chord at the inboard section of the wing is also allowed to change and the optimisation problem is again solved. With both the inboard chord and the taper allowed to change, span is added to the set of design variables. Since the planform of the wing was allowed to change some important trends were observed: wing area was being reduced and the aspect ratio was increased. The lack of a stall constraint allowed the optimiser to take advantage of the problem formulation. This resulted in configurations that were flying with $\alpha = 10.0^o$ to ensure equilibrium, which was not possible.

Later, the optimisation problem was re-formulated to take stall into account. For some solutions, the constraint was active in cruise and, therefore, the optimiser increased the wing size to satisfy the constraint.

Finally, to enforce that the stall speed of 28 kts was respected, a third flight point was added with the equilibrium and stall constraints and no loads applied. From this optimisation, a smaller wing with a higher span was obtained which resulted in a 21% increase in the expected endurance of the UAV when compared to the baseline. Other solutions with extended endurance times were found but they do not satisfy this constraint.

6.2 Future Work

During this work, the shape of the wing was optimised with respect to endurance. Still, there is room for further investigation as OpenAeroStruct is not only capable of modelling but also perform optimisation with respect to multiple lifting surfaces. Therefore, it is possible to add design variables to modify the tail and possibly search for even better solutions.

Lastly, independently if improvements can be made or not, the use of higher fidelity software is recommended to validate the current design. After validation, experimental tests with small scale models could be performed at the Portuguese Air Force Research Centre's wind tunnel before the manufacture of a prototype is attempted.

Bibliography

Alves, B., Coelho, V., Silva, P., Marta, A., Afonso, F., Sá, P., … Caetano, J. (2021, July). Design of a hydrogen powered small electric fixed-wing uav with vtol capability. In A. C. Marta & A. Suleman (Eds.), *International conference on multidisciplinary design optimization of aerospace systems* (p. 290-304). Lisbon, Portugal: Aerobest 2021. (ISBN: 978-989-99424-8-6)

Anderson, J. D. (2017). *Fundamentals of aerodynamics.* McGraw-Hill Education.

Austin, R. (2011). *Unmanned aircraft systems: UAVS design, development and deployment* (Vol. 54). John Wiley & Sons.

Belegundu, A. D., & Chandrupatla, T. R. (2019). *Optimization concepts and applications in engineering.* Cambridge University Press.

Blank, J. (2020). *pymoo - termination criteria.* Retrieved July 22, 2021, from `https://pymoo.org/interface/termination.html`

Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, *8*, 89497-89509. doi: 10.1109/ACCESS.2020.2990567

Blank, J., & Deb, K. (2020). A running performance metric and termination criterion for evaluating evolutionary multi- and many-objective optimization algorithms. In *2020 IEEE congress on evolutionary computation (CEC)* (p. 1-8). doi: 10.1109/CEC48606.2020.9185546

Caetano, J. V., & Morgado, J. P. (2019). Portuguese air force uav development, certification and operations in maritime environment. In *Unmanned air vehicles: Technological challenges, concepts of operations and regulatory issues.* doi: 10.14339/STO-EN-AVT-274

Chauhan, S. S., & Martins, J. R. R. A. (2019). Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using openaerostruct. In H. Rodrigues et al. (Eds.), *Engopt 2018 proceedings of the 6th international conference on engineering optimization* (pp. 418–431). Cham: Springer International Publishing.

Coelho, V. (2021). *Aerodynamic detailed design of an unmanned aerial vehicle with a fuel cell powered energy system* (Unpublished master's thesis). Academia da Força Aérea, Sintra, Portugal.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms.* MIT press.

Cutrona, L. (1990). Synthetic aperture radar. *Radar handbook*, *2*, 2333–2346.

de Miguel, N., Acosta, B., Baraldi, D., Melideo, R., Ortiz Cebolla, R., & Moretto, P. (2016). The role of initial tank temperature on refuelling of on-board hydrogen tanks. *International Journal of Hydrogen Energy*, *41*(20), 8606-8615. doi: 10.1016/j.ijhydene.2016.03.158

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. USA: John Wiley & Sons, Inc.

Deb, K., & Agrawal, S. (1999). A niched-penalty approach for constraint handling in genetic algorithms. In *Artificial neural nets and genetic algorithms* (pp. 235–243). Vienna: Springer Vienna. doi: 10.1007/978-3-7091-6384-9_40

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197. doi: 10.1109/4235.996017

Drela, M. (1989). Xfoil: An analysis and design system for low reynolds number airfoils. In T. J. Mueller (Ed.), *Low reynolds number aerodynamics* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-84010-4_1

Drew, C. (2010, February 19). *Drones are playing a growing role in afghanistan.* Retrieved November 10, 2021, from `https://www.nytimes.com/2010/02/20/world/asia/20drones.html?smid=url-share`

Edwards, P., Kuznetsov, V., David, W., & Brandon, N. (2008). Hydrogen and fuel cells: Towards a sustainable energy future. *Energy Policy*, *36*(12), 4356-4362. (Foresight Sustainable Energy Management and the Built Environment Project) doi: 10.1016/j.enpol.2008.09.036

European Defense Agency. (2020, March). *First energy consultation forum project to receive eu funding.* Retrieved November 7, 2021, from `https://eda.europa.eu/news-and-events/news/2020/03/10/first-energy-consultation-forum-project-to-receive-eu-funding`

European Union. (2020, March). *Long-term low greenhouse gas emission development strategy of the european union and its member states.* Retrieved November 7, 2021, from `https://unfccc.int/documents/210328`

Giguère, P., & Selig, M. S. (1998, May). New Airfoils for Small Horizontal Axis Wind Turbines. *Journal of Solar Energy Engineering*, *120*(2), 108-114. doi: 10.1115/1.2888052

Gilli, M., & Winker, P. (2008, June). *A review of heuristic optimization methods in econometrics* (Swiss Finance Institute Research Paper Series No. 08-12). Swiss Finance Institute. Retrieved from `http://ssrn.com/abstract=1140655.`

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st ed.). USA: Addison-Wesley Longman Publishing Co., Inc. doi: 10.1023/A:1022602019183

Goldberg, D. E., Deb, K., & Clark, J. H. (1991). Genetic algorithms, noise, and the sizing of populations. *Complex systems*, *6*, 333–362. doi: 10.1016/B978-0-08-094832-4.50014-3

Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., & Naylor, B. A. (2019). OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, *59*(4), 1075–1104. doi: 10.1007/s00158-019-02211-z

Gudmundsson, S. (2013). *General aviation aircraft design: Applied methods and procedures.* Butterworth-Heinemann. doi: 10.1016/C2011-0-06824-2

Gundlach, J. (2014). *Designing unmanned aircraft systems: A comprehensive approach, second edition.* American Institute of Aeronautics and Astronautics. doi: 10.2514/4.102615

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., . . . Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. doi: 10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

Hwang, J. T., & Martins, J. R. R. A. (2018). A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives. *ACM Transactions on Mathematical Software*, *44*(4), Article 37. doi: 10.1145/3182393

Jasa, J. P., Hwang, J. T., & Martins, J. R. R. A. (2018). Open-source coupled aerostructural optimization using python. *Structural and Multidisciplinary Optimization*, *57*(4), 1815-1827. doi: 10.1007/s00158-018-1912-8

Jordan, B. R. (2019). Collecting field data in volcanic landscapes using small uas (suas)/drones. *Journal of Volcanology and Geothermal Research*, *385*, 231-241. doi: 10.1016/j.jvolgeores.2019.07.006

Katz, J., & Plotkin, A. (2001). *Low-speed aerodynamics* (Vol. 13). Cambridge university press.

Kelley, T. R. (2010). Optimization, an important stage of engineering design. *The Technology Teacher*, *69*(5), 18.

Kraft, D. (1988). *A software package for sequential quadratic programming* (Tech. Rep. No. DFVLR-FB 88-28). Koln, Germany: DLR German Aerospace Center – Institute for Flight Mechanics.

Kreisselmeier, G., & Steinhauser, R. (1979). Systematic control design by optimizing a vector performance index. *IFAC Proceedings Volumes*, *12*(7), 113-117. (IFAC Symposium on computer Aided Design of Control Systems, Zurich, Switzerland, 29-31 August) doi: 10.1016/S1474-6670(17)65584-8

Lambe, A. B., & Martins, J. R. R. A. (2012). Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, *46*(2), 273-284. doi: 10.1007/s00158-012-0763-y

Malone, B., & Mason, W. H. (1995). Multidisciplinary optimization in aircraft design using analytic technology models. *Journal of Aircraft*, *32*(2), 431-438. doi: 10.2514/3.46734

Marta, A. C. (2020). *Aircraft optimal design, msc course notes, instituto superior técnico.*

Martins, J. R. R. A., Alonso, J. J., & Reuther, J. J. (2005, Mar 01). A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, *6*(1), 33-62. doi: 10.1023/B:OPTE.0000048536.47956.62

Martins, J. R. R. A., & Ning, A. (2022). *Engineering design optimization*. Cambridge University Press.

Michalewicz, Z. (2013). *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media.

Morgado, J., & Sousa, J. (2009). O programa de investigação e tecnologia em veículos aéreos autónomos não tripulados da academia da força aérea. *IDN Cadernos*, *II Série*(4), 9-24. Retrieved from `https://www.idn.gov.pt/pt/publicacoes/idncadernos/Documents/2009/caderno4_II.pdf`

Newcome, L. R. (2004). *Unmanned aviation: a brief history of unmanned aerial vehicles*. American

Institute of Aeronautics and Astronautics. doi: 10.2514/4.868894

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2^{nd} ed.). Springer. doi: 10.1007/978-0-387-40065-5

Openaerostruct documentation- geometry creation and manipulation [Computer software manual]. (2018). Retrieved August 22, 2021, from `http://mdolab.engin.umich.edu/OpenAeroStruct/tutorials/geometry_manipulation.html`

Ravindran, A., Ragsdell, K. M., & Reklaitis, G. V. (2006). *Engineering optimization: Methods and applications* (2^{nd} ed.). John Willey & Sons, Inc.

Raymer, D. P. (1992). *Aircraft design: A conceptual approach*. American Institute of Aeronautics and Astronautics, Inc.

Reeves, C., & Rowe, J. E. (2002). *Genetic algorithms: principles and perspectives: a guide to ga theory* (Vol. 20). Springer Science & Business Media.

Rivers, M. B. (2019, June 14). NASA common research model: A history and future plans. American Institute of Aeronautics and Astronautics. (0) doi: 10.2514/6.2019-3725

Rudolph, G. (1999). Evolutionary search under partially ordered sets. *Dept. Comput. Sci./LS11, Univ. Dortmund, Dortmund, Germany, Tech. Rep. CI-67/99*.

Saaty, T. L., & Vargas, L. G. (2012). *Models, methods, concepts & applications of the analytic hierarchy process* (2nd ed.). Springer. doi: 10.1007/978-1-4614-3597-6

Schwefel, H.-P. (1987). Collective phenomena in evolutionary systems. In P. Checkland & I. Kiss (Eds.), *Problems of constancy and change - the complementarity of systems approaches to complexity* (p. 1025-1033). Budapest.

Silva, G. (2021). *Definição da arquitetura funcional de um veículo aéreo não tripulado equipado com o piloto automático veronte* (Unpublished master's thesis). Academia da Força Aérea, Sintra, Portugal.

Silva, J. (2017). *Design and optimization of a wing structure for a uas class i 145 kg* (Master's thesis, Academia da Força Aérea, Sintra, Portugal). Retrieved from `http://hdl.handle.net/10400.26/23156`

Silva, P. (2021). *Projeto detalhado do sistema propulsivo para um veículo aéreo não tripulado com uma célula de combustível de hidrogénio* (Unpublished master's thesis). Academia da Força Aérea, Sintra, Portugal.

Stetson, N., McWhorter, S., & Ahn, C. (2016). 1 - introduction to hydrogen storage. In R. B. Gupta, A. Basile, & T. N. Veziroğlu (Eds.), *Compendium of hydrogen energy* (p. 3-25). Woodhead Publishing. doi: 10.1016/B978-1-78242-362-1.00001-8

Sá, P. (2021). *Projeto detalhado estrutural de veículo aéreo não tripulado com uma célula de combustível de hidrogénio* (Unpublished master's thesis). Academia da Força Aérea, Sintra, Portugal.

The Bureau of Investigative Journalism. (2011). *The bush years: Pakistan strikes 2004 – 2009.* Retrieved November 10, 2021, from `https://www.thebureauinvestigates.com/drone-war/data/the-bush-years-pakistan-strikes-2004-2009`

Tsunoda, S. I., Pace, F., Stence, J., Woodring, M., Hensley, W. H., Doerry, A. W., & Walker, B. C. (2000).

Lynx: a high-resolution synthetic aperture radar. In *2000 ieee aerospace conference. proceedings (cat. no.00th8484)* (p. 51-58 vol.5). doi: 10.1109/AERO.2000.878471

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. doi: 10.1038/s41592-019-0686-2

Weisstein, E. W. (2021, November). *Relative error.* `https://mathworld.wolfram.com/RelativeError.html`.

Werrell, K. P. (1985). *The evolution of the cruise missile* (Tech. Rep.). 10 S Pine St, Maxwell AFB, AL 36112, United States: AIR UNIV MAXWELL AFB AL.

Zaloga, S. J. (2011). *V-1 flying bomb 1942–52: Hitler's infamous "doodlebug"*. Bloomsbury Publishing.

Zitzler, E., Deb, K., & Thiele, L. (1999). Comparison of multiobjective evolutionary algorithms on test functions of different difficulty. In *Proceedings of the 1999 genetic and evolutionary computation conference. workshop program* (pp. 121–122). doi: 10.1162/106365600568202

Züttel, A. (2004, Apr 01). Hydrogen storage methods. *Naturwissenschaften*, *91*(4), 157-172. doi: 10.1007/s00114-004-0516-x