UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Mission-Based Multidisciplinary Design Optimization Methodologies for Unmanned Aerial Vehicles with Morphing Technologies

Pedro Filipe Godinho Lopes Fernandes de Albuquerque

Tese para obtenção do Grau de Doutor em
Engenharia Aeronáutica
(3º ciclo de estudos)

Orientador: Prof. Doutor Pedro Vieira Gamboa
Co-orientador: Prof. Doutor Miguel Ângelo Silvestre

Covilhã, Setembro de 2017

À minha querida Mãe.

Abstract

One of the most challenging aspects of aircraft design is to synthesize the mutual interactions among disciplines in order to achieve enhanced design solutions from the earliest stages of the design process. The complexity of the aircraft physics and the multiple couplings between disciplines complicates this task. The advance of design tools and optimization methods alongside with the computer's exponential increase in data handling capacity is paving the way for the development of comprehensive multidisciplinary design codes that gradually contribute to a paradigm change, leading to a revolution in the design methodologies.

The research work presented in this thesis features two unmanned aerial vehicles preliminary design optimization methodologies - a Parametric Design Analysis and a Multilevel Design Optimization. A specific code has been developed for each methodology, with low-fidelity models being used for the main design disciplines, namely the aerodynamics, propulsion, weight, static stability and dynamic stability. To increase the usability of the codes a graphical user interface for both programs has also been developed.

The first methodology is called Parametric AiRcRaft design OpTimization (PARROT) and relies on a parametric study that optimizes the wing layout for one of two different goals: surveillance mission or maximum payload. Whereas in the former the goal is to maximize the flight range or endurance, the latter's objective is to maximize the useful payload lifted. Constraints include the take-off distance, climb rate, bank angle, cruise velocity, among others. The results have shown to be in line with some experimental benchmarking data and to allow the user to easily evaluate the impact of varying two key design variables (wing mean chord and wingspan) on multiple performance metrics, thus significantly contributing to help the designer's decision-making process.

The second methodology is called MulTidisciplinary design OPtimization (MTOP) and adopts the Enhanced Collaborative Optimization (ECO) architecture, together with a gradient-based optimization algorithm. As the goal is to minimize the energy consumption for the specified mission profile, it results in an unconstrained system problem which aims to assure compatibility between subspaces and dully constrained subspace level problems, which aims to minimize the energy consumption. Instead of each subspace representing the traditional design disciplines (e.g. aerodynamics, structures, stability, etc), the author has chosen to make a different subspace out of each flight stage (e.g. take-off, climb, cruise, etc). The main reason for this choice was the inclusion of morphing technologies as part of the optimization process, namely a variable span wing (VSW), a variable camber flap (VCF) and a variable propeller pitch (VPP). The software final output is the combination of design variables that better suits the objective function subjected to the design constraints. The results have shown how the selection of the optimum combination of morphing/adaptive technologies highly depends on the mission profile. Moreover, the morphing mechanisms weight has a strong impact on the overall performance, which is not easily grasped without an optimization methodology like the one presented.

Globally, these two methodologies foster a more efficient and effective preliminary design stage by feeding the designer's decision-making process with a large set of relevant data.

Keywords

Multidisciplinary, Multilevel, Parametric, Design, Optimization, Morphing, Unmanned Aerial Vehicle, Mission-based

Resumo

Um dos aspetos mais desafiantes do projeto de aeronaves é a gestão das múltiplas interações entre disciplinas, com vista à obtenção de soluções de projeto otimizadas desde os primeiros estágios do projeto de aeronaves. A complexidade da física aeronáutica e os múltiplos acoplamentos entre disciplinas complicam esta tarefa. Com o desenvolvimento de ferramentas de projeto e metodologias de otimização aliadas ao aumento exponencial da capacidade de processamento dos computadores e o desenvolvimento de abrangentes códigos de otimização multidisciplinar estão a contribuir para uma mudança de paradigma, que se espera vir a revolucionar os atuais processos de projeto aeronáutico.

Esta investigação inclui duas metodologias de otimização de projeto preliminar de veículos aéreos não-tripulados - uma otimização paramétrica e uma otimização multinível. Foi desenvolvido um código para cada metodologia, tendo sido utilizados modelos de baixa-fidelidade para as várias disciplinas de projeto, nomeadamente aerodinâmica, propulsão, peso, estabilidade estática e dinâmica. Para aumentar o leque de utilizadores, foi desenvolvido um interface gráfico para ambos os programas.

A primeira metodologia denomina-se *Parametric AiRcRaft design OpTimization* (PARROT) e segue uma abordagem paramétrica que otimiza a geometria da asa para um de dois objetivos: missão de vigilância ou máximo peso. Enquanto na primeira o objetivo passa por otimizar o alcance ou autonomia, na segunda o foco passa por maximizar o peso útil sustentado. Constrangimentos incluem a distância de descolagem, a velocidade de subida, o ângulo de pranchamento, a velocidade cruzeiro, entre outros. Os resultados mostraram estar em linha com resultados experimentais de referência e ainda permitir ao utilizador avaliar o impacto da variação de duas variáveis-chave (corda média aerodinâmica e envergadura) em diversas métricas de desempenho, desta forma contribuindo significativamente para auxiliar o processo decisório do engenheiro de projeto.

A segunda metodologia chama-se *MulTidisciplinary design OPtimization* (MTOP) e adota a arquitetura *Enhanced Collaborative Optimization* (ECO), juntamente com um algoritmo de otimização do tipo gradiente. Uma vez que o objetivo passa por minimizar a energia consumida para um perfil de missão específico, cinge-se a um problema de otimização não constrangido ao nível do sistema, a solução do qual visa a compatibilidade entre subespaços, e um problema devidamente constrangido com o objetivo de minimizar a energia consumida ao nível dos subespaços. Ao invés de cada subespaço representar as disciplinas tradicionais de projeto (e.g. aerodinâmica, estruturas, estabilidade, etc), o autor decidiu criar um subespaço diferente para cada estágio da missão (e.g. descolagem, subida, cruzeiro, etc). A principal razão para esta escolha foi a inclusão de metodologias adaptativas como parte do processo de otimização, nomeadamente uma asa de envergadura variável (VSW), um perfil alar com curvatura variável através de um flap (VCF) e um hélice de passo variável (VPP). O resultado final é a combinação de variáveis que melhor se adequa à função objetivo, sujeitos aos constrangimentos de projeto. Os resultados mostraram que a seleção da combinação de tecnologias adaptativas adequada está altamente dependente do tipo de missão. Além disso, o peso das tecnologias adaptativas tem um elevado impacto que não é facilmente percecionado sem uma metodologia de otimização como a que é apresentada.

Globalmente, estas duas metodologias contribuem para um projeto preliminar mais eficaz e eficiente, alimentando a tomada de decisão do projetista com muita informação relevante.

Palavras-chave

Multidisciplinar, Multinível, Paramétrico, Projeto, Optimização, Tecnologias adaptativas, Veículo Aéreo Não-tripulado, Missão

Contents

List of Figures

List of Tables

Acknowledgements

The current work has been partially funded by the European Community's Seventh Framework Programme (FP7) under the CHANGE project (Combined morphing assessment software using flight envelope data and mission based morphing prototype wing development).

The results of my thesis are the corollary of several years of work at the Department of Aerospace Sciences of University of Beira Interior, lead by Professor Pedro Gamboa and Professor Miguel Silvestre. The completion of this thesis was only possible with the support and guidance of many people to whom I wish to leave a written testimony of my gratitude.

First and foremost I would like to thank my advisor, Professor Pedro Vieira Gamboa on behalf of his priceless commitment, assertive guidance, permanent availability and insightful advice throughout the whole of my thesis. There are no words to express how privileged I feel about working and learning with him.

I would also like to have a word of gratitude to my co-advisor, Professor Miguel Ângelo Silvestre and to Professor Kouamana Bousson for their worthy help and advice.

A special thank goes to my colleague, student and friend Filipe Fraqueiro on behalf of our fruitful collaboration on the elaboration of a graphical user interface for the two computational codes developed in the current thesis.

Many thanks to my colleague and friend Pedro Santos for his crucial support in a wide range of matters as well as for our early morning runs; and to my friend Diogo Vicente for the many useful technical discussions we had.

I thank my sister Ana Albuquerque, my wife Inês Albuquerque, my mother-in-law Mª Manuela Lopes, my mother Mª Amália Pontes Fernandes and my friends Ângela Pisco, Bruno Tojo, Hugo Lopes and Stefano Carli for their support in the revision of the current text.

I would also like to express my gratitude to my current employer, Embraer Portugal, specially to my supervisor, Engineer Sérgio Carvalho, for his flexibility, understanding and encouragement in the final stage of my PhD.

For UBI academic community and the population of Covilhã in general for the 31 months I lived there, which I will always miss and remember with joy.

Last but not least, I would like to thank my mother for her strong personal investment and involvement in my education. I cannot thank her enough.

A final word of gratitude to all the Portuguese taxpayers for their co-funding of my academic path. I will do my best to live up to my compatriots' expectations, thus honoring and hopefully returning their investment.

Glossary

Acronyms	
AAO	All-At-Once
AC	Aerodynamic Center
ACC	Air Cargo Challenge
AoA	Angle-of-Attack
ANAC	*Autoridade Nacional de Aviação Civil* - Portugal
ASO	Asymmetric Subspace Optimization
ATC	Analytical Targeting Cascading
BLISS	Bi-Level Integrated System Synthesis
BLISS-2000	Bi-Level Integrated System Synthesis-2000
CB	Climb
CG	Center of Gravity
CHANGE	Combined morpHing Assessment software usiNG flight Envelope data and mission based morphing prototype wing development
CO	Collaborative Optimization
CSSO	Concurrent Subspace Optimization
CZ	Cruise
DT	Descent
DTOW	Design Take-Off Weight
EASA	European Aviation Safety Agency
ECO	Enhanced Collaborative Optimization
EPD	Exact Penalty Decomposition
ESC	Electronic Speed Control
FAA	Federal Aviation Administration
FP7	European Union’s Seventh Framework Programme
FW	Fixed Wing
GV	Global/shared Variable
GUI	Graphical User Interface
HTail	Horizontal tail

IDF	Individual Design Feasible
IPD	Inexact Penalty Decomposition
ISA	International Standard Atmosphere
LLT	Lifting-line theory
LO	Lift-Off
LT	Loiter
LV	Local Variable
MAV	Micro air vehicle
MDF	Multidisciplinary Feasible
MDO	Multidisciplinary Design Optimization
MDOIS	MDO of Independent Subspaces
MLG	Main Landing Gear
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
MTOP	MulTilevel design OPtimization
NLG	Nose Landing Gear
NP	Neutral Point
PARROT	Parametric AiRcRaft design OpTimization
PZT	Piezoelectric
QSD	QuasiSeparable Decomposition
RMS	Root mean square
RoC	Rate-of-Climb
RoD	Rate-of-Descent
s.t.	subject to
SAND	Simultaneous Analysis aNd Design
SFC	Specific Fuel Consumption
SMA	Shape-memory alloy
UBI	Universidade da Beira Interior
USAF	United States Air Force
VCF	Variable Camber Flap
VLM	Vortex Lattice Method
VPP	Variable Propeller Pitch

VSW	Variable-Span Wing
VTail	Vertical tail
TO	Take-Off

Nomenclature

Latin Symbols	
a	linear acceleration
$\bar{A}, \bar{B}, \bar{D}, \bar{E}, \bar{F}, \bar{G}, \bar{K}$	generic coefficients
b	wingspan
c	wing chord
$\bar{c}$	wing mean aerodynamic chord
c_f	flap chord
C	capacitance
C_d	airfoil drag coefficient
C_D	lifting surface drag coefficient
C_f	friction coefficient
C_l	airfoil lift coefficient
C_L	lifting surface lift coefficient
C_{L_0}	lifting surface lift coefficient at null angle-of-attack
C_{L_α}	lifting surface lift curve slope
C_m	airfoil pitching moment coefficient
C_M	lifting surface pitching moment coefficient
C_{m_0}	airfoil pitching moment coefficient at zero lift
C_{M_0}	lifting surface pitching moment coefficient at zero lift
C_{l_β}	rolling moment coefficient derivative with respect to the side-slip angle
C_{n_β}	yawing moment coefficient derivative with respect to the side-slip angle
C_p	power coefficient
C_{p_0}	power coefficient when the advance ratio is null
d	propeller diameter
d_{fus}	fuselage diameter
D	aerodynamic drag
$D_{friction}$	friction drag
e	Span efficiency factor
$\tilde{e}$	relative error

e_0	Oswald efficiency factor
e_{spec}	specific energy
E	energy
f	generic function
F	force
F	fuselage form factor
F_c	centripetal force
$F_{inertial}$	inertial force
g	acceleration of gravity
h	altitude
$\bar{h}$	aircraft CG position as a fraction of the wing mean aerodynamic chord
h_{limit}	CG forward-most position
h_n	aircraft NP position as a fraction of the wing mean aerodynamic chord
$h_{n_{wb}}$	wing plus body NP position as a fraction of the wing mean aerodynamic chord
I	electric current
I_x, I_y, I_z	moments of inertia about the (x,y,z) axes
J	propeller advance ratio
J_i	subspace level objective function
J_{sys}	system level objective function
i_{ht}	horizontal tail incidence angle
i_w	wing incidence angle
k_C	compatibility factor
k_{st}	stall velocity coefficient (quotient between minimum operating velocity and the stall velocity)
k_v	motor's speed constant
K_n	static margin
l_{fus}	fuselage length
l_{ht}	distance between the wing AC and the horizontal tail AC
$\bar{l}_{ht}$	distance between the aircraft CG and the horizontal tail AC
L	aerodynamic lift
L, M, N	components of the resultant external force moment acting at the airplane CG

L_m	MLG length
L_n	NLG length
m	mass
M	aerodynamic pitching moment
M	Mach number
n	load factor
$n_{cseries}$	number of battery cells in series
n_{legs}	number of mission legs flown
N	rotational velocity
N_l	landing gear ultimate load factor
p	propeller pitch
p, q, r	scalar components of the rotational velocity
$\bar{p}$	wingspan extension factor
P	power
q	dynamic pressure
Q	fuselage interference factor
R^{ele}	electrical resistance
R	radius
Re	Reynolds number
S	lifting surface area
u, v, w	scalar components of the velocity vector
v_{ht}	horizontal tail volume coefficient
$\bar{v}_{ht}$	horizontal tail volume coefficient (alternative definition)
v_{vt}	vertical tail volume coefficient
V_{st}	stall velocity
V	velocity
V_{fuel}	fuel volume
V_w	wind velocity
s	slack variables
S	lifting surface planform area
t	time

t/c	airfoil relative thickness
T	thrust
U	voltage
w	vertical velocity, rate-of-climb or rate-of-descent
W	weight
W_l	landing gear design gross weight
x, y	generic variables
x_L	local variables' vector
x_S	shared variables' vector
x^*	subspaces best attempt to match the system level targets
X, Y, Z	components of the resultant aerodynamic forces acting on the airplane
z	system level targets

Greek Symbols

α	angle-of-attack
α_i	induced angle-of-attack
β	side-slip angle
γ_{cb}	climb angle
γ_{dt}	descent angle
Γ	wing dihedral angle
δ	generic variable increment
δ_{th}	engine/motor power setting
δ_e	elevator deflection angle
δ_f	flap deflection angle
Δt	time interval
Δx	distance interval
ε	wing down-wash angle
η_{prop}	propeller efficiency
$\eta_{prop_{max}}$	maximum propeller efficiency
θ	aircraft pitch angle
λ	taper ratio
λ_C	compatibility parameter

λ^*_C	dynamic compatibility parameter
λ_F	feasibility parameter
λ_{weight}	weighting coefficient
Λ	wing aspect ratio
$\bar{\Lambda}$	wing sweep angle
μ	dynamic friction coefficient
ν	kinematic viscosity of air
ξ	non-dimensional damping coefficient
ρ	density
τ	propulsive force angle with respect to the fuselage axis
ϕ	aircraft bank angle
ψ	aircraft yaw angle

Subscripts and Superscripts

0	reference flight condition
ac	aerodynamic center
$aero$	aerodynamic
av	available
bat	battery
bat_0	battery with motor in idle
C	center
cb	climb
$comb$	combustion
$comp$	compatibility
cz	cruise
dt	descent
E	Earth
eff	effective
ele	electrical
ene	energy
eng	engine
esc	electronic speed control

i	initial
$iter$	iteration
f	flap, final
$fric$	friction
fus	fuselage
ht	horizontal tail
i, j, k, l, n	generic index
$limit$	limit
lo	lift-off
L	lower
m	generic index
max	maximum
min	minimum
$misc$	miscellaneous
mlg	main landing gear
mot	motor
n	generic index
nlg	nose landing gear
$norm$	normalized
p, q, r	scalar components of the rotational velocity
p	propulsive
pay	payload
$prop$	propeller
ref	reference aircraft
req	required
st	stall
str	structures
$shaft$	shaft
$step$	step
sys	systems
to	take-off

thr	threshold
vt	vertical tail
T	trailing edge
TE	trailing edge
u, v, w	scalar components of the velocity vector
U	upper
w	wing
wb	wing and body

Chapter 1

Introduction

1.1 Background and Motivation

Aircraft design involves a comprehensive analysis of a wide range of mutually interacting phenomena, requiring a sound knowledge of disciplines like materials, aerodynamics, structures, control, stability and performance, thus being an inherently multidisciplinary task. Indeed, aircraft design is commonly regarded as a separate design discipline [1], which is different from the former in the way that the designers need to be well versed in all of them while aiming to create the aircraft's geometrical description. Nonetheless, aircraft design is probably amongst the most subjective disciplines of aeronautics which despite having its roots anchored on the laws of physics, neither it is a linear process, nor there are standardized procedures for it.

Aircraft design is an intrinsically iterative process, since there are much more unknowns than equations. Only after years of experience do designers acquire the proficiency level that not only enables them to get closer to the global optimum but also significantly reduce the optimization time, making the overall design process concomitantly more effective and efficient. Hence, Multidisciplinary Design Optimization (MDO) is undoubtedly of utmost relevance and a topic of intense research. In recent decades, the development of MDO architectures has shown that the disciplinary optimization breakthroughs witnessed over the last century will not be the only ones to revolutionize the design process. As an increasing number of experts and researchers devote their time to MDO, it becomes clearer that the discipline of design has got a tremendous progress margin. The possibilities these new methodologies unfold show that they will definitely pave the way of engineering design in a range of subjects that goes far beyond the aerospace industry. The significant improvements on MDO assessed and formalized by Martins et al (2013) [2], combined with the computer's exponential increase of data handling capability show that their scope of applicability is widening. As a corollary, there has been an increasing interest from the industry, with a growing number of companies adopting MDO in order to further enhance both their products' design and design processes as discussed by Weck et al (2011) [3].

Another research topic in vogue is morphing. Despite not being a new concept in the sense it exists since the dawn of aviation - starting with the Wright Brothers use of wing warping - the definition of morphing adopted in this study is narrower in the way it will focus on more recent shape changing technologies, as it will be further discussed in section 2.3. Morphing technologies have been adopted with the goal of widening the aircrafts' flight envelopes, contributing to an increased aerodynamic efficiency and optimized performance. These technologies include, but are not limited to, variable wingspan, wing dihedral, wing sweep, wing torsion, airfoil camber and variable propeller pitch.

The synergy created by the combined use of increasingly efficient solutions with the search for more versatile and capable vehicles, does clearly foster combined research works on Multidisciplinary Design Optimization and Morphing Technologies.

While the use of unmanned aerial vehicles (UAVs) to test these ideas significantly reduces the cost and risks involved, the number of applications of this kind of vehicles has been continuously growing over the last decade, with the only obstacle preventing an effective boom being the scarce regulatory framework. Nevertheless this last impediment is expected to be overcome before the end of the decade, as both the European Aviation Safety Agency (EASA) and the Federal Aviation Administration (FAA) are expected to issue certification specifications for the use of unmanned aerial systems. Some national certification authorities have already started issuing regulatory frameworks, like the Portuguese *Autoridade Nacional de Aviação Civil* [4].

In addition, it is apparent that the most significant shape-changing methodologies application to UAVs is much more likely than their use for manned aircraft, because of the wider range of multimission capabilities of UAVs with respect to manned aircraft. Yet, the performance assessment of morphing solutions done for UAVs as primary test benches as well as probable primary users can provide the cornerstones for the employment of such technologies in manned aircraft.

1.1.1 Greater Research Project

The current work has been partially funded by the European Union's Seventh Framework Programme (FP7) (ACP2-GA-2012-314139-CHANGE-GA) under the Grant Agreement 314139. The CHANGE project's main objective is "to define a stepping stone to insert morphing technologies into air transport aircraft enabling the aircraft to fly with increased performance during the length of their mission.", thus aiming to include three different morphing concepts in the same wing, something that had not been done before the start of the project, in August, 2012.

The CHANGE project (Combined morphing assessment software using flight envelope data and mission based morphing prototype wing development) is a Level 1 project involving 9 partners: Tekever (Portugal), DLR (Germany), ARA (United Kingdom), University of Beira Interior (Portugal), Cranfield University (United Kingdom), Swansea University (United Kingdom), Invent (Germany), Middle East Technical University (Turkey) and the Technical University of Delft (Netherlands). The responsibilities of each member are summarized in Appendix A.

1.2 Aircraft Design

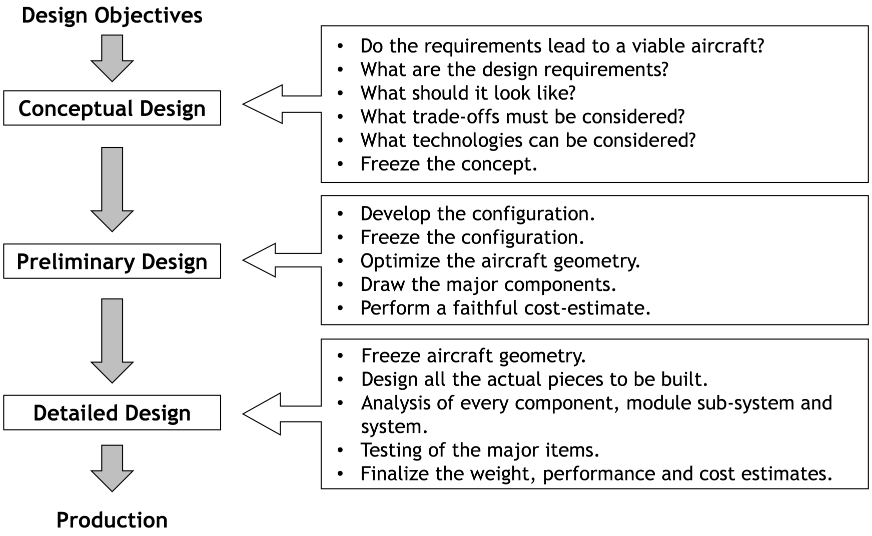
The aircraft designer needs to have a comprehensive knowledge on the mainstream disciplines of aircraft design, namely, on materials, aerodynamics, structures, control, stability and performance. However, the most challenging part of designing an aircraft is to synthesize the mutual interactions among these disciplines in order to achieve enhanced design solutions from the earliest stages of the design process.

The holy grail of aircraft designers was, and in many cases still is, that their concepts can get through subsequent evaluations by different disciplinary experts without major changes being required. If the designer is talented, there is surely much more content on the drawing than what the eye can perceive. Globally, it can thus be said that the earliest stages of the aircraft design process are about a powerful and duly weighted mix of intuition and knowledge.

However, the increasing number of disciplines, the ever greater complexity of the physical models that depict current technologies and the multiple couplings between disciplines makes this task to some extent herculean.

For the successfulness of this demanding undertaking, an increasing number of design companies are adopting the so-called Multidisciplinary Design Optimization (MDO) methodologies (section 2.1). These solutions can not only be far more encompassing than human brains but also provide a quantitative output. Nonetheless, the profitability of using MDO methodologies is uneven throughout the design process.

It can be said that the aircraft design process is divided in three major steps: Conceptual Design, Preliminary Design and Detailed Design, respectively, as depicted in Figure 1.1. While the earlier Conceptual Design stage decision making-process is commonly still based on the designers' themselves, MDO methodologies have proven that they can be particularly worthwhile in saving time and resources while getting closer to the global optimum at the Preliminary Design stage [2].



Design Objectives
Conceptual Design
Do the requirements lead to a viable aircraft?
What are the design requirements?
What should it look like?
What trade-offs must be considered?
What technologies can be considered?
Freeze the concept.
Preliminary Design
Develop the configuration.
Freeze the configuration.
Optimize the aircraft geometry.
Draw the major components.
Perform a faithful cost-estimate.
Detailed Design
Freeze aircraft geometry.
Design all the actual pieces to be built.
Analysis of every component, module sub-system and system.
Testing of the major items.
Finalize the weight, performance and cost estimates.
Production

Figure 1.1: Aircraft design steps.

It is important to stress that this schematic representation of Figure 1.1 fails to provide a feeling of both the relative time span of each major step as well as to reveal that these stages are usually not exactly sequential. The most time-consuming step is obviously the Detailed Design - where literally all parts will be designed, analyzed and tested - whereas the Conceptual Design is the fastest stage - where the analysis is much more generic and largely qualitative. Moreover, these steps are not fully sequential in the way there is an actual time-line overlap between the end of Conceptual Design and the beginning of Preliminary Design as well as the end of Preliminary Design and the beginning of Detailed Design. This happens because aircraft design is not straightforward. As a matter of fact, it is an indelible iterative task, where it is common place to go upstream in the design process as a result of analysis' findings.

1.2.1 Conceptual Design

As depicted in Figure 1.1, the conceptual design phase follows the design requirements definition, thus being the first of three major steps that are usually present in the design of any aircraft. First and foremost, the designer shall determine if the design requirements lead to a feasible vehicle. If that is not the case, one should go back to the design objectives definition and loosen up one or more requirements. Secondly, an usual concern is about the aircraft's affordability. Likewise, if this is not met, the design requirements might be relaxed, depending on the vehicle's purpose.

At this stage the designer's goal is to answer the fundamental questions of design arrangement, like aircraft configuration, size, weight and cost. This is a stage where the experience of the designer is absolutely paramount to getting closer to the optimum and to do it within the least possible amount of time, as there is a great design freedom. It usually starts with a so-called brainstorming where literally all ideas are considered and is followed by some rough analysis and sizing methodologies which will tend to eliminate inefficient or poor-performing solutions with the objective of narrowing down the number of possible configurations. At the end of this stage, the main design concepts shall be frozen.

1.2.2 Preliminary Design

The Preliminary Design stage aims at defining the vehicle's dimensions and geometry. At the beginning of this stage, the aircraft configuration shall be frozen. At this point, the disciplinary experts will design and analyze their respective part of the aircraft, (e.g. landing gear, structure, flight controls). Some actual testing in areas like aerodynamics, propulsion, performance and controls shall also begin.

However, the most important task at this stage is what is usually called ”lofting”. Its goal is to define an appropriate geometrical definition of the outside skin of the aircraft while ensuring a proper fitting between the various components. Despite current industrial practice breaking up the design task between different design groups, with the major aircraft manufacturers commonly subcontracting many smaller companies to develop specific parts of the aircraft, the lofting is usually performed by the aircraft manufacturers themselves. This is concomitantly justified by the relevance of the task to the success of the design and due to confidentiality requirements.

At the end of this stage, the designers should be absolutely confident that the aircraft can perform as required and provide reasonably accurate costing estimates and delivery date. This is usually known as ”you-bet-your-company” as meaningful flaws will impact the designer's company image and will most likely incur in financial losses, and ultimately can lead to bankruptcy.

1.2.3 Detailed Design

The detailed design stage name is largely self-explanatory about its goal of developing the actual design of all the smaller parts and components with the greatest detail. If the wing layout and perhaps the wing box have been designed at the preliminary design stage, now the goal is to design its ribs, spars, skin, bolts and rivets. At this point, the structure sizing is made using high fidelity computational methods to make sure the vehicle will perform according to its operational requirements complying with the applicable certification specifications established by the civil aviation authorities or military specifications, depending on the vehicle's target market segment. The ultimate structural studies, also include ground and flight experiments to physically accept the compliance with the regulations.

At this design stage, several minor modifications to the original design shall be at stake. Some of them will be accepted in order to ease production or lower manufacturing times and costs, whereas some others will not because they would have an unacceptable impact on performance or weight. This is probably one of the most thrilling undertakings at this stage - the compromise between various disciplinary experts. Globally, it must be assured that none of these changes impact the original design requirements.

The manufacturing of the whole aircraft will follow this last design stage. In most cases however, some components and parts shall still undergo detailed design concurrently with early manufacturing of other components and parts.

1.2.4 Design Optimization Programs

The most challenging part of a high performance design is to synthesize the mutual interactions among disciplines in order to achieve enhanced design solutions at the earliest stages of the design process. However, the large number of disciplines, the complexity of the aircraft physics and the multiple couplings between disciplines complicates this task.

Nevertheless, the development of comprehensive multidisciplinary design codes is gradually contributing to a paradigm change, in the way it is expected to revolutionize the design process. While the earlier conceptual design phase decision making-process is commonly still based on the designers themselves, multidisciplinary design optimization methodologies have proven that they can be particularly worthwhile in saving time and resources while getting closer to the global optimum at a preliminary design stage, as shown by Martins et al (2013) [2].

Amongst the different multidisciplinary design programs which include a graphical user interface, it is worthwhile to mention some cornerstone developments in the context of aircraft disciplinary analysis and design optimization.

One of the earliest such works was the Advanced Aircraft Analysis (AAA) [5], a tool which enables aircraft design and optimization as it allows a wide spectrum of analysis, although this is a complex software and a license is required to use it. AAA is divided into ten independent modules such as weight, aerodynamics, performance, stability and controls, among others. Due to its multidisciplinarity, this software allows a comprehensive aircraft design analysis and optimization even though the latter is generally user guided through an informal process.

CEASIOM (Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods) [6] is a freeware software featuring a geometry module which makes it possible to have a general view about the aircraft geometry under analysis. It also includes modules related to stability, controls and aerodynamics.

It is also worthwhile to refer XFRL5 [7], which was developed by André Deperrois. It uses XFOIL for the 2D airfoil analysis and VLM or LLM for 3D wing analysis. XFOIL [8] was developed at the Massachusetts Institute of Technology, being a widely known code to calculate airfoils' aerodynamic coefficients. Despite being an accessible and widely used tool, XFLR5 does not enable automatic optimization of airfoils, lifting surfaces and/or fuselages. Conversely, the analyses data can be used by the designer for optimization purposes, but the overall optimization workflow will be far more toilsome.

A more recent example of design optimization programs is SUAVE [9] – developed at Stanford University in collaboration with Embraer – which is a comprehensive tool with four calculation methods: Traditional Aircraft Design, Advanced Configuration/Unconventional Technology Design, Optimization, Aircraft/Discipline Analysis. SUAVE is an open source and has been written with the Python language. It can be incorporated using extensible interfaces and prototyped with a top-level script that allows the creation of arbitrary mission profiles, unconventional propulsion networks, and right-fidelity at right-time discipline analyses.

Other tools have been developed, which enable the assessment of morphing technologies. Suleman et al (2014) [10] have detailed a MDO framework for conceptual design and analysis of new aircraft configurations, including the capability to analyze and quantify the effect of morphing wing solutions on aircraft performance. Two main modifications to the MDF architecture were implemented. The first change consisted on replacing the real models (aircraft disciplines) by surrogate models; the second modification is rather more particular and was specially introduced in the MDO Framework to better suit the capabilities of the aircraft to morph during flight by splitting up the general aircraft configuration optimization and control optimization of each performance goal in two levels of MDO problems. The first/lower level is responsible for determining the best morphing strategy for the selected performance metric. So for each chosen metric a MDO problem is defined. The selected performance parameter is set as the objective and the variables able to change during flight can be employed as design variables (for example: aerodynamic controls, morphing devices and engine controls). In the upper/second level, the geometric configuration variables are optimized. The performance parameters optimized in the lower level can be included in a weighted multi-objective function and/or in the MDO problem set of constraints. It has been shown how the devised MDO framework can contribute to an enhanced design of a conventional wingtip and proved that a morphing wingtip would probably present no advantage due to its low aerodynamic gains compared to its weight penalty.

Lyu and Martins (2015) [11] have shown that adaptive morphing trailing edge wings have the potential to reduce the fuel burn of transport aircraft. The effectiveness of the trailing edge morphing was demonstrated by comparing with the optimized results of a hypothetical fully morphing wing. From an aerodynamic perspective, an adaptive morphing trailing edge can easily offer additional drag reduction without a complete redesign of the wing (1% drag reduction at on-design conditions, and 5% drag reduction near off-design conditions). However, to provide a comprehensive evaluation benefit, a multidisciplinary study is required to examine the trade-offs between aerodynamics, structures, and controls.

Adaptive morphing trailing edge technology offers the potential to decrease the fuel burn of transonic transport aircraft by allowing wings to dynamically adjust to changing flight conditions. Conversely, current aircraft use flap and ailerons to adjust the wing during flight, thus introducing unnecessary drag. Burdette et al (2015) [12] have compared a standard non-morphing wing against a wing retrofitted with a morphing trailing edge, and a clean sheet wing designed with the morphing trailing edge. The morphing trailing edge was found to achieve fuel burn reductions via two mechanisms. The first was its ability to improve maneuver load alleviation, allowing for lighter wing structures. The second was a reduction of the coupling between the cruise and maneuver cases, which allows the cruise configuration to improve without causing adverse effects on maneuver performance. The wing retrofitted with the morphing trailing edge improved the fuel burn as effectively as the full wing redesign without morphing. Additional fuel burn reductions were observed for the clean sheet design. The morphing trailing edge decreased the fuel burn by performing load alleviation at the maneuver condition, weakening the trade-off between cruise performance and maneuver structural constraints, resulting in lighter wing-boxes and more aerodynamically efficient cruise configurations.

Vale et al (2016) [13] have also used multidisciplinary design optimization in a comparative study between the lift over drag ratio (L/D) performance of a fixed wing glider and a camber morphing wing glider, featuring aerodynamics, structures and static stability analyses. The proposed camber morphing concept featured a coupled effect of changing the wing twist. For that reason, and for the sake of computing efficiency, the CMWG MDO procedure is a sequential optimization, with the goal of converging the wing twist on the surrogate and FEM, while the FWG MDO procedure is a single level optimization. The results indicate that the CMWG can effectively reduce the magnitude of the load factor experienced by the aircraft.

1.2.4.1 Work Scope

The current work aims at providing a comprehensive multidisciplinary framework for mission-based optimization of morphing aircraft. In particular, and contrarily to the aforementioned methodologies, one aim to tailor the optimization methodology for the assessment of morphing technologies. In addition to the development of an arbitrary mission-profile based on generic flight conditions and adaptive technologies, the iterative propulsion model matches the blade versus motor/engine performance for any combination of velocity, altitude, throttle setting and required power.

1.3 Objectives

This PhD thesis' main goal is the development of mission-based state-of-the-art aircraft design optimization methodologies that can contribute to an efficient (not costly) and effective (accurate) preliminary design stage. Two computational codes have been created using the mainstream disciplines of aeronautical design: aerodynamics, propulsion, mass distribution, static and dynamic stability and performance. In addition, these tools enable the assessment of adaptive technologies. In order to make the codes easily usable by any interested aircraft designer, the development of a user friendly graphical user interface is also part of the current research project goals.

1.4 Contributions

The core contributions of this thesis are two preliminary UAVs design optimization methodologies - a **PAR**amet**R**ic aircraft design **O**p**T**imization (PARROT) and a **M**ul**T**ilevel aircraft design **OP**timization (MTOP), which distinguishing feature is the fact of being mission-based optimization methodologies, which means that the aircraft is being optimized for a specific mission profile, rather than being optimized for cruise conditions. The latter enables the assessment of morphing technologies. These two methodologies have been materialized in two different codes largely sharing the physical analysis' models.

The former methodology enables the user to optimize the wing layout (wing mean chord and wingspan) for one of two different goals: maximum endurance/range mission or maximum payload, whereas the latter relies on a multidisciplinary and multilevel design optimization methodology combined with the use of morphing solutions making it possible to optimize the UAVs for a more detailed pre-defined mission profile.

1.5 Thesis Structure

The current thesis is divided into four main chapters: State-of-the-art Review (2), Analysis Models (3), Parametric Design Optimization (4) and Multilevel Design Optimization (5).

Chapter 2 introduces the basic definitions and provides an overview on the latest developments in MDO, particularly on distributed optimization architectures, highlighting present and future challenges. An insight on morphing technologies is also included.

Chapter 3 describes the analysis models used in the development of the two codes. It includes a comprehensive overview of the disciplinary models (e.g. aerodynamics, weight, propulsion and stability), a description of the possible mission stages (e.g. take-off, climb, cruise) flight mechanics and also specific modifications tailored for morphing technologies.

Chapter 4 presents a mission-based parametric study-based aircraft design optimization methodology and explains how the PARROT code has been developed highlighting the merits of such approach with several case studies. This chapter also includes a description of the PARROT's graphical user interface.

Chapter 5 features a mission-based distributed aircraft design optimization methodology and thoroughly describes the MTOP code, presenting the merits of this broader approach along the discussion of several case studies, which, among other advantages, can assess the profitability and optimize the use of morphing technologies. In addition, this chapter also features an overview of MTOP's graphical user interface.

Notwithstanding that each chapter includes short introduction and concluding sections, the conclusions from the design optimization methodologies and computational implementations are summarized in Chapter 5. Future work and recommendations are also presented in this last chapter.

Chapter 2

State-of-the-art Review

2.1 Multidisciplinary Design Optimization

As shown by Gohman et al (2012) [14] and already discussed, the design of future aerial vehicles requires the consideration of multiple and interacting disciplines as well as different design goals, like lowering manufacturing and operational costs, minimizing weight without compromising structural behavior or lowering emissions without jeopardizing performance. As shown by Martins and Lambe (2013) [2] and La Rocca et al (2009) [15], Multidisciplinary Design Optimization (MDO) is definitely of utmost relevance in the quest for these goals.

The works of Schmit et al (1960, 1981 and 1984) [16–18] and Haftka et al (1973, 1977 and 1979) [19–21] are the genesis of MDO when they widened the scope of their structural design optimization to other disciplines, the first applications being wing design, where there is an inherent coupling between aerodynamics, structures and controls.

In recent years, however, MDO algorithms have been extended to the full aircraft, Kroo et al (1994) [22] Antoine et al (2005) [23] and Henderson et el (2012) [24]. In this context, the works of Wakayama et al (2000) [25] that developed a comprehensive MDO methodology for aircraft preliminary design, of Gur et al (2010) [26] that developed a complete design optimization study for a wing and the work of Gohmal et al (2012) [14] that developed a conceptual multifidelity, multistrategy and multidisciplinary design optimization for a generic jet wing are noteworthy.

There have been several surveys of MDO over the last 25 years. Haftka et al (1992) [27] were among the first to review the MDO architectures. Cramer et al (1994) [28] formalized the monolithic architectures and detailed the required optimization algorithms (gradient-based methods). Balling and Sobieski (1996) [29] identified a number of possible monolithic approaches and estimated their computational cost.

In a collection of articles edited by Alexandrov and Hussaini (1997) [30], Kroo et al (1997) [31] provided a comprehensive overview on MDO, including a description of both monolithic and distributed architectures. Later, in that same year, Sobieski and Haftka (1997) [32] published an exhaustive survey of the MDO literature known at that time.

More recently, Martins and Lambe (2013) [2] have presented a survey of all the optimization architectures presented in the literature by the time of its publication. In this review, the architectures have been compared using a unified nomenclature, which is adopted throughout this thesis, since this work is concomitantly the latest and most comprehensive effort to benchmarking the existing methodologies.

The forthcoming paragraphs include some introductory remarks to numerical optimization concepts (subsection 2.1.1) and are followed by a thorough yet abridged overview on the two fundamental MDO architecture categories - Monolithic (subsection 2.1.2) and Distributed (subsection 2.1.3).

2.1.1 Introduction to Numerical Optimization Concepts

Before providing a quick but comprehensive overview of the multidisciplinary design optimization architectures developed so far, it is important to provide some relevant definitions with which the reader may not be fully acquainted:

- **compatibility:** is what happens to variables that are shared among different subspaces at the end of multilevel optimization schemes;
- **condition number <> gradient:** of a function with respect to an argument measures how much the output value of the function can change for a small change in the input argument;
- **consistency constraints:** in system design, consistency enforces that both the target and actual variable values are the same;
- **convex design space:** a design space is said to be convex if the line segment between any two points on the graph of the function lies above the graph;
- **design constraints:** in system design, these constraints are usually externally imposed, either by one organization, some external regulation or some other disciplinary groups working on the same project;
- **feasibility:** is the set of all possible points (sets of values of the choice variables) of an optimization problem that satisfy the problem's constraints, potentially including inequalities, equalities, and integer constraints. It is thus the initial set of candidate solutions to the problem, before the set of candidates is narrowed down;
- **sensitivity analysis:** a technique used to determine how different values of an independent variable will impact a particular dependent variable under a given set of assumptions;
- **objective function:** in linear programming, the optimization goal is to maximize or minimize a given function or a linear combination of mathematical functions which measure physical quantities of interest. The objective function minimization or maximization is thus the ultimate optimization target;
- **optimization:**

  distributed: the original problem is partitioned into a number of subproblems each with its own subset of variables and constraints, (subsection 2.1.3);

  monolithic: optimization where a single optimization problem is being solved, (subsection 2.1.2);

  multilevel: is a special kind of optimization where one or more problem is embedded within another in cascade fashion and is a particular type of distributed optimization where there is a hierarchy among the different problems;
- **variable:**

  coupled: exhibits some kind of dependence on some other variable;

  independent: does not have any kind of dependence;

  local: is not shared among disciplinary analyses;

  shared/global: is shared by different disciplinary analyses;

  slack: is added to inequality constraints to transform them into equalities;

state: one of the set of variables that are used to describe the mathematical ”state” of a dynamical system. The state of a system describes enough about the system to determine its future behavior in the absence of any external forces affecting the system. Models that consist of coupled first-order differential equations are said to be in state-variable form;

- **surrogate model:** is a method used when an outcome of interest cannot be easily directly measured, and a model of the outcome has to be built to be used instead;
- **well-conditioned:** a problem with a low condition number is said to be well-conditioned, which means the output value of a function does not significantly change due to small changes in the input argument.

Despite the fact that different MDO architectures can be used to solve the same problem, the architecture's choice has shown to have great influence on both the solution time and the final design. The same goes for the optimization algorithm employed, e.g. on the one hand the use of a purely gradient-based algorithm may result on the solution getting prematurely stuck in a local maximum or minimum, on the other hand the computational cost of a such choice is likely to be considerably lower than most non-gradient-based algorithms.

As per Martins and Lambe (2013) [2], the optimization architectures can be categorized in the groups shown in Figure 2.1.

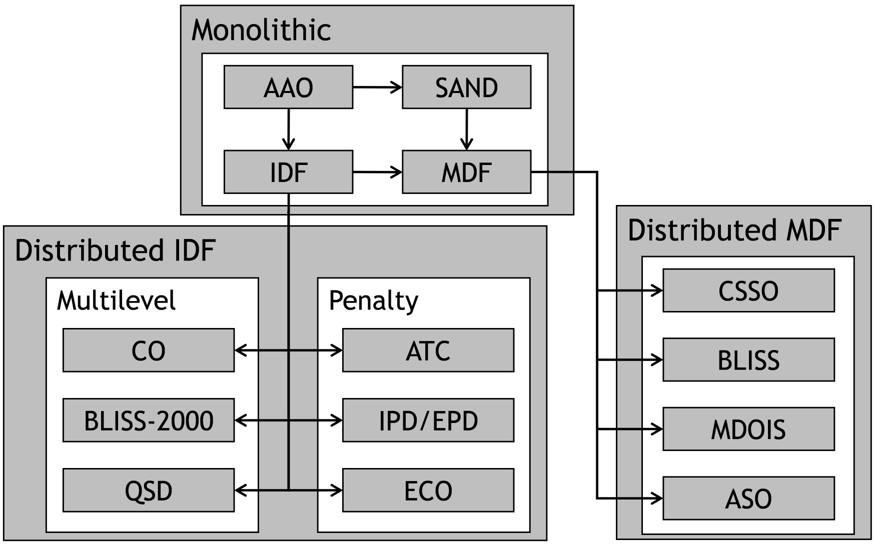


Figure 2.1: MDO architectures.

2.1.2 Monolithic Optimization Architectures

If the boundaries between the different disciplinary analyses are disregarded, an MDO problem is reduced to a standard constrained nonlinear problem which means finding the optimum combination of design variables that minimizes or maximizes the design objective function subject to its constraints. The architectures presented in this subsection are referred to as Monolithic architectures. These solve a single optimization problem, with different strategies being adopted for the sake of multidisciplinary feasibility.

Prior to presenting the specific architectures shown and discussed in the literature, the optimization problem from which all the others are derived - All-at-Once (AAO) problem - will be presented. The definition of AAO problem adopted in the current work, is in line with the ones presented by Cramer et al (1994) [28] and Martins and Lambe (2013) [2], which corresponds to “the most general formulation”. Accordingly, the AAO problem includes all the design, state, input and output variables. However, the optimization problem is never solved in this AAO architecture because the consistency constraints can be easily eliminated.

The remaining three optimization architectures - Simultaneous Analysis and Design (SAND), Individual Design Feasible (IDF) and Multidisciplinary Design Feasible (MDF) - can be found by eliminating equality constraints to this fundamental AAO formulation, as depicted in Figure 2.2. The reader should be aware that despite the formulation differences presented below, the problem being solved is always the same.

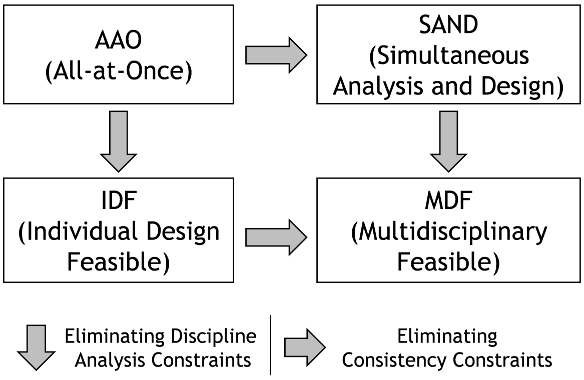


Figure 2.2: Schematic representation of the different monolithic architectures.

2.1.2.1 Simultaneous Analysis and Design (SAND)

A logic step following the AAO formulation is to eliminate the consistency constraints, which can be done by introducing a single group of coupling variables to replace the separate target and response groups, as discussed by Haftka (1985) [33].

In this formulation, there is no need to solve any discipline analysis explicitly or exactly at each iteration, which results that the optimization problem can potentially be solved rather quickly by letting the optimizer explore regions that are infeasible with respect to the design constraints. Additionally, the SAND methodology is not restricted to multidisciplinary systems and can be used in single-discipline optimization as well. Contrarily, this problem formulation still requires all state variables and discipline analyses to be known and therefore problem size and potential premature termination of the optimizer at an infeasible point are issues of concern. The discipline analysis equations are treated explicitly as constraints, which means the residual values - and possibly their derivatives - need to be available to the optimizer. However, it is important to note that even if the software can be modified to return the residuals, the cost and effort required may be excessive, according to Haftka (1985) [33].

2.1.2.2 Individual Design Feasible (IDF)

The Individual Design Feasible (IDF) architecture is reached by eliminating the consistency constraints from the AAO formulation.

IDF enables the discipline analyses to be performed in parallel, since the coupling between disciplines is resolved by the coupling variable copies and consistency constraints. As a consequence, the IDF problem is substantially smaller than the SAND, requiring minimum modification to the existing discipline analysis, which is one of the main assets of this formulation.

Conversely, if the number of coupling variables is too large, the resulting optimization problem might be too heavy to solve efficiently. When the disciplinary analyses are expensive, evaluating the objective and constraint functions' gradients becomes costly. In practice gradients are estimated using some finite-differencing procedure. While this approach preserves discipline feasibility it is costly and unreliable Martins and Lambe (2013)[2].

For a comprehensive overview on the available options for computing derivatives in MDO problems, the reader is encouraged to see the work of Martins and Hwang (2013) [34].

2.1.2.3 Multidisciplinary Feasible (MDF)

The Multidisciplinary Feasible (MDF) problem formulation, presented by Cramer et al (1994) [28] - also known in the literature by fully integrated optimization (Alexandrov and Hussaini (1997) [30]) and by nested analysis and design (Balling and Sobieszczanski (1996) [29]) - is as small as it can be for a monolithic architecture, since only the design variables, objective function and design constraints are under the control of the optimizer. Accordingly, this formulation is obtained after removing the analysis and consistency constraints from the AAO problem (thus combining the SAND and IDF architectures).

MDF returns a system design that always satisfies the consistency constraints, even if the optimization process is terminated early. However, as per Martins and Lambe (2013) [2], the design constraints satisfaction may not be guaranteed.

A consistent set of coupling variables must be computed and returned to the optimizer every time the objective and constraint functions are evaluated. MDF requires a full multidisciplinary analysis to be performed at each iteration. Instead of simply running each individual discipline analysis once per iteration, as in IDF, it requires running every analysis multiple times until a consistent set of coupling variables is found.

For instance, the block Jacobi coupling process is considered to be the most conceptually straightforward iterative approach for strong coupling of two or more codes, Matthies et al (2006) [35]. The Gauss-Seidel iterative approach, however, has shown to converge faster than block Jacobi coupling because it uses information about the dependent parameters as soon as they become available, Ismail Farajpour and Sez Atamturktur (2012) [36]. Newton-like methods offer gradient-based iterative coupling techniques relying on the Newton iterations completed on the Jacobians in the coupled system of equations.

Furthermore, gradient calculations are also harder for MDF than IDF. While in IDF gradient information must be discipline-feasible, in MDF the gradient information must be feasible with respect to all disciplines, because a full multidisciplinary analysis is performed at each iteration. However there are several semi-analytic methods that contribute to a very significant reduction of this step by eliminating finite-differencing over the full multidisciplinary analysis, Sobieszczanski (1990) [37] and Martins (2005) [38].

2.1.3 Distributed Optimization Architectures

Up to now, the focus has been put on the Monolithic architectures - those that solve a single optimization problem. However, a significant number of other architectures decompose the optimization problem into a set of smaller optimization problems, or subproblems, that have the same solution when reassembled - these are called Distributed architectures.

The primary motivation for decomposing the MDO problem comes from the inherent architecture of the engineering design environment. Current industrial practice involves breaking up the design of large systems and distributing them between several different design groups, which are often geographically apart from each other and therefore tend to communicate less frequently than what would be desirable. They are likely to work in different domains of expertize and probably have different internal procedures and optimization methodologies. Decomposition through distributed architectures allows each design group to undertake its own design decisions and to control its own design variables, with the advantage of periodically reviewing that information against data from other design groups, in such a way that the overall design is optimized concurrently, as discussed by Martins and Lambe (2013) [2].

Moreover, in Monolithic Architectures all discipline analysis' programs are run exactly the same number of times, which significantly impacts the optimization performance as inexpensive analyses will experience long periods of inactivity while waiting for updates from most expensive analyses. However, a proper load balance using parallel processing (smart distribution of computation among several processors) mitigates this issue. One of the main assets of distributed architectures is that the design process can not only run in parallel, but also asynchronously [2].

2.1.3.1 Concurrent Subspace Optimization (CSSO)

CSSO is one of the oldest distributed architectures for large-scale MDO problems. The original formulation Sobieszczanski (1988) [39] decomposes the system problem into independent subproblems whose optimizations can be run in parallel, as featured in Figure 2.3.

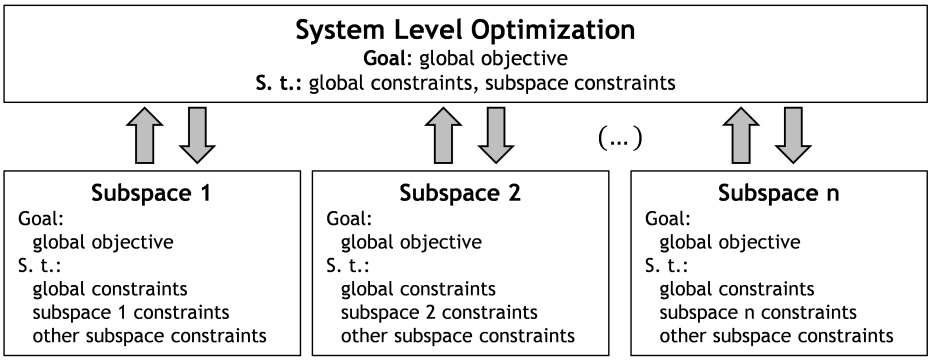


Figure 2.3: Schematic representation of the Concurrent Subspace Optimization architecture.

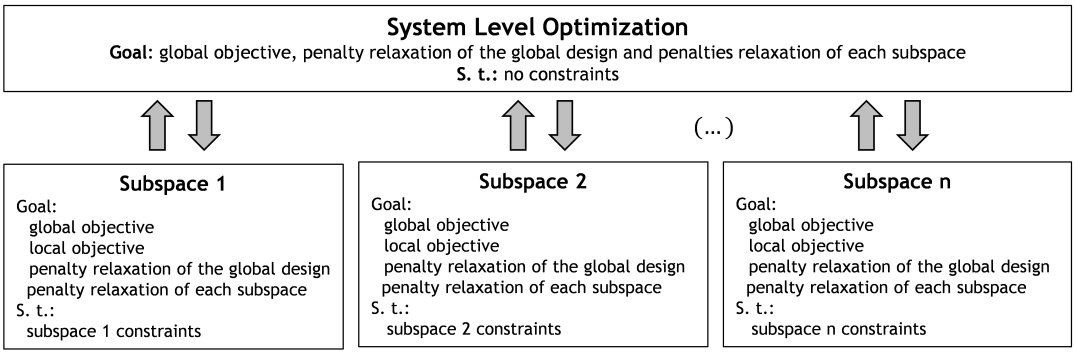
It takes advantage of the fact that the approximations of non-local disciplinary states help to understand the influences of local disciplinary variables on system level constraints and objective functions. The system level optimization - a coordination problem - uses the approximation models to replace the required disciplinary analysis. These disciplinary level models are progressively updated based on the optimized disciplinary states.

Shankar et al (1993) [40] proposed several improvements, including methods for updating the coefficients, and tested them on two- and three-variable quadratic optimization problems. One of the main issues with CSSO is that the architecture performance is sensitive to parameter selection, and extensive tuning may be required to efficiently run this formulation on larger nonlinear problems.

Perez et al (2004) [41], Yi et al (2008) [42], and Tedford and Martins (2010) [43] have compared CSSO to other architectures (IDF, MDF, MDOIS and CO) using gradient-based optimization algorithms. CSSO required many more analysis calls and function evaluations to converge to an optimal design. The results of de Wit and van Keulen (2007) [44] showed that CSSO was unable to reach the optimal solution of even a simple minimum-weight two-bar truss problem. Hence, it is believed that CSSO performs poorly when compared with newer MDO architectures.

2.1.3.2 Analytical Targeting Cascading (ATC)

The Analytical Targeting Cascading (ATC) architecture relies on a system design approach that enables top level design targets to be cascaded down to lower levels of the modeling hierarchy, as shown by Kim (2001) [45] and Kim et al (2003) [46]. In contrast to the other architectures that handle only two (system and subproblems), it can handle multiple levels, thus its name being cascading. It is not different from other MDO architectures with a system objective of minimizing the squared difference between a set of system targets and the model responses, Figure 2.4.



System Level Optimization
Goal: global objective, penalty relaxation of the global design and penalties relaxation of each subspace
S. t.: no constraints

(...)

Subspace 1
Goal:
global objective
local objective
penalty relaxation of the global design
penalty relaxation of each subspace
S. t.:
subspace 1 constraints

Subspace 2
Goal:
global objective
local objective
penalty relaxation of the global design
penalty relaxation of each subspace
S. t.:
subspace 2 constraints

Subspace n
Goal:
global objective
local objective
penalty relaxation of the global design
penalty relaxation of each subspace
S. t.:
subspace n constraints

Figure 2.4: Schematic representation of the Analytical Targeting Cascading architecture.

A particular feature of this architecture is the usage of a penalty relaxation term for the global design constraints and a penalty relaxation term for each discipline consistency constraint. Their function is to ensure compatibility between subspaces. The most common penalty functions in ATC are quadratic functions. An adequate selection of the penalty weights is paramount as it impacts both the final consistency in the discipline models and the algorithm convergence, as discussed by Martins and Lambe (2013) [2].

The ATC architecture is able to solve both monolithic (discipline-based systems) and distributed (multilevel hierarchic systems) problems. In the latter case, penalty functions are applied to all constraints that combine local information with data from other subspace level. One of the assets of this formulation is that the post-optimality derivatives are not required in any of the subproblems, since non-local data are always treated as parameters.

ATC is likely to be the most used architecture in automotive industry problems, but it has also performed well in aircraft design optimization problems, as shown by Allison et al (2006) [47, 48] and by Tosserams et al (2010) [49], among others [2]. Overall, and in spite of the relatively scarce benchmarking backlog, ATC seems to be competitive with all the other benchmarked distributed architectures, including standard versions of CSSO, CO and BLISS, as shown by de Wit and van Keulen (2007) [44]. Finally, it is worthwhile to stress that the ATC with a gradient-based optimizer has shown not to be competitive with other monolithic architectures, requiring more function and discipline evaluations.

2.1.3.3 Collaborative Optimization (CO)

The main idea behind the collaborative optimization (CO) architecture is that disciplinary experts (at subspace level) take part in the design-decision making process without the need for addressing local changes imposed by other groups (or subspaces) of the system, allowing parallel development of the design - an unarguable asset. A coordination problem ensures compatibility between subspaces, while the discipline optimization subproblems are made independent of each other by using copies of the coupling and shared design variables, as shown by Braun (1996) [50] and by Braun et el (1996) [51]. There are two different versions of the CO architecture, CO1 and CO2, as presented by Martins and Lambe (2013) [2]. Being the most widely used, the forthcoming considerations refer to the latter.

With respect to CSSO, the CO architecture provides a higher degree of design freedom between disciplines while reducing the interdisciplinary communication requirements (Figure 2.5).

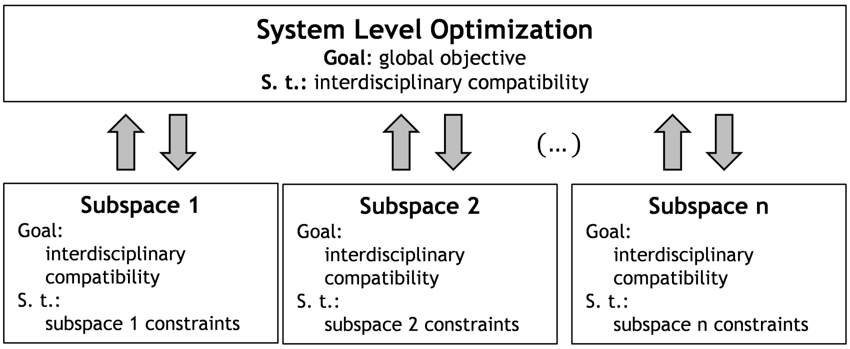

System Level Optimization
Goal: global objective
S. t.: interdisciplinary compatibility
(...)
Subspace 1
Goal:
interdisciplinary compatibility
S. t.:
subspace 1 constraints
Subspace 2
Goal:
interdisciplinary compatibility
S. t.:
subspace 2 constraints
Subspace n
Goal:
interdisciplinary compatibility
S. t.:
subspace n constraints

Figure 2.5: Schematic representation of the Collaborative Optimization architecture.

Hence, the system-level problem is responsible for minimizing the design objective(s) whereas the subspace-level problem is in charge of minimizing inconsistency between subspaces.

In spite of the organizational gains of having fully separable discipline subproblems, the CO architecture has shown to have remarkable weaknesses in its mathematical formulation that has lead to poor performance, including system level infeasibility, low convergence rate and even lack of convergence, as shown by De Miguel and Murray (2000) [52] and by Alexandrov and Lewis (2002) [53]. These problems have motivated different researchers to look for ways of enhancing the CO architecture performance.

Despite the aforementioned numerical issues, CO has been widely implemented on a number of MDO problems, mostly in the design of aerospace systems, but not only. Examples range from the design of launch vehicles (Braun et al (1997) [54]) and rocket engines (Cai et al (2010) [55]) to the preliminary design of complete aircraft (Kroo et al (1994) [56]) and an aircraft family design (Allison et al (2006) [47]).

Globally, CO has shown to be an inefficient and sometimes, even ineffective architecture. Albeit being competitive with other distributed architectures if their system level equality constraints are relaxed, CO is never competitive with monolithic approaches.

2.1.3.4 Enhanced Collaborative Optimization (ECO)

The most recent version of CO, known as enhanced collaborative optimization (ECO) was developed by Roth and Kroo (2008) [57, 58]. While still deriving from the same problem as the CO formulation, the system and discipline optimizations have been reversed in ECO, (Figure 2.6).

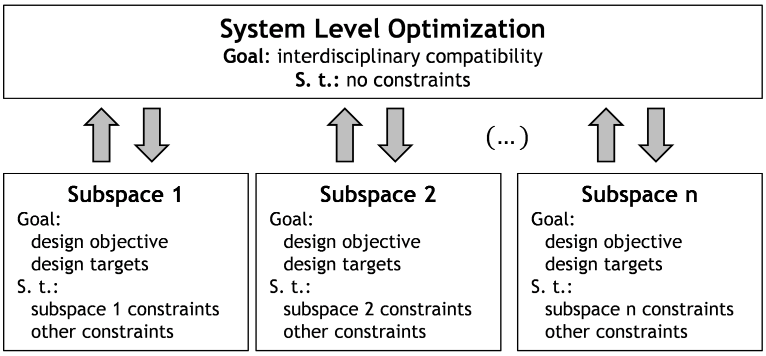


Figure 2.6: Schematic representation of the Enhanced Collaborative Optimization architecture.

In ECO, the system subproblem minimizes system infeasibility while the discipline subproblems minimize the objective function. The system subproblem is an unconstrained problem and its post-optimality derivatives are not required by the system subproblem because the discipline responses are treated as parameters. The system subproblem chooses the shared design variables by averaging all the discipline preferences.

The key idea of this improved version with respect to the original formulation [58] is to infuse the subspace optimization problems with additional information, while maintaining the low dimensionality of the system level problem.

Results from an analytic test case and an aircraft family design problem [57] suggest that this novel approach is not only robust, but also leads to a significant reduction in computational effort. The analytic test case results [57], in particular, suggest that ECO yields significant computational savings, when compared with CO. The trade-off for this computational savings is a small increase in the complexity of the method. In specific, a potentially significant amount of subspace constraint information must be shared among the disciplines - contrarily to what happens in CO - which, in some cases, can negatively impact the solution's performance.

ECO has also been benchmarked against Analytical Targeting Cascading (ATC) [58]. The results have shown that both formulations succeed on efficiently locating the global minima.

Altogether, ECO seems to be effective, whereas the original CO formulation tends to be an inefficient architecture for MDO problems. Without the modifications proposed in ECO, the architecture requires a disproportionately large number of function and discipline evaluations, assuming it converges, as shown by Kodiyalam (1998) [59] and Yi et al (2008) [42].

2.1.3.5 Bi-level Integrated System Synthesis (BLISS)

Like the CSSO, the Bi-level Integrated System Synthesis (BLISS) is a method for optimization of engineering systems by decomposition. It separates the system level optimization, having a relatively small number of design variables, from the potentially numerous subspace optimizations which can have a large number of local design variables, as shown by Sobieszczanski et al (1998) [60]. The subsystem optimizations are autonomous and may be conducted concurrently, which means that different design groups can work concomitantly, thus enabling parallel processing.

The design is improved at each iteration as system and subsystem optimizations alternate. In order to prevent violation of the discipline constraints by changes in the shared design variables, post-optimality derivative information is required to solve the system subproblem.

As the BLISS architecture relies on linearization of a generally nonlinear optimization, its effectiveness depends on the degree of non-linearity [60]. If the problem being solved is highly non-linear, the problem may converge slowly. As any gradient-guided method, it guarantees a cycle-to-cycle improvement, but the global optimum is only found for convex problems. Thus, identifying the global optimum relies on the rightful selection of the starting point.

Sobieszczanski et al (2003) [61] have shown that the BLISS performance is improved in problems with a large number of design variables that are local with respect to the subspaces and a relatively low number of system level design variables. Nevertheless, the available results from Perez et al (2004) [41], de Wit and van Keulen (2007) [44] and Yi et al (2008) [42] show that the BLISS architecture is not competitive with other architectures.

However, there have been two adaptations of the original BLISS architecture that are worthy of notice, since they have contributed to a relevant performance improvement. The first is Ahn and Kwon's proBLISS (2006) [62], an architecture for reliability-based MDO, while the second is LeGresley and Alonso's BLISS/POD (2004) [63], which integrates a reduced-order modeling technique called proper orthogonal decomposition, as shown by Berkooz et al (1993) [64].

2.1.3.6 Bi-level Integrated System Synthesis-2000 (BLISS-2000)

An extremely different formulation called Bi-level Integrated System Synthesis-2000 (BLISS-2000) was developed by Sobieski et al (2003) [61]. Since it does not require a multidisciplinary analysis to restore the feasibility of the design, it is worthwhile to devote it its own subsection.

Like other IDF-derived architectures, BLISS-2000 uses coupling variable copies to enforce consistency at the optimum. Surrogate models of the discipline optima are used for the exchange of information between the system and discipline subproblems. A particular feature of the BLISS-2000 is the use of a vector of weighting coefficients, (w_i), attached to the discipline states. The coefficients should be chosen based on the structure of the global objective to allow the discipline subproblems to find an optimum more quickly.

Despite tending to generate a larger amount of computing and voluminous intermediate data with respect to the original BLISS formulation, the BLISS-2000 architecture simplifies the entire procedure, reducing the human effort and the time required to learn a new method [61]. This appears to be a cost effective tradeoff, considering that computing costs are steadily decreasing, while labor costs tend to grow.

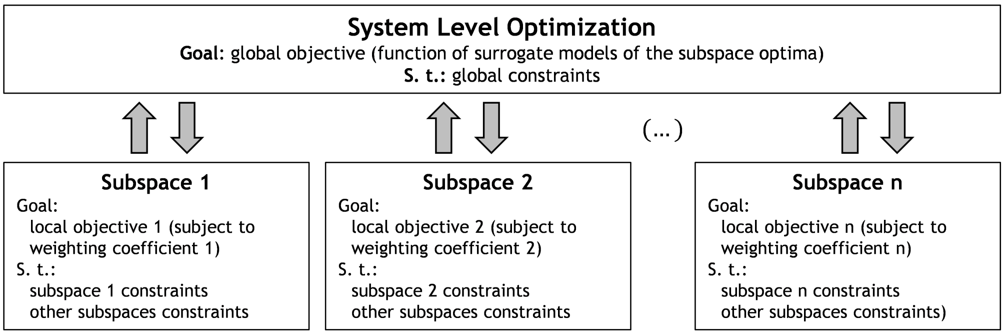

System Level Optimization
Goal: global objective (function of surrogate models of the subspace optima)
S. t.: global constraints
(...)
Subspace 1
Goal:
local objective 1 (subject to weighting coefficient 1)
S. t.:
subspace 1 constraints
other subspaces constraints
Subspace 2
Goal:
local objective 2 (subject to weighting coefficient 2)
S. t.:
subspace 2 constraints
other subspaces constraints
Subspace n
Goal:
local objective n (subject to weighting coefficient n)
S. t.:
subspace n constraints
other subspaces constraints)

Figure 2.7: Schematic representation of the Bi-level Integrated System Synthesis-2000 architecture.

The available benchmarking result for BLISS-2000 is the reusable launch-vehicle design problem of Brown and Olds (2006) [65] where this architecture outperformed other distributed architectures. Despite the advantages it presents and the promising results obtained to date, the BLISS-2000 has not been nearly as used as its predecessor. Nevertheless, the original BLISS is hardly used, being the BLISS-2000 the default choice, which often misleadingly referred as BLISS.

2.1.3.7 Exact and Inexact Penalty Decomposition (EPD and IPD)

Exact and Inexact Penalty Decomposition (EPD and IPD) architectures apply whenever there are neither system level constraints nor objectives, see Figure 2.8. In the same way as in the ATC architecture, at the subspace level, the penalty function is used to assure compatibility. Indeed, both formulations objective functions are the sum of the local objective and a penalty function which is associated with the inconsistency between the discipline information and the system information.

The difference between the EPD and the IPD is that the former includes a linear penalty function with additional variables and constraints to assure smoothness, while the latter uses a quadratic penalty function with appropriate penalty weights. Accordingly, at the subspace level, an unconstrained minimization problem is solved with the goal of minimizing the sum of the aforementioned penalty terms.

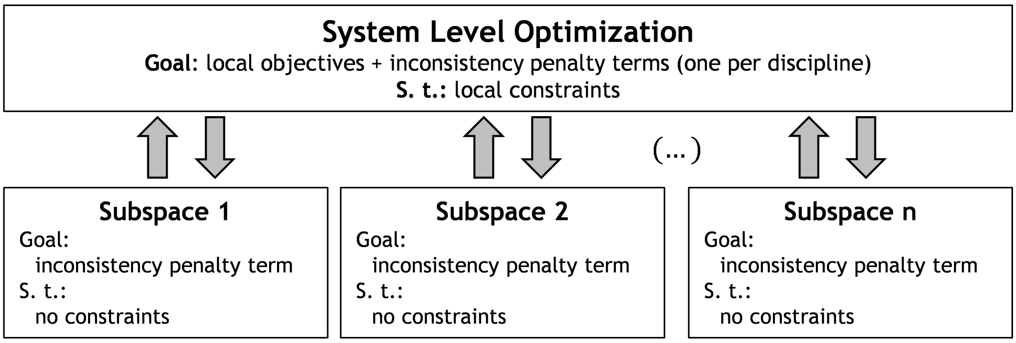

System Level Optimization
Goal: local objectives + inconsistency penalty terms (one per discipline)
S. t.: local constraints
(...)
Subspace 1
Goal:
inconsistency penalty term
S. t.:
no constraints
Subspace 2
Goal:
inconsistency penalty term
S. t.:
no constraints
Subspace n
Goal:
inconsistency penalty term
S. t.:
no constraints

Figure 2.8: Schematic representation of the Exact and Inexact Penalty Decomposition architectures.

The advantage of using EPD is that the exact solution is computed for finite values of the penalty parameter and thus ill-conditioning introduced by large penalty parameters is avoided. The underlying challenge is that exact penalty functions are non-smooth. DeMiguel and Murray (2006) [66] show that this difficulty can be overcome by employing barrier terms in conjunction with the exact penalty function.

Like CO, IPD allows the global variables to take on a different value within each of the subproblems, and makes use of quadratic penalty functions to force the global variables to asymptotically converge to the target variables. However, unlike CO, the formulation presented by DeMiguel and Murray (2006) [66] explicitly includes a penalty parameter that allows control over the speed at which the global variables converge to the target variables.

The only known benchmark study where these architectures have been compared against others by Tosserams et al (2007) [67] suggesting that their performance is highly dependent on the adopted penalty functions.

2.1.3.8 MDO of Independent Subspaces (MDOIS)

The Multidisciplinary Design Optimization of Independent Subspaces (MDOIS) applies whenever there are neither system-wide constraints or objectives nor shared design variables making the discipline subproblems fully separable, Figure 2.9.

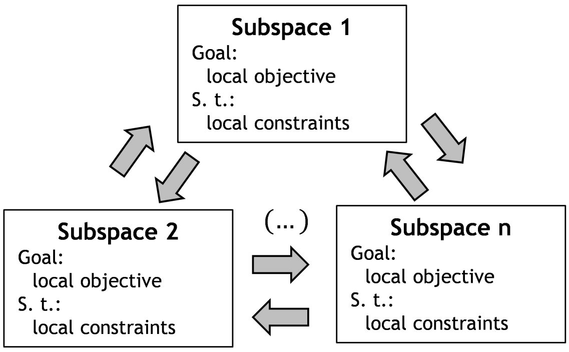


Figure 2.9: Schematic representation of the MDO of Independent Subspaces architecture.

The coupling variable copies are just parameters representing system state information. After the solution of the subspace (disciplinary) problems a full multidisciplinary analysis is performed to update all parameter values. Hence, unlike the system subproblem solved by other architectures, the multidisciplinary analysis is used to guide the discipline subproblems to a design solution.

On a benchmark study with some of the older architectures Yi et al (2008) [42], it has been shown that MDOIS requires fewer analysis calls than the MDF but more than the IDF. However, it does require more information, making it far less flexible than the MDF.

2.1.3.9 Quasiseparable Decomposition (QSD)

The Quasiseparable Decomposition (QSD) architecture was developed with the aim of solving *quasiseparable* optimization problems. In this architecture, the system objective and constraint functions are assumed to be dependent only on global variables, as discussed by Martins and Lambe (2013) [2] (i.e., shared and coupling variables), Figure 2.3.

In line with CO, QSD is an architecture where the solutions of the discipline subproblems are constraints in the system subproblem, which means that post-optimality derivatives or surrogate model approximations of the optimized discipline subproblems are required to solve the system problem.

Liu et al (2004) [68] have applied QSD with surrogate models to a structural optimization problem but no comparison with other architectures has been presented. However, a version of QSD without surrogate models was benchmarked by the de Wit and van Keulen (2007) [44]. Their results have shown that this architecture had the worst performance amongst the ones considered.

Although they have not been benchmarked, QSD versions with surrogate models are expected to yield better performance because of the smoothness introduced in the model [2].

2.1.3.10 Asymmetric Subspace Optimization (ASO)

This architecture is a new distributed MDF architecture having been motivated by high fidelity aero-structural optimization. It has been tailored to solve problems where the aerodynamic analysis may take an order of magnitude more time to run than the respective structural analysis Martins et al (2004) [69], as depicted in Figure 2.10. Obviously, this methodology is readily applicable to any other problem where there are such high discrepancies between the discipline analysis times. This does not apply when using parallel processing with load balancing among discipline analyses.

Provided the aerodynamic analysis is significantly more costly than the structural one, ASO exhibits a significant reduction in the number of calls of the aerodynamic analysis and even a slight reduction in the number of structural analysis with respect to MDF Chittick et al (2009) [70]. Contrarily, if the two disciplinary analysis take roughly the same running time, MDF is still much faster.

In conclusion, it can be said that this architecture only proves to be better than its competitors when there is a significantly computational cost variation between disciplines. Conversely, choosing between the available distributed architectures and ASO is not an easy task as the benchmarking backlog reveals to be scarce.

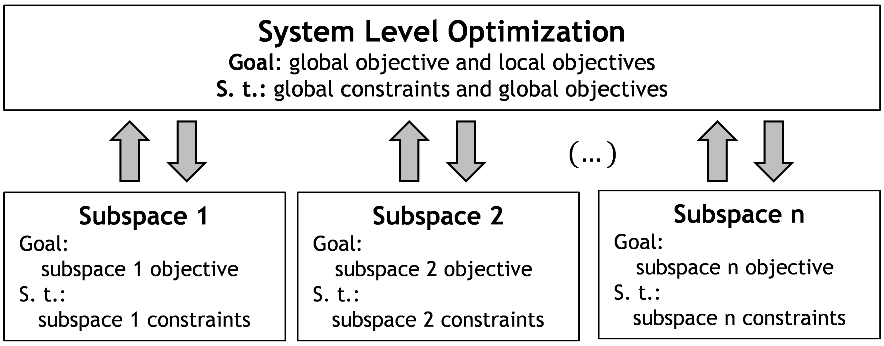


Figure 2.10: Schematic representation of the Asymmetric Subspace Optimization architecture.

2.1.4 Remarks

One of the challenges facing the MDO community is determining which architecture is most efficient for a given MDO problem or class of problems. One way is to benchmark the architectures using simple test problems, because their performance can only be fairly compared using the same problem since the most appropriate architecture often depends on the problem in hands. Much work is still to be done in this domain.

In order to facilitate that task, Table 2.1 has been built, so that the reader can easily choose two or more architectures he/she aims to compare and easily know whether there exist such works or not, and in case there have been such studies, what reference(s) should be considered.

Table 2.1: Benchmarking studies available in the literature at present time.

	Monolithic			Distributed MDF				Distributed IDF					
	SAND	IDF	MDF	CSSO	BLISS	MDOIS	ASO	CO	BLISS-2000	QSD	ATC	EPD/IPD	ECO
SAND		[43]	[43]	[43]				[43]					
IDF	[43]		[41–43, 59]	[41–43]	[41, 42]	[42]		[41–43, 59]					
MDF	[43]	[41–43, 59]		[41–43]	[41, 42]	[42]	[70]	[41–43, 59]					
CSSO	[43]	[41–43]	[41–43]		[41, 42, 44]	[42]		[41–44, 71]		[44]	[44]		
BLISS		[41, 42]	[41, 42]	[41, 42, 44]		[42]		[41, 42, 44]		[44]	[44]		
MDOIS		[42]	[42]	[42]	[42]			[42]					
ASO			[70]										
CO	[43]	[41–43, 59]	[41–43, 59]	[41–44, 71]	[41, 42, 44]	[42]			[65]	[44]	[44]		[57, 72]
BLISS-2000								[65]					
QSD				[44]	[44]			[44]			[44]		
ATC				[44]	[44]			[44]		[44]			
EPD/IPD													
ECO								[57, 72]					

2.2 Optimization Algorithms

Despite not being the core of this thesis, it is unavoidable to briefly talk about optimization algorithms, since these have an indelible relevance in MDO. In fact, the effectiveness and efficiency of the final solution largely depends on an appropriate choice between the different optimization algorithms available.

As described in Section 1.2, the traditional design approach used to be carried out through the use of charts and graphs which have been developed over many years of experience. These methods are usually an efficient means of obtaining a reasonable solution to traditional design problems. However, as the design task becomes more complex, one tends to rely more heavily on computers. One can change one or more design variables and rerun the computer program to see if any design improvement can be obtained. One then takes the results of many computer runs and plot the objective and constraint values versus the various design parameters. From these plots one can interpolate or extrapolate to what is believed to be the optimum design.

If there are over three design variables, the true optimum may be extremely difficult to obtain graphically. Thus, the automation of the optimization process comes as a corollary. Mathematical programming provides a logical and formal framework for undertaking an automated optimization process. As with any methodology, some advantages and limitations to the use of numerical optimization techniques are present, as detailed by Vanderplaats (2007) [73].

Advantages

- The reduction in design time, contributing to increasing the overall design efficiency.
- Optimization provides a formal logical design procedure.
- It becomes possible to have a wide variety of design variables and constraints which are difficult to visualize using graphical or tabular methods.
- Optimization virtually always yields some design improvement.
- It is not biased by intuition or experience in engineering. Therefore, the possibility of obtaining improved, nontraditional designs is enhanced and the designer experience affects the quality of the result less.

Limitations

- Computational time increases with the number of design variables. If one wishes to consider all possible design variables, the cost of automated design may become hard to manage. Also, as the number of design variables increases, these methods tend to become numerically ill-conditioned.
- If the analysis program is not theoretically precise, the results of optimization may be misleading, and therefore the results should always be checked very carefully. Optimization will invariably take advantage of analysis errors in order to provide mathematical design improvements.
- Most optimization algorithms have difficulty in dealing with discontinuous functions. Also, highly nonlinear problems may converge slowly or not at all. This requires particular carefulness in formulating the automated design problem.
- It can seldom be guaranteed that the optimization algorithm will obtain the globally optimum design. Therefore, it may be worthwhile to restart the optimization process from

several different points to increase the chances of obtaining the global optimum.

2.2.1 Gradient-Based Algorithms

The gradient-based optimization techniques are useful in finding the optimum solution of continuous and differentiable functions. These are analytical methods and make use of differential calculus in locating the optimum solution. The classical methods have limited scope in practical applications as some of them involve objective functions which are not continuous and/or differentiable. Yet, the study of these classical techniques of optimization form a basis for developing most of the numerical techniques that have evolved into advanced techniques more suitable to today's practical problems.

According to Witherell (2006) [74], the types of optimization algorithms can be categorized as presented in Figure 2.11. These are divided in continuous and discrete optimization.

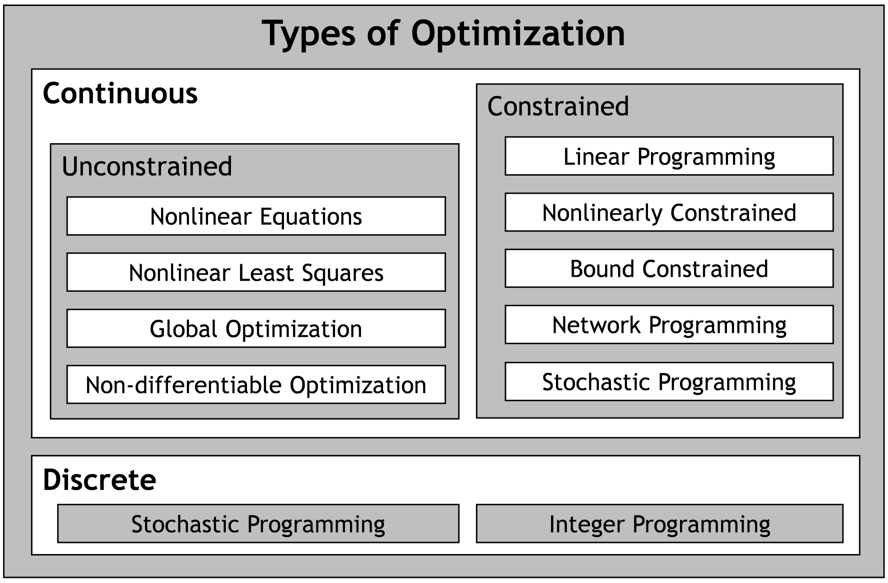


Figure 2.11: Types of optimization algorithms.

2.2.2 Heuristic Methods

According to Pearl (1984) [75], «*The study of heuristics draws its inspiration from the ever amazing observation of how much people can accomplish with that simplistic, unreliable information source known as intuition.*».

Heuristics methods have been developed for finding an approximate solution when classic methods fail to find an exact solution. Heuristics are typically used to solve complex (large, nonlinear, non-convex) multivariate combinatorial optimization problems which are usually hard to solve to optimality. Unlike gradient-based methods in a convex design space, heuristics are not guaranteed to find the true global optimal solution in a single objective problem, but should find many good solutions. Heuristics are good at dealing with local optima without getting stuck in them while searching for the global optimum, contrarily to what happens in non-heuristics based algorithms.

Heuristics are problem-dependent techniques that try to take full advantage of the particularities of this problem. However, they usually get trapped in a local optimum and thus fail, in general, to obtain the global optimum solution. Meta-heuristics techniques, on the other hand, are problem-independent techniques. Hence, they can easily be used as black boxes.

Metaheuristics techniques can be categorized in Population based vs Trajectory based. Population based metaheuristics works with the concept of creating a solution that mix components of good solutions. Trajectory based metaheuristics works with the concept of creating a solution and iteratively improving it. With Figure 2.12 ([76]), it is possible to visualize this dichotomy and other possible parameters for comparison.

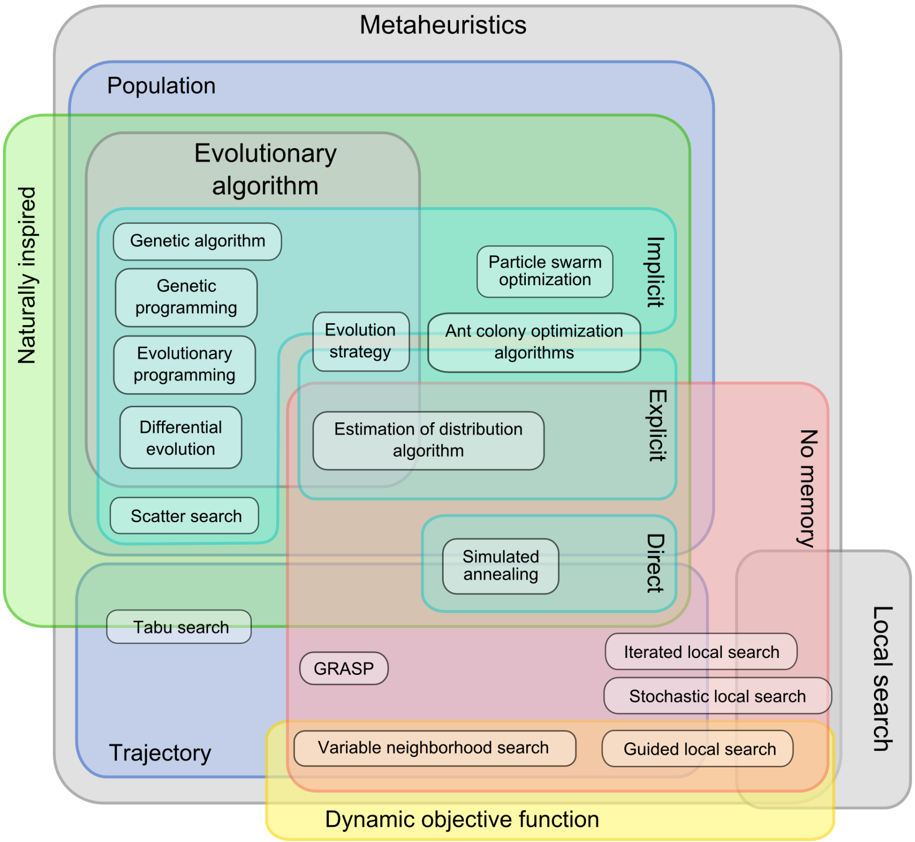


Figure 2.12: Metaheuristics Taxonomy.

With the ever-growing computational optimization applications along with the need to solve more complicated domains, one could expect that the non-classical methods would flourish. However, as computational capacity increases, so does the degree of complexity of the disciplinary models and therefore the classical and non-classical methods have complementary roles. Choosing the most suitable optimization for the problem in hands is a very important and demanding task which is far from straightforward, as the best algorithm from one problem can be a complete failure for another problem.

2.3 Morphing Technologies

Since long before the dawn of aviation, humans have dreamt of ways of flying, with their primary inspiration being birds. By directly comparing aircraft with nature, designers still seek inspiration in birds. These are able to rapidly change shape to transition from efficient cruise to aggressive maneuvering and precision descents. Aircraft morphology allows a wide range of wing configurations, each of which can be used for a particular flight scenario.

The idea of changing the wing shape or geometry is far from new. In its broadest sense, morphing technologies are defined as means of changing the aircraft shape to adapt or to allow different flight conditions. As already introduced in section 1.1, the first successful use of a morphing concept can be drawn to the first heavier-than-air flight (1903), Weisshaar (2006) [77] - where the Wright Brothers, managed to control the aircraft bank angle using wing warping (twist motion of the wing). The usage of flaps, slats, ailerons, variable pitch prollers and retractable landing gears would also fit into this definition. In fact, many authors consider these devices as early morphing aircraft technologies, although the morphing aircraft technologies are usually associated with wider and more recent shape changing concepts. Indeed, some authors consider flaps, slats, ailerons and retractable landing gears conventional aircraft technologies and make the distinction between them and the morphing aircraft technologies, considering the former associated to fixed-wing technologies and based on traditional materials and mechanical components, such as hinged surfaces, whilst the latter are related to innovative materials and actuating technologies. It is this narrower definition of morphing that is adopted hereinafter.

Barbarino et al (2011) [78] have divided morphing wing concepts in three categories: airfoil, planform and out-of-plane, as seen in Figure 2.13.

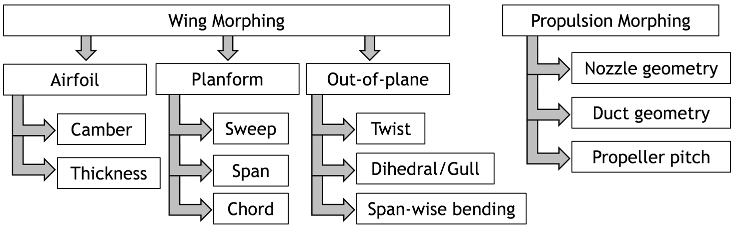


Figure 2.13: Schematic representation of the different wing morphing categories.

In recent decades, there has been an increasing academic, civil and military interest in these solutions. They enable significant improvements to the overall aircraft performance and multimission capabilities, Joshi et al (2004) [79]. However, historically, morphing solutions have encompassed penalties in terms of cost, complexity and/or weight although, in several situations, these were overcome by system-level benefits. Not considering these penalties, and provided that the morphing design task was successful, it would always make sense. For most applications, however, there is a crossover point beyond which the energetic penalty for not morphing outweigh the aforementioned penalties. Duly evaluating the profitability of morphing is thus a paramount designer's assignment.

Performance gains thanks to an enlarged flight envelope is the main and only granted consequence of adopting properly designed wing morphing technologies. For military aircraft this may be in many situations an important enough asset to justify these solutions because the same vehicle can perform different roles in a mission that would otherwise need two or more different aircraft, meaning that more versatile vehicles compensate for the additional complexity and weight. Contrarily, the commercial aircraft market focus is obviously far more directed towards financial savings through enhanced energetic performance. If a given aircraft has a wider flight envelope with respect to its fixed-wing homologue but that results in an energetic penalty, it is highly likely that it will be disregarded for all commercial applications. The same applies to business jets, because energetic penalties will result in lower range and endurance.

Although it is clear that most large shape modification techniques have been developed for military applications, because of the ruling of performance over energetic efficiency, in recent years, the focus has moved to small aircraft, mostly UAVs. Besides being the class of flying vehicles that is expected to grow faster in the upcoming decade as the European Aviation Safety Agency (EASA) [80] and the Federal Aviation Administration (FAA) [81] move towards the adoption of comprehensive regulatory frameworks for their flight in non-segregated airspace, UAVs bring up some additional benefits. In Portugal, the *Autoridade Nacional de Aviação Civil* (ANAC) has already started issued UAV-specific regulations [4].

The move towards UAVs results from greater efficiency requirements and a short time-to-deliver because of reduced certification issues and qualification tests. UAVs are also cheaper and safer solutions as test-benches for the validation and in-flight assessment of new concepts, such as morphing solutions. Furthermore, the lower aerodynamic loads on UAVs also increases the number of potential morphing applications.

Despite the little impact of morphing to date, the relevant developments in 'smart' materials witnessed over the last years may contribute to overcome the limitations and enhance the benefits from the already developed morphing solutions Barbarino et al (2011) [78]. The challenge is to design a structure that is capable of withstanding the prescribed loads, but is also able to change its shape: ideally, without distinction between the structure and the actuation system. Moreover, the conceptual and preliminary design approach has to be modified to allow for morphing to be considered and optimized early in the design process rather than being retrofitted Ajaj et al (2013) [82].

The need for considering aircraft optimization with morphing technologies and smart structures will require multidisciplinary thinking and a meaningful overall complexity step-up, starting at the earliest design stages.

Subsections 2.3.1 through 2.3.3 include a brief overview on some of the most significant developments of each of the considered morphing categories (airfoil, planform and out-of-plane). Although the use of morphing is not limited to fixed-wing aircraft, other morphing applications lie out of the scope of this research work and are therefore not discussed. Notice that, in this context, the term fixed-wing aircraft (airplanes) has been used as opposed to rotary-wing aircraft (helicopters).

2.3.1 Airfoil Morphing

Airfoil morphing, namely airfoil camber morphing is doubtlessly the dominant subsonic morphing research topic when compared with wing-planform and out-of-plane morphing, following the wing morphing categorization featured in Figure 2.13. Airfoil morphing is mainly concerned with camber variation, Figure 2.14, although there have also been works on variable airfoil thickness mechanisms. The motivation behind this type of morphing mechanism comes from the interest in varying airfoil camber in subsonic flight regimes in order to obtain the best lift-to-drag ratio for each flight condition.

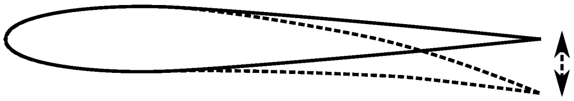


Figure 2.14: Variable camber airfoil representation, using a trailing edge mechanism.

There are multiple ways of varying the airfoil camber. From the different solutions presented to date, the variable camber mechanisms can be divided in three different groups, depending on their actuation mechanisms: conventional actuators, shape-memory alloys (SMAs) and Piezoelectric (PZT) actuators. In their review of morphing aircraft, Barbarino et al (2011) [78] concluded that SMAs are more common and more likely to be used for large vehicles, whereas PZT actuators are more interesting for small vehicles like most of the UAVs and Micro air vehicles (MAVs). Despite the large number of research works devoted to variable camber mechanisms, they have been mostly focused on 2D aerodynamic configurations usually overlooking the skin technical issues, which largely justifies the scarce number of flight tests to date.

2.3.2 Planform Morphing

Wing planform morphing depends primarily on three parameters, which can work individually or combined, namely span (b), chord (c) and sweep (Λ). These parameters are of primary relevance since they all affect the lift-to-drag ratio. The different types of in-plane morphing are shown in Figure 2.15, Basaeri et al (2014) [83].

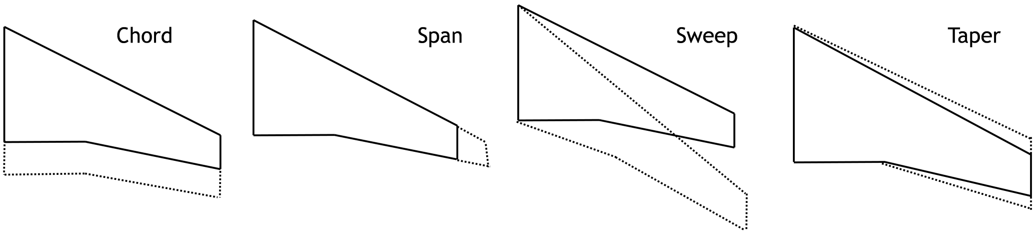


Figure 2.15: In-plane morphing configurations.

Military applications, particularly for UAVs that must loiter during surveillance and rapidly turn into high-speed dash mode to move away from the reconnaissance area or to attack a given target, have lead to the investigation of variable span and sweep. Conversely, chord morphing has mostly been used in helicopter rotor blades [78].

Variable sweep has been implemented on real fixed-wing aircraft more than any other morphing concept [78]. The first application of this morphing methodology (with the American Bell X-5) was motivated by the development of supersonic aircraft, as it was found that wing sweep angles enabled an increased efficiency at high subsonic, transonic and supersonic flight regimes. It also contributes to an enlarged flight envelope, which is obviously very important for military applications. There have been a number of military examples of use of variable sweep morphing: the Americans F-14 Tomcat and Rockwell B-1 Lancer; the Soviets Mikoyan-Gurevich (MiG-23 and MiG-27), Sukhoi (Su-17, 20, 22 and 24) and Tupolev (Tu-22M and Tu-160); and the European Panavia Tornado. More recent research has focused on the application of this morphing methodology to Micro Air Vehicles (MAVs) and UAVs for the sake of aerodynamic efficiency and multi-mission capabilities.

Despite lacking maneuverability, high aspect ratio wings tend to be more efficient and can thus deliver higher range and/or endurance than low-aspect ratio wings. Contrarily, the latter have greater maneuverability but a poorer aerodynamic performance. The idea behind variable-span-wing (VSW) mechanisms comes from the potential of having the advantages of low and high aspect ratio solutions in the same design. As already stated, military applications are obviously of primary interest in this context. Having long range vehicles which are able to fly very long-distances non-stop but which are also able to deliver high maneuverability and higher velocities is obviously a crucial requirement. Most of the solutions adopted to date, have used the telescopic wing concept, which was firstly introduced by Ivan Makhonine, where the wing outer panel telescoped inside the wing inner panel. The first flight of the MAK-10, took place in 1931, as discussed by Weisshaar (2006) [77]. The results of the works developed to date have shown that the span increase has been considered to enhance cruise performance at low speeds without the structural and dynamic problems of high aspect ratio wings [78], as at higher speeds and during more demanding maneuvering requirements, the VSW is retracted. Variable span morphing has also been successfully used in roll control by means of an asymmetrical wing span increase.

From the planform morphing methodologies, the variable chord has been investigated. Conventional wings with variable chords usually rely on either leading or trailing edge mechanisms actuated by screw systems. Additionally, in fixed-wing aircraft - as opposed to rotary-wing - the presence of spars, fuel tanks and other systems increase the complexity of adopting such solutions, which justifies a wider use of such solution in the latter category of vehicles. Amongst fixed-wing aircraft, the first aircraft ever to enable chord variation was most likely the Baksaev LIG-7 designed by the Soviet Union in 1937 [77]. It consisted of an unusual and innovative morphing aircraft with two dimensional in-plane actuation. It was a telescopic mechanism with six chord-wise overlapping wing sections, which were retracted and extended using tensioned steel wire, enabling an overall area change of 44%. Overall, there is a small number of works targeting this morphing solution and are mostly based on small aircraft and rotor blades [78]. The limited investigation on variable wing chord morphing for fixed-wing aircraft is more due to the significant technical challenges it involves than to its low performance benefits.

2.3.3 Out-of-plane Morphing

With the exception of wing twist, which is the oldest type of morphing, out-of-plane morphing is the least common solution. The different types of out-of-plane morphing are shown in Figure 2.16, Basaeri et al (2014) [83]. However, due to aeroelastic problems wing twist was disregarded for about 80 years [78]. The significant developments in aerospace materials, namely with the use of composites, have contributed to overcome this problem.

The main reason behind the adoption of variable twist solutions is that twist has shown

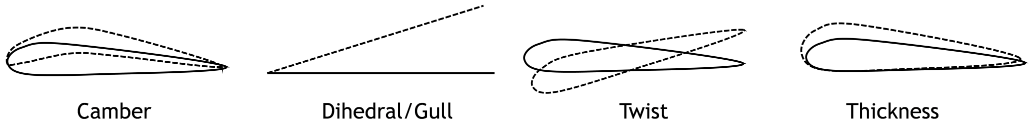


Figure 2.16: Out-of-plane morphing configurations.

to have a significant impact on the lifting surface aerodynamics. It has been realized that wing morphing can produce significant performance impact without the need of more complex structures such as the ones required for variable span and sweep. In addition, wing twist can have several different effects, like alleviating gust and/or maneuver load, increasing or decreasing the wing lift coefficient and even replacing conventional control surfaces, by using asymmetrical twist.

Another out-of-plane type of morphing is dihedral/gull and spanwise bending. The interest in these morphing solutions came from the interest in optimizing winglets [78]. Most recent studies have focused on continuous geometric morphing which does however require complex mechanisms with flexible skins, as discussed by Manzo et al (2005) [84] and Sofla et al (2010) [85].

2.3.4 Propulsion Morphing

Besides morphing wing solutions, there is also what can be called propulsion morphing, which refers to mechanisms built on the propulsive system that can change its performance depending on the mission/flight stage requirements.

Amongst the propulsion morphing solutions devised to date, variable pitch propeller (VPP) is undoubtedly the most used. VPP is a type of propeller with blades that can be rotated around their long axis to change the blade pitch. These propellers can adapt the blade angle-of-pitch as a function of the flight regime to provide an improved performance and enlarge its operational envelope. Ultimately, VPP can lead to reversible propellers (those where the pitch can be set to negative values) which can also create reverse thrust for braking or going backwards without the need to change the direction of shaft revolution.

Propelling nozzles may have variable geometry to give different exit areas to control the operation of the engine. When after-burning engines are equipped with a convergent-divergent nozzle the throat area is variable. Nozzles for supersonic flight speeds, at which high nozzle pressure ratios are generated, also have variable area divergent sections [86].

The F119 is an after-burning turbofan engine developed by Pratt & Whitney for the Lockheed Martin F-22 Raptor advanced tactical fighter. This engine includes vectored thrust using variable geometry nozzles.

2.3.5 Combinations of Morphing

It is worthwhile to mention some recent developments that make use of at least two morphing concepts combined. Neal et al [87] have developed a wing that is able to change its span and sweep. Flanagan et al (2007) [88] have built a bat-shaped wing, which is able to continuously vary its span, chord, sweep and has already been successfully flown. Vale et al (2006, 2007) [89, 90] and Gamboa et al (2007) [91] have created a variable span and variable airfoil chord and shape by the use of extendable ribs and spars. Leite et al (2009) [92] have also designed and analyzed a telescopic wing with airfoil shape change capability.

Finally, the reader's attention is drawn to the fact that, in the forthcoming chapters, the term fixed-wing will be used as opposed to variable span wing (VSW).

Chapter 3

Analysis Models

3.1 Introduction

The present Chapter aims at describing the low-fidelity disciplinary models that have been used for the two codes developed in this thesis, which are presented and assessed in Chapters 3 and 4.

This chapter starts with a brief explanation of the Programming Language adopted, Section 3.2. Section 3.3 is focused on describing the analysis models themselves. It includes a comprehensive overview of all developed disciplinary models, namely, Aerodynamics (3.3.1), Propulsion (3.3.2), Weight (3.3.3), Static Stability (3.3.4) and Dynamic Stability (3.3.5). The aforementioned sections also include a description of some off-the-shelf low-fidelity models which are also used. Furthermore, all numerical methods implemented for data handling purposes and all modifications to the standard models which were necessary for the assessment of morphing concepts are also explained.

One of the most distinguishing features of the two codes developed is their ability to perform a mission-based optimization of the vehicle's geometry. Therefore, a detailed description of the four main generic mission stages, namely, take-off, climb, cruise/loiter and descent is included in Section (3.4). These sections also include a description of the flight mechanics and the built-in iterative procedures.

All assumptions and simplifications made in the adopted physical models are also presented and discussed throughout the forthcoming sections. Other analysis models could have been used, however, the ones presented are thought to provide a good trade-off between efficiency and effectiveness in the context of preliminary design optimization. As the analysis models have been developed and implemented in a modular way, it is possible to easily replace them in future uses.

3.2 Programming Languages

For programming purposes, the Fortran language has been adopted for all the mathematics and physics applications. This programing language name is derived from FORmula TRANslation indicating that its primary purpose was to translate scientific equations into computer code, as discussed by Chapman et al (2008) [93]. Indeed, it is a general-purpose programming language especially suited to numeric, scientific and engineering computing. Before its development (1950s), all computer programs were generated by hand in machine language - a rather inefficient and error-prone process. Hence, Fortran came to dominate scientific areas of programming early on and has been in continuous use ever since, standing at the forefront of developments in areas like numerical weather prediction, finite element analysis, computational fluid dynamics, computational physics and computational chemistry. Not only is it one of the most popular languages in the area of high-performance computing but it is also the language used for benchmarking the world's fastest supercomputers.

The programming language decision making process was also influenced by having access to previously developed Fortran code. The programming language compatibility with previous work fostered a faster development of the two main codes of the current research work: PARROT and MTOP.

The graphical user interface (GUI) for both PARROT and MTOP has been made using the open source XFLR5 GUI, which is programmed in C++ language. The main reasons for using the XFLR5 framework were the fact that it is an open source code, easy to handle and already having expedite methods for the aerodynamic analysis of airfoils (using XFOIL).

3.3 Disciplinary Analyses

This section is devoted to the presentation of the different disciplinary analysis models which are used in the software developed throughout this thesis, whose methodologies are presented in Chapters 3 and 4.

As already highlighted, design optimization methodologies aim at enhancing both the efficiency and the effectiveness of the preliminary design stage of aircraft design, thus contributing to guide the designer decision making from the earliest stages. As statistician George Edward Pelham Box (1919-2013) once said:

«Essentially, all models are wrong, but some are useful.»

The big challenge lies in choosing the most suitable mathematical model to represent the aircraft physics for each problem. Given the purposes of both the parametric aircraft design optimization and the multilevel design optimization methodologies developed, an underlying concern when choosing between the different possible models is their computational cost. All in all, it can be broadly stated that there is a crossover point beyond which the quality of the results do not outweigh the increased computational cost. Nevertheless, this crossover point is highly dependent on the problem in hands.

The disciplinary models presented throughout the next Subsections are low-fidelity models, which are not expected to deliver high accuracy but rather to provide reliable results for guiding the designer choices improving the design optimization process assertiveness in an expedite way. This choice has been taken bearing in mind that the results are to be used in a preliminary design stage.

Lastly, it is worthwhile to mention that no loss of generality regarding the design optimization methodologies is expected from the use of low-fidelity models. Indeed, provided that their usefulness is attested with these models, it can be considered that these are valuable design optimization tools. Hence, different user provided analysis models (low- and medium-fidelity) can thus easily replace the ones developed.

3.3.1 Aerodynamics

In all conventional subsonic aircraft with medium/high aspect ratio wings ($\Lambda > 6$) the major contribution to the overall aerodynamic performance comes from the wing airfoil and therefore its careful selection is paramount. For this reason, the driving parameter of the design study is the cruise/loiter airfoil lift coefficient (C_l). This value is user defined but is bounded by the airfoil's stall lift coefficient and the cruise speed interval given by the user, according to the vehicle's operational requirements.

3.3.1.1 2D Aerodynamic Coefficients

For the airfoils analysis – and given the low Reynolds number expected for the UAVs that aim to be optimized – the XFOIL software has been chosen, which provides good results for subsonic Mach number. The user selects an airfoil for each of the three lifting surfaces (wing, horizontal tail and vertical tail) and a set of different Reynolds numbers, which are run in XFOIL for a given range of angles-of-attack.

As each mission stage is being assessed, the airfoil aerodynamic coefficients (C_l, C_d and C_m) for a specific Reynolds are required. There are multiple ways of estimating the aerodynamic coefficients from the matrices generated by XFOIL, where these are functions of the Angle-of-Attack (AoA) and Reynolds number (Re). The forthcoming paragraphs will describe the two methods used.

The simplest approach consists of either interpolating or extrapolating the aforementioned matrices to estimate the aerodynamic coefficients in the AoA and (Re). This approach has been adopted for fixed-wings, because it is easy to implement and the computational cost is not very high.

One of the goals of this thesis is to address morphing technologies, and since one of the most relevant wing shape changing methodologies is achieved by changing the wing airfoil camber with a trailing edge continuous flap mechanism, a multilinear interpolation approach would be required to estimate the 2D aerodynamic coefficients at each design point. However, that could have a high computational cost, specially taking into consideration the significant number of function evaluations that would required for the gradient-based multilevel optimization methodology implemented in the MTOP code (Chapter 4). Furthermore, not only would one need to interpolate the aerodynamic coefficients for the AoA and (Re), but also for the flap chord-wise fraction (c_f) and flap deflection (δ_f). Nevertheless, after several simulations, it has been found that, in the problem in hands, a four variable multilinear interpolation was after all not particularly bad in terms of computational efficiency and has thus been adopted.

The continuous variable camber flap airfoil aerodynamic coefficients must be estimated for a number of flap relative chord and flap deflection combinations which can thereafter be interpolated. As such, the airfoil geometry must be determined for every combination of flap relative chord and flap deflection to be analyzed, which means that an expedite way of estimating the flapped airfoil geometrical definition must be devised. The approach implemented and used for the current developments is presented in Appendix B.

Multilinear Interpolation

As already stated, a multilinear interpolation has been adopted for determining the aerodynamic coefficients (C_l, C_d and C_m) as a function of (AoA,Re,c_{flap},δ_{flap}).

Denoting by (f) a generic function that aims to be interpolated, by (x^n) the interpolation variable of the (n^{th}) dimension, and by (x_0,x_1 and x_2) the variable values immediately before, after and on the particular point being interpolated, respectively, in a multidimensional space, the interpolation of variable (x_2^n) comes from Equation 3.1:

$$\begin{aligned} f(x_2^1, x_2^2, \dots, x_2^n) =& f(x_2^1, x_2^2, \dots, x_1^n) + \\ &+ \left[f(x_2^1, x_2^2, \dots, x_1^n) - f(x_2^1, x_2^2, \dots, x_0^n) \right] \frac{(x_2^n - x_0^n)}{(x_1^n - x_0^n)} \end{aligned} \tag{3.1}$$

The number of points delimiting this multidimensional polytope is equal to 2^n, with n being the number of problem dimensions and the number of linear interpolations to be performed is equal to $\left(\sum\limits_{i=1}^{n} 2^i \right)$.

A more detailed explanation of the multilinear interpolation has been included in Appendix C.

3.3.1.2 3D Aerodynamic Coefficients

Lifting Surfaces

In order to obtain the lifting surfaces aerodynamic coefficients (3D), the following procedure has been adopted, which is an iterative solution for finding both the induced angle-of-attack (α_i) and the 3D lift coefficient (C_L), using:

$$\begin{cases} \alpha_i = \frac{C_L}{\pi \Lambda}(1 + \tau) \\ C_L = C_l cos(\alpha_i) \end{cases} \tag{3.2}$$

, where τ is a correction to the induced angle-of-attack arising from the fact that the lift distribution is not elliptical, which has been approximated as a function of the Oswald efficiency factor e_0:

$$1 + \tau = \frac{1}{e_0} \tag{3.3}$$

The Oswald efficiency factor is a parameter in a symmetrical drag polar that can measure the aerodynamic efficiency of a wing. If one assumes a symmetrical drag polar, the 3D drag coefficient (C_D) is the sum of the profile drag (C_d) with the lift induced drag, which accounts for the aspect ratio (Λ) and wing planform shape via the Oswald coefficient:

$$C_D = C_d + \frac{C_L^2}{\pi \Lambda e_0} \tag{3.4}$$

, where (Λ) is the wing aspect ratio, which is defined by:

$$\Lambda = \frac{b^2}{S} \tag{3.5}$$

In reality, the wing drag polar is not symmetric because the profile drag (C_d) is also dependent on the lift coefficient (C_L), which means ($C_d = f(C_L)$). This means that one can re-write Equation 3.4:

$$C_D = C_d + \frac{C_L^2}{\pi \Lambda e} \tag{3.6}$$

, where all the physical quantities being the same with the exception of (e), which is usually called span efficiency factor to differ it from the Oswald factor. The difference between these two quantities lies on assuming constant profile drag for the Oswald factor and a profile drag being a function of the wing lift coefficient for the span efficiency factor.

As the airflow is viscous and the profile drag depends on the wing lift coefficient, the span efficiency factor (e_0) can be calculated for each (C_L). From experimental or numerical curves of (C_D vs. C_L) and (C_d vs. C_L):

$$e = \frac{C_L^2}{\pi \Lambda (C_D - C_d)} \tag{3.7}$$

This factor has a maximum value equal to unity in inviscid flow in planar wings (without dihedral and torsion) and an elliptical chord distribution (generating a constant down-wash). In viscous flow with optimized geometries (elliptical sweep, variable dihedral, wingtips, etc) it is possible to have a span efficiency factor greater than unity ($e > 1$).

Finally, and assuming a wing without twist, taper or sweep, the lifting surface pitching moment coefficient at zero lift can be assumed equal to the airfoil pitching moment coefficient at zero lift. The lifting surfaces pitching moment will be given by:

$$C_m = C_{m_{ac_{wb}}} + C_{L_{wb}}(\bar{h} - h_{n_{wb}}) \tag{3.8}$$

Fuselage

In order to determine the fuselage parasite drag, the equivalent skin-friction method is adopted because a well-designed aircraft in subsonic cruise will have parasite drag that is mostly skin-friction drag plus a small separation pressure drag. According to Raymer (1989) [1]. For laminar ($Re < 1000$) and turbulent flow ($Re \geq 1000$), respectively:

$$C_f = \frac{1.328}{\sqrt{Re_x}}, Re < 1000 \tag{3.9}$$

$$C_f = \frac{0.455}{(\log_{10} Re_x)^{2.58}(1 + 0.144M^2)^{0.65}}, Re \geq 1000 \tag{3.10}$$

Equations (3.9) and (3.10) refer to the local friction coefficient and must be integrated along the characteristic length to obtain the total friction coefficient, with (M) representing the Mach number. Several corrections for the local Reynolds number ($Re_x = \frac{Vx}{\nu}$) may be found in Raymer (1989) [1] and Corke (2002) [94] which account for early transition on rough surfaces. The total viscous drag can be computed from Equation (3.11), where (q) is the dynamic pressure, (S_{wet}) the fuselage surface area (wetted area), (C_f^{total}) is the total friction coefficient, (*F*) is the fuselage form factor and (Q) the interference factor, which accounts for the fact that parasite drag is increased due to the mutual interference with the lifting surfaces and other components. This effect is usually negligible in the case of the fuselage ($Q = 1$) [1].

$$D_{fus} = qS_{wet}C_f^{total}\mathit{F}Q \tag{3.11}$$

The form factor (*F*) is a function of the fuselage characteristic dimensions [1], as per Equation (3.12).

$$\mathit{F} = 1 + \frac{1}{\left(\frac{l_{fus}}{d_{fus}}\right)^3} + \frac{\frac{l_{fus}}{d_{fus}}}{400} \tag{3.12}$$

, where (l_{fus}) refers to the fuselage length and (d_{fus}) refers to the fuselage average diameter.

Miscellaneous

In what concerns to miscellaneous aerodynamic drag, it is at least worthwhile to consider the landing gear contribution for the overall drag since most UAVs do not have retractable landing gears and they do not have streamlined shapes like the lifting surfaces or the fuselage. Its drag is best estimated by comparison to test data for a similar landing gear configuration [1]. In case that data is not available, the gear drag can be estimated as a summation of the wheels, struts and other components using typical drag coefficients for each component as featured in reference [1].

Total Aerodynamic Coefficients

Figure 3.1 depicts the most relevant forces and moments acting on the wing and on the horizontal tail. As it is common, the aerodynamic forces and moments acting on the lifting surfaces have been placed on the respective aerodynamic center (AC) - which is, by definition, the chord fraction position where its aerodynamic pitching moment is independent from the angle-of-attack (α_w, α_{ht}).

The total aerodynamic coefficients, which include the contributions of the lifting surfaces (wing, horizontal tail and vertical tail), fuselage, propulsion and miscellaneous are computed using Equations 3.13 through 3.15.

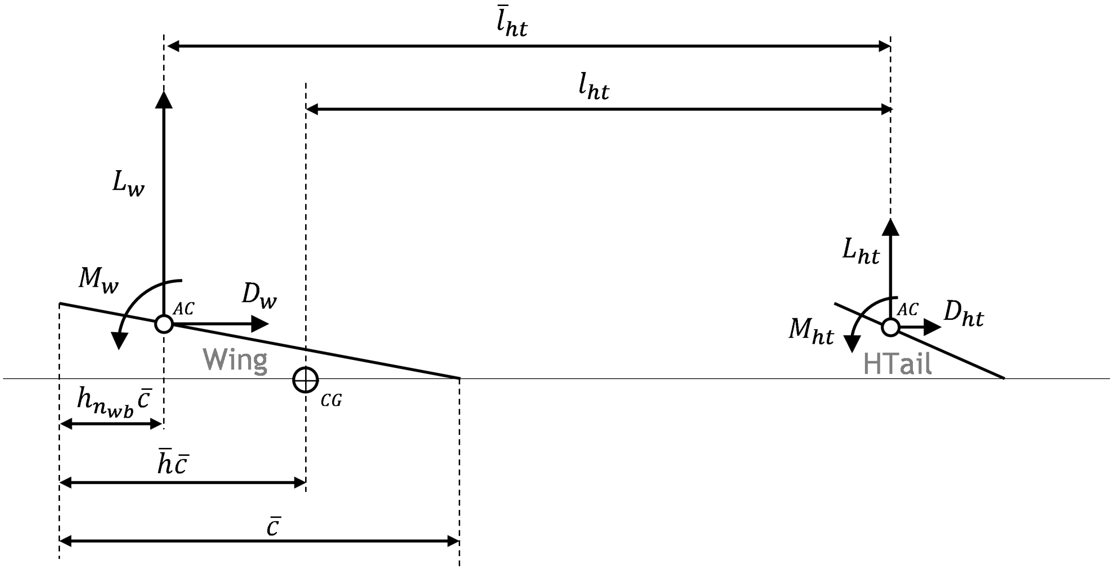


Figure 3.1: Schematic representation of the acting forces and moments on the wing and horizontal tail.

$$C_L = C_{L_w} + \frac{S_{ht}}{S} C_{L_{ht}} \tag{3.13}$$

$$C_D = C_{D_w} + \frac{S_{ht}}{S} C_{D_{ht}} + \frac{S_{vt}}{S} C_{D_{vt}} + \frac{S_{wet}}{S} C_{D_{fus}} + \frac{S_{misc}}{S} C_{D_{misc}} \tag{3.14}$$

$$C_m = C_{m_{ac_{wb}}} + C_{L_{wb}}(\bar{h} - h_{n_{wb}}) - \bar{v}_{ht} C_{L_{ht}} + C_{m_P} \tag{3.15}$$

, where C_{m_P} is the propulsive system pitching moment coefficient, $\bar{v}_h t = (\bar{l}_{ht} S_{ht})/(\bar{c}S)$ and the subscript $misc$ identifies the miscellaneous sources.

3.3.2 Propulsion

This subsection is devoted to the propulsion model. It includes a description of the electric motor 3.3.2.1 and combustion engine 3.3.2.2 models and also the propeller performance model 3.3.2.3. Since the design optimization routines are expected to be used in the design of small to medium-size low speed UAVs, it was found unnecessary to consider other propulsive systems.

Before presenting the aforementioned sections, a more detailed definition of the different power sources than the one presented in the list of symbols is featured in Table 3.1 .

In unaccelerated leveled flight, the required power is given by Equation (3.16), (D) is the vehicle drag and (V) is the vehicle's airspeed.

Table 3.1: Power sources nomenclature explanation.

Symbol	Description	Relevant formula
P_{comb}	Combustion engine power, actual power consumption	(3.32)
P_{ele}	Electric power consumed by the motor, ESC and battery(ies)	(3.28)
P_{eng}	Power delivered by the engine	(3.32)
P_{mot}	Electrical power consumed by the motor	(3.25), (3.26), (3.27)
P_{prop}	Propeller/propulsive power delivered	(3.34)
P_{req}	Power required for flight	(3.16)
P_{shaft}	Propeller shaft power	(3.34), (3.40)
P_{sys}	Aircraft systems' power which can be generated by the engine/motor shaft using an alternator	(3.33)

$$P_{req} = D.V \tag{3.16}$$

As for the energy, it is the time integral of the power consumption between two instants t_1 and t_2.

$$E = \int_{t_1}^{t_2} P(t)dt \tag{3.17}$$

However, it is possible to discretize this time domain and estimate the energy as a product of the power per the time interval. This numerical approach is exact if the power is constant between the two instants of time under consideration, whereas it becomes an approximate solution otherwise. Accordingly, the energy becomes the product of the power per the respective time interval.

$$E_i = P_i \delta t_i \tag{3.18}$$

3.3.2.1 Electric Motor

Let (I) be the electric current, the battery voltage (U_{bat}) is the battery voltage necessary for the motor to be rotating in idle (U_{bat_0}) less the voltage losses due to the battery internal resistance (R_{bat}^{ele}) and the Electronic Speed Control (ESC) internal resistance (R_{esc}^{ele}), which would take the form:

$$U_{bat} = U_{bat_0} - IR_{bat}^{ele} - IR_{esc}^{ele} \tag{3.19}$$

The voltage being transmitted to the motor (U) itself is the battery voltage times the throttle setting (δ_{th}), equal to zero for the motor-off and equal to one for maximum throttle setting:

$$U = U_{bat}\delta_{th} \tag{3.20}$$

The effective current (I_{eff}) and voltage (U_{eff}) working on the motor at any time is given by:

$$\begin{cases} I_{eff} = I - I_0 \\ U_{eff} = U - IR_{mot}^{ele} \end{cases} \tag{3.21}$$

, where I_0 is the idle current.

The motor power is the product of the effective current by the effective voltage, while the electric power will be the total current and voltage being supplied to the motor. Accordingly, the electric motor efficiency is given by:

$$\eta_{mot} = \frac{P_{mot}}{P_{ele}} = \frac{U_{eff}I_{eff}}{UI} \tag{3.22}$$

The shaft's rotational velocity (N) is given by:

$$N = k_v U_{eff} \tag{3.23}$$

, where (k_v) is defined as the motor's speed constant and it is one of the motor specifications.

A scheme with the propulsion model for the electric motor case is shown in Figure 3.2. The power setting (δ_{th}) and current (I) are initially guessed and after iterated, while the user has to know the idle voltage (U_{bat_0}) as well as the internal resistances (R_{bat}^{ele}) of the battery and the ESC (R_{esc}).

More often than not, the internal resistances of the battery and ESC are not provided by the manufacturers. It is thus worthwhile to discuss a way of estimating them. As for the battery internal resistance, when the battery is not charging (resting voltage) its voltage does not depend from its internal resistance. However, when an electric current is applied, the voltage falls proportionally to the battery's internal resistance. Let ($V_{unloaded}$) be the battery's unloaded voltage, (V_{loaded}) be the battery's loaded voltage, (I_{loaded}) be the battery's electric current, the battery internal resistance can be given by:

$$R_{bat}^{ele} = \frac{V_{unloaded} - V_{loaded}}{I_{load}} \tag{3.24}$$

It should be noted that the battery internal resistance is dependent on the battery temperature. In this way, the measurements should be made at a constant temperature. LiPo batteries tend to have very high internal resistances at temperature below $5C$.

As for the electronic speed control internal resistance, it is possible to compute an approximate internal resistance based upon the MOSFET's internal resistance. By knowing the ESC layout, it is possible to know which batteries are in series and which are in parallel. One can thus compute the equivalent electric resistance, which is will correspond to the electronic speed control internal resistance.

Finally, the required power (P_{req}), is the product of the airplane drag (D) by its velocity (V) for the flight condition under study, which comes from the flight mechanics analysis and vehicle mission.

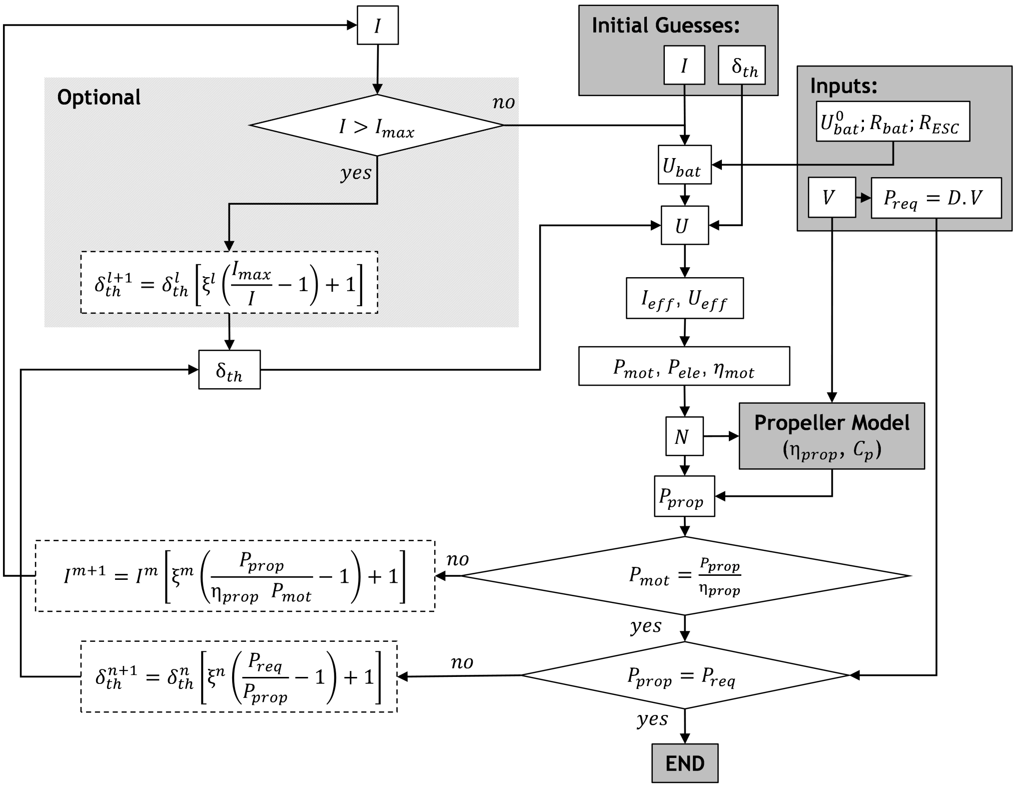

I
Initial Guesses:
I
δ_{th}
Inputs:
$U^0_{bat}; R_{bat}; R_{ESC}$
Optional
$I > I_{max}$
no
yes
U_{bat}
U
V
$P_{req} = D.V$
$\delta_{th}^{l+1} = \delta_{th}^{l}\left[\xi^{l}\left(\frac{I_{max}}{I} - 1\right) + 1\right]$
I_{eff}, U_{eff}
δ_{th}
$P_{mot}, P_{ele}, \eta_{mot}$
N
Propeller Model
(η_{prop}, C_p)
P_{prop}
$I^{m+1} = I^{m}\left[\xi^{m}\left(\frac{P_{prop}}{\eta_{prop}\ P_{mot}} - 1\right) + 1\right]$
no
$P_{mot} = \frac{P_{prop}}{\eta_{prop}}$
yes
$\delta_{th}^{n+1} = \delta_{th}^{n}\left[\xi^{n}\left(\frac{P_{req}}{P_{prop}} - 1\right) + 1\right]$
no
$P_{prop} = P_{req}$
yes
END

Figure 3.2: Propulsion model implemented for the electric motor.

Three iteration cycles have been built. One is optional and is only used if one wants to establish a maximum current and correct the power setting if this limit is exceeded. A second iteration cycle makes sure the electric current is corrected so that the shaft power (P_{mot}), equals the absorbed propeller power (P_{prop}/η_{prop}) which is a must since there is no slippage between the two. A last iteration corrects the power setting to ensure the available propulsive power (P_{prop}) equals the required power (P_{req}). In the diagram of Figure 3.2 (η_{mot}) refers to the motor efficiency and (η_{prop}) to the propulsive efficiency. The propeller model is described in Subsection 3.3.2.3.

In the diagram of Figure 3.2, the symbols (ξ^{l}), (ξ^{m}) and (ξ^{n}) refer to non-dimensional damping coefficients which are used in their respective iterative convergence procedures to make sure that it is always possible to have convergence. These require a careful tuning and testing before they can be considered a good choice for the problem and variable under consideration and they can assume values within the following interval: $\xi \in]0,1[\backslash\{0\}$.

It is important to clarify how the energy consumption will be estimated. The electrical power (P_{ele}) can be computed from the electrical current (I) and electrical voltage (U), in accordance with:

$$P_{mot} = UI \tag{3.25}$$

Using Equations (3.25) and (3.20), it is possible to obtain:

$$P_{mot} = U_{bat}\delta_{th}I \tag{3.26}$$

Finally, using Equations (3.26) and (3.19), it is possible to obtain:

$$P_{mot} = \left(U_{bat_0} - IR_{bat}^{ele} - IR_{esc}^{ele}\right)\delta_{th}I \tag{3.27}$$

The electrical power consumed by the motor, ESC and battery(ies) (P_{ele}) can be computed from the electric current (I) and electrical voltage (U), in accordance with:

$$P_{ele} = U_{bat_0}\delta_{th}I \tag{3.28}$$

As for the energy, it is the time integral of the power consumption between two instants t_1 and t_2.

$$E_{ele} = \int_{t_1}^{t_2} P_{ele}(t)dt \tag{3.29}$$

Once the time interval (Δt) at which the electrical power (P_{ele}) is constant is known, it is possible to estimate the respective electrical energy consumption using finite diferences:

$$E_{ele} = P_{ele}.\Delta t \tag{3.30}$$

Finally, it is possible to estimate the number of batteries required to successfully perform the mission under analysis, by dividing the total required energy (E_{ele}) available energy (E_{av}), which can be computed from:

$$E_{av} = 3600.(U_{cell}).C.n_{cseries} \tag{3.31}$$

, where C is the battery capacitance $n_{cseries}$ is the number of battery cells in series and U_{cell} is the unitary cell voltage, which equals $3.7V$ for a standard LiPo battery.

3.3.2.2 Combustion Engine

As for the combustion engine, the engine power (P_{eng}) can be computed from the engine combustion power (P_{comb}) and the engine's efficiency:

$$P_{eng} = P_{comb} \cdot \eta_{eng} \tag{3.32}$$

The propeller shaft power (P_{shaft}) depends generically on the engine power (P_{eng}) and on the systems power (P_{sys}), in which case engine power would be used by the aircraft systems by means of an alternator:

$$P_{shaft} = P_{eng} - P_{sys} \tag{3.33}$$

From the propeller efficiency model, by knowing the propeller pitch (p), diameter (d) and operating conditions (velocity (V) and rotational velocity (N)), it is possible to estimate both the propeller efficiency (η_{prop}) and propeller power (P_{prop}). It must then be assured that Equation (3.34) is respected.

$$P_{shaft} = \frac{P_{prop}}{\eta_{prop}} \tag{3.34}$$

This can be done by adjusting the power setting (δ_{th}), as depicted in Figure 3.3.

Finally, it must be assured that the propeller power (P_{prop}) equals the required power (P_{req}), which is made by varying the propeller rotational velocity (N). These two iterative cycles and how they relate with the most relevant design variables are shown in Figure 3.3.

In the diagram of Figure 3.3, the symbols (ξ^l), (ξ^m) and (ξ^n) refer to non-dimensional damping coefficients. These have been defined in Subsection 3.3.2.1.

The specific fuel consumption (SFC) is generically known to be a function of both the engine power setting (δ_{th}) and engine angular velocity (N). However, it has been shown that the primary parameter influencing the SFC is the power setting. A qualitative representation of a typical curve of SFC vs. power setting is shown in Figure 3.4.

Thus, the SFC can be defined as:

$$SFC = \frac{SFC_0}{{\delta_{th}}^k} \tag{3.35}$$

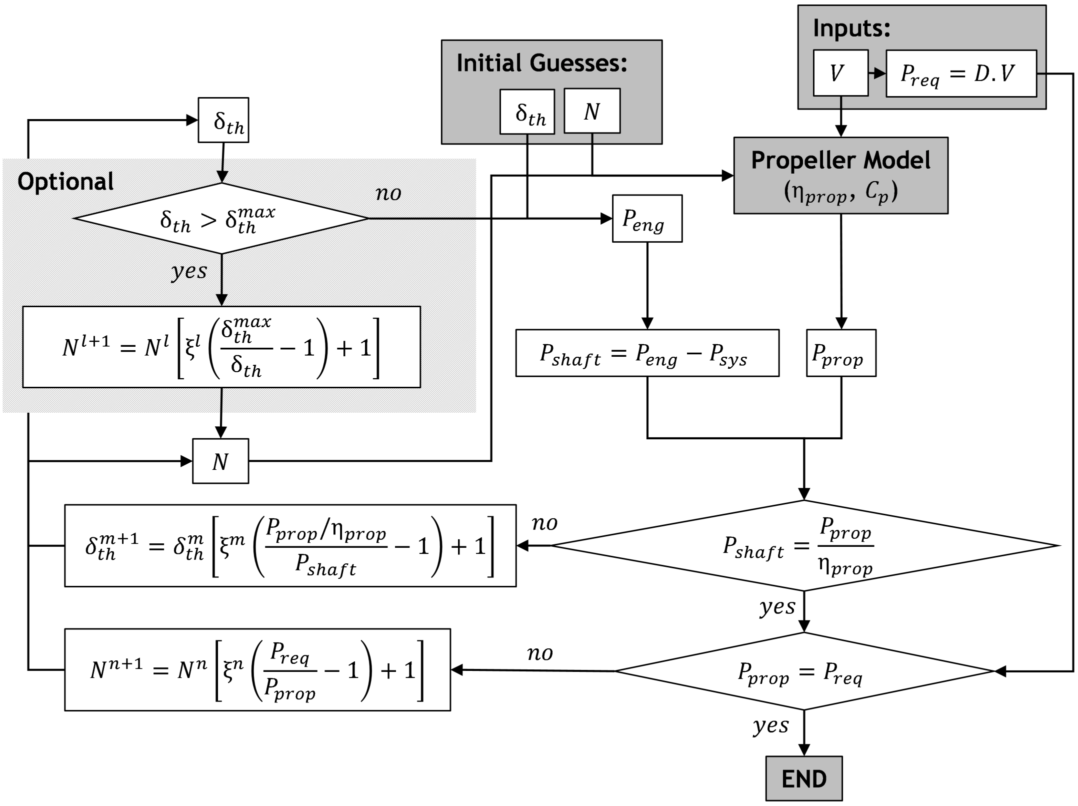

Inputs:
V
$P_{req} = D.V$
Initial Guesses:
δ_{th}
N
Propeller Model
(η_{prop}, C_p)
δ_{th}
Optional
$\delta_{th} > \delta_{th}^{max}$
no
yes
P_{eng}
$N^{l+1} = N^l \left[\xi^l \left(\frac{\delta_{th}^{max}}{\delta_{th}} - 1\right) + 1\right]$
$P_{shaft} = P_{eng} - P_{sys}$
P_{prop}
N
$\delta_{th}^{m+1} = \delta_{th}^m \left[\xi^m \left(\frac{P_{prop}/\eta_{prop}}{P_{shaft}} - 1\right) + 1\right]$
no
$P_{shaft} = \frac{P_{prop}}{\eta_{prop}}$
yes
$N^{n+1} = N^n \left[\xi^n \left(\frac{P_{req}}{P_{prop}} - 1\right) + 1\right]$
no
$P_{prop} = P_{req}$
yes
END

Figure 3.3: Propulsion model implemented for the combustion engine.

, where SFC_0 is the SFC for maximum power setting $\delta_{th} = 1$ and k is a generic constant, which depends on the engine under analysis, Prouty (2007) [95].

Once the shaft power is known, it is possible to estimate the required fuel mass (m_{fuel}), from the SFC, engine power (P_{eng}) and time interval (Δt), in accordance with:

$$m_{fuel} = \int_{t_1}^{t_2} SFC(t).P_{eng}(t)dt \tag{3.36}$$

, which, after discretized in the time domain becomes:

$$m_{fuel} = SFC(t).P_{eng}.\Delta t \tag{3.37}$$

Finally, it is possible to estimate the fuel tank volume required for the mission under analysis, which can be computed via Equation (3.38), by knowing the fuel mass (m_{fuel}) and respective density (ρ_{fuel}).

$$V_{fuel} = \frac{m_{fuel}}{\rho_{fuel}} \tag{3.38}$$

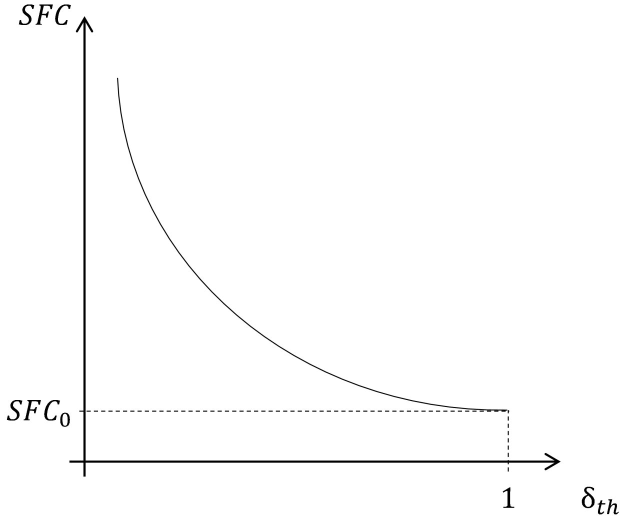


Figure 3.4: Engine SFC as a function of the power setting.

3.3.2.3 Propeller Model

For the eletric motor and combustion engine propulsion models depicted in Figures 3.2 and 3.3, respectively, it is required to estimate the propeller efficiency (η_{prop}) and power coefficient (C_p). These are functions of the propeller advance ratio (J), which is a non-dimensional parameter that depends on the linear velocity of the propeller with respect to the flow-field (V), its rotational velocity in revolutions per second ($N/60$) and its diameter (d). The propeller advance ratio is defined in:

$$J = \frac{60.V}{Nd} \tag{3.39}$$

By knowing how the power coefficient (C_p) evolves with the advance ratio (J), it is possible to compute the propeller absorbed power or shaft power (P_{shaft}), as:

$$P_{shaft} = C_p \rho \left(\frac{N}{60}\right)^3 d^5 \tag{3.40}$$

Several different softwares can be used for computing the power coefficient (C_p) and propulsive efficiency (η_p) curves against the propeller advance ratio (J), e.g., by increasing degree of complexity and output accuracy, possible propeller analysis programs include the PropSelector [96], JavaProp [97] and QPROP [98]. The first is relatively straightforward to use - making use of experimental propeller data - and does not need many input parameters while the latter has got a relatively sophisticated and accurate aerodynamic model - with an advanced Blade-Element/Vortex Method - thus requiring more inputs including detailed information regarding the propeller shape. However, it is also possible to use experimental wind-tunnel measurements.

The remaining of this section describes the different approaches that have been implemented for the sake of estimating the propeller's power coefficient (C_p) and the propeller's propulsive efficiency (η_p). Generically, the propeller performance model can be tailored to consider a single or a generic propeller. By single propeller it should be understood that both the propeller diameter (d) and the propeller pitch (p) are user-defined parameters, whereas by generic propeller it should be understood that these quantities (d,p) can be optimization variables. Accordingly, three different approaches have been developed:

- For single propeller, when the propeller diameter and pitch are constant **parameters**:

  Linear Interpolation 3.3.2.3;

  Polynomial Approximation (Least-square approximation) 3.3.2.3.

- For generic propeller, when the propeller diameter and pitch are **optimization variables**:

  Generic Polynomial Approximation 3.3.2.3.

Linear Interpolation

The linear interpolation is a widely used approximation provided there is a short distance between consecutive known points and/or the function that aims to be approximated exhibits a linear or approximately linear behavior. It is obvious that this last requirement is not met in both the power coefficient (C_p) versus the propeller advance ratio (J) and propulsive efficiency (η_p) versus the propeller advance ratio (J) curves. However the number of data points known shall be high enough for the linear interpolation not to put the results at risk.

Polynomial Approximation (Least-square approximation)

An alternative to the linear interpolation is creating a polynomial which can better mimic the actual behavior of the power coefficient (C_p) and propeller efficiency (η_{prop}) curves as a function of the advance ratio (J). The least-square method has been used with such goal. Accordingly, a number of points (n) of (C_p) and (η_{prop}) coefficients that correspond to different values of (J) shall be known upfront. Then, it is possible to define the polynomial degree (m). The purpose is thus to find two polynomials of the form:

$$y = \bar{A}_1 + \bar{A}_2 x + \bar{A}_3 x^2 + ... + \bar{A}_{m+1} x^m \tag{3.41}$$

, where, (x) represents the independent variable, (y) represents the dependent variable and ($\bar{A}_i$) represent the polynomial coefficients.

The least-square method can be summarized in the generic Equation (3.42), where the single unknown is column vector ($[A]$)with its ($\bar{A}_i$) coefficients, as per Equation (3.41).

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} {x_i}^2 & \cdots & \sum_{i=1}^{n} {x_i}^m \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} {x_i}^2 & \sum_{i=1}^{n} {x_i}^3 & \cdots & \sum_{i=1}^{n} {x_i}^{m+1} \\ \sum_{i=1}^{n} {x_i}^2 & \sum_{i=1}^{n} {x_i}^3 & \sum_{i=1}^{n} {x_i}^4 & \cdots & \sum_{i=1}^{n} {x_i}^{m+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} {x_i}^m & \sum_{i=1}^{n} {x_i}^{m+1} & \sum_{i=1}^{n} {x_i}^{m+2} & \cdots & \sum_{i=1}^{n} {x_i}^{2m} \end{bmatrix} \begin{bmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \bar{A}_3 \\ \vdots \\ \bar{A}_{m+1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} {x_i}^2 y_i \\ \vdots \\ \sum_{i=1}^{n} {x_i}^m y_i \end{bmatrix} \tag{3.42}$$

In the problem in hands, the dependent variables are either (C_p) or (η_{prop}) whereas the independent variable is the propeller advance ratio (J). Accordingly, and defining the polynomial degree (m_1) for the (C_p) polynomial and (m_2) for the (η_{prop}) polynomial, it comes (3.43) and (3.44).

$$C_p = \bar{A}_1 + \bar{A}_2 J + \bar{A}_3 J^2 + ... + \bar{A}_{m_1+1} J^{m_1} \tag{3.43}$$

$$\eta_{prop} = \bar{B}_1 + \bar{B}_2 J + \bar{B}_3 J^2 + ... + \bar{B}_{m_2+1} J^{m_2} \tag{3.44}$$

The above discussion regarding the application of the least-square method for estimating the (C_p) and (η_{prop}) coefficients refers to a specific value of propeller revolutions per minute (RPM). If the user has data points for different values of RPM, a least-square curve is generated for each RPM and the final coefficients will be a weighted average of the values obtained for two different RPMs, one immediately below and other immediately above the actual RPM. If it happens than the RPM is below the minimum or above the maximum, their respective values are either extrapolated or assumed to be equal to the minimum or maximum RPM known, respectively.

Generic Polynomial Approximation

When the propeller pitch and diameter are not parameters but rather optimization variables, a method different than the previous two is required. The approach presented in the following paragraphs makes it possible to estimate the power coefficient (C_p) and propeller efficiency (η_{prop}) for an arbitrary propeller, within a predefined range of diameters and pitches, notwithstanding a loss of accuracy with respect to the two formerly presented methods.

Figures 3.5 and 3.6 depict typical curves for the power coefficient (C_p) and propeller efficiency (η_{prop}) with the propeller advance ratio (J), respectively.

The implemented approach assumes that the (C_p) and (η_{prop}) curves are similar for the range of propeller diameters and pitches considered. Accordingly, the generic polynomials $(C_p = C_p(d,p,J))$ and $(\eta_p = \eta_p(d,p,J))$, are actually obtained by knowing each propeller's maximum advance ratio (J_{max}), power coefficient for null advance ratio (C_{p_0}) and maximum efficiency $(\eta_{p_{max}})$.

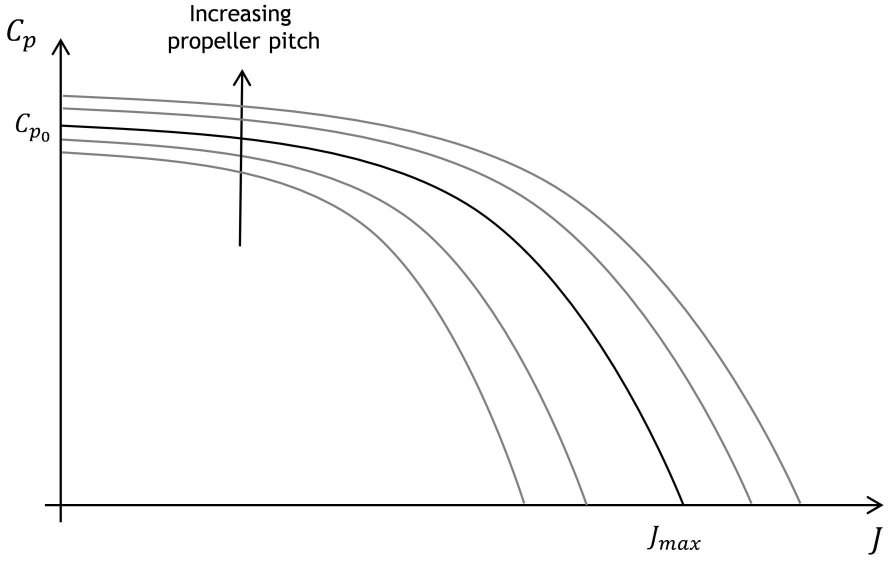


Figure 3.5: Typical propeller power coefficient (C_p) curves as a function of the advance ratio (J).

The first step towards this implementation is thus to estimate the characteristic values of (J_{max}), (C_{p_0}) and ($\eta_{p_{max}}$) for a large number of different sized propellers, as a function of the propeller diameter (d) and pitch (p). As already discussed these can be done in a number of ways using different off the shelf software. Once this data is collected, the least-square method can be used to estimate the coefficients ($\bar{C}_i$), ($\bar{D}_i$) and ($\bar{E}_i$) of the polynomial representations, shown in Equations (3.45), (3.46) and (3.47):

$$J_{max} = \bar{C}_1 d + \bar{C}_2 p + \bar{C}_3 d^2 + \bar{C}_4 dp + \bar{C}_5 p^2 + \bar{C}_6 d^3 + \bar{C}_7 d^2 p + \bar{C}_8 p^2 d + \bar{C}_9 p^3 \tag{3.45}$$

$$C_{p_0} = \bar{D}_1 d + \bar{D}_2 p + \bar{D}_3 d^2 + \bar{D}_4 dp + \bar{D}_5 p^2 + \bar{D}_6 d^3 + \bar{D}_7 d^2 p + \bar{D}_8 p^2 d + \bar{D}_9 p^3 \tag{3.46}$$

$$\eta_{p_{max}} = \bar{E}_1 d + \bar{E}_2 p + \bar{E}_3 d^2 + \bar{E}_4 dp + \bar{E}_5 p^2 + \bar{E}_6 d^3 + \bar{E}_7 d^2 p + \bar{E}_8 p^2 d + \bar{E}_9 p^3 \tag{3.47}$$

Once the coefficients of Equation (3.45), Equation (3.46) and Equation (3.47) have been estimated, the final goal is to estimate the power coefficient (C_p) and propulsive efficiency curves (η_{prop}). The non-dimensional representations below are used for all propellers where the coefficients ($\bar{F}$) and ($\bar{G}$) were assumed for typical curve shapes of (C_p) and (η_{prop}), respectively.

$$\frac{C_p}{C_{p_0}} = \bar{F}_0 + \sum_{i=1}^{4} \bar{F}_i \left(\frac{J}{J_{max}}\right)^i \tag{3.48}$$

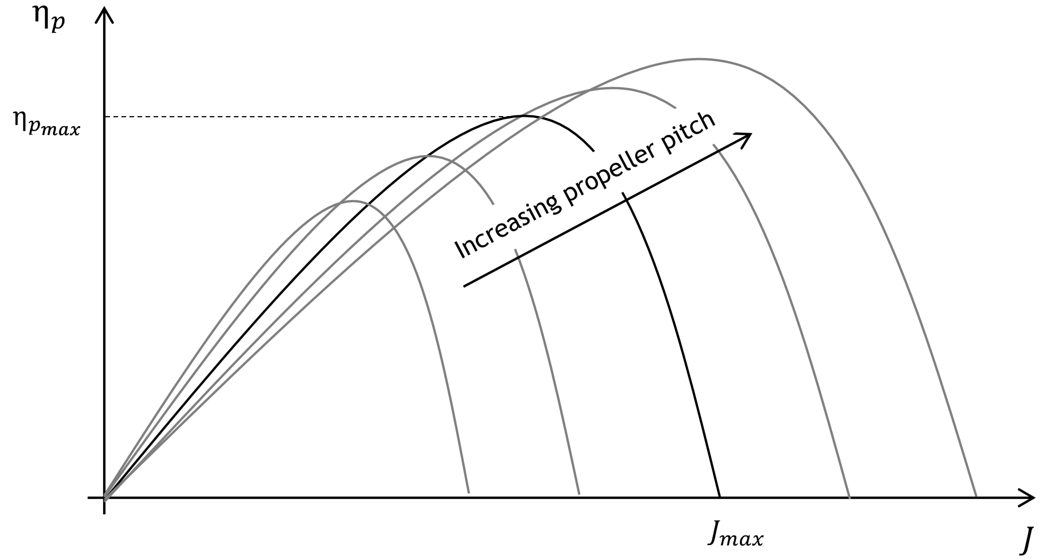


Figure 3.6: Typical propeller efficiency (η_p) curves as a function of the advance ratio (J).

$$\frac{\eta_p}{\eta_{p_{max}}} = \bar{G}_0 + \sum_{i=1}^{6} \bar{G}_i \left(\frac{J}{J_{max}}\right)^i \tag{3.49}$$

3.3.3 Weight

The design take-off weight (DTOW) is the sum of the aircraft structural weight (W_{str}), the systems weight (W_{sys}), the energy weight (W_{ene}) and payload weight (W_{pay}), where: structure refers to the structural components, like the wing, tail, fuselage and landing gears; systems refers to all devices required for flight that are not structural, such as the motor, propeller, ESC, cables, servomechanisms and the receiver; energy refers to the weight of the power source used, either fuel, in the case of a combustion engine or a battery in the case of the electric motor; and payload refers to all payload which is not essential for flight:

$$DTOW = W_{str} + W_{sys} + W_{ene} + W_{pay} \tag{3.50}$$

The type of propulsion, payload weight and systems weight, together with the intended mission profile are user inputs. Conversely, the structural weight will also depend on the energy weight which in turn will depend on the mission profile and has to be duly estimated. For a typical mission profile with a cruise/loiter phase the energy weight is iterated until the distance/time required is attained.

3.3.3.1 Structural Weight

Having a reliable structural weight estimate is of utmost relevance for the quality of the optimization results. As such, an empirical approach for estimating the structural weight of the main components - the wing, the horizontal stabilizer, the vertical stabilizer, the fuselage, the main landing gear (MLG) and the nose landing gear (NLG) - has been adopted. This method, originally presented by Gamboa et al (2009) [99] and further extended in Albuquerque et al (2015) [100] makes use of Raymer's [1] structural weight correlations and relates the actual structure weight with an already built similar structure. The fidelity of this approach is, of course, primarily based on the quality of Raymer's correlations, however, not least important is choosing a reference structure built with the same type of materials and manufacturing techniques, so that the actual and reference structures are actually comparable.

In order to exemplify the approach, let us consider the case of the wing, and let (W_w) be the wing weight, (W_w^{ref}) be the reference wing weight and (ΔW_w) the difference between the wing and the reference wing weights:

$$\Delta W_w = W_w - {W_w}^{ref} \tag{3.51}$$

By dividing Equation (3.51) by $({W_w}^{ref})$, it is possible to obtain:

$$\Delta W_w = {W_w}^{ref}\left(\frac{W_w}{{W_w}^{ref}} - 1\right) \tag{3.52}$$

By defining $(\bar{K}_w = {W_w}^{ref}/{W_{str}}^{ref})$, Equation (3.52) becomes:

$$\Delta W_w = \bar{K}_w {W_{str}}^{ref}\left(\frac{W_w}{{W_w}^{ref}} - 1\right) \tag{3.53}$$

From Equation (3.53), and since both $(\bar{K}_w)$ and $({W_{str}}^{ref})$ are known, what is left is a way to relate the wing weight with the reference wing weight, which can be done using Raymer's correlations. The same approach used for the wing is likewise valid for the remaining components.

Accordingly, the structural weight of the aircraft under analysis is the sum of the reference aircraft structural weight and the main structural components weight variations, as:

$$W_{str} = {W_{str}}^{ref} + \Delta W_w + \Delta W_{ht} + \Delta W_{vt} + \Delta W_{fus} + \Delta W_{mlg} + \Delta W_{nlg} \tag{3.54}$$

Raymer's correlations allow for variations in the lifting surfaces airfoil's relative thickness (t/c), planform areas (S_w, S_{ht}, S_{vt}), aspect ratio $(\Lambda_w, \Lambda_{ht}, \Lambda_{vt})$ and taper ratio $(\lambda_w, \lambda_{ht}, \lambda_{vt})$ as well as on the fuselage length (l_{fus}), diameter (d_{fus}), the MLG length (L_m), the NLG length (L_n), landing gear ultimate load factor (N_l) and landing gear design gross weight (W_l).

The weight increments of each main structural component can be estimated as follows:

$$\begin{aligned}\Delta W_w =& \bar{K}_w {W_{str}}^{ref} \left[\left(\frac{S_w}{S_w^{ref}} \right)^{0.758} \left(\frac{\Lambda_w}{{\Lambda_w}^{ref}} \right)^{0.6} \left(\frac{\lambda_w}{{\lambda_w}^{ref}} \right)^{0.04} \right. \\ & \left. \left(\frac{t/c}{t/c^{ref}} \right)^{-0.3} \left(\frac{nW}{n^{ref} W^{ref}} \right)^{0.49} - 1 \right]\end{aligned} \tag{3.55}$$

$$\Delta W_{ht} = \bar{K}_{ht} W_{str}^{ref} \left[\left(\frac{S_{ht}}{S_{ht}^{ref}} \right)^{1.344} \left(\frac{\Lambda_{ht}}{\Lambda_{ht}^{ref}} \right)^{-0.448} \left(\frac{nW}{n^{ref} W^{ref}} \right)^{0.414} - 1 \right] \tag{3.56}$$

$$\Delta W_{vt} = \bar{K}_{vt} {W_{str}}^{ref} \left[\left(\frac{S_{vt}}{S_{vt}^{ref}} \right)^{1.31} \left(\frac{\Lambda_{ht}}{{\Lambda_{ht}}^{ref}} \right)^{0.437} \left(\frac{nW}{n^{ref} W^{ref}} \right)^{0.376} - 1 \right] \tag{3.57}$$

$$\begin{aligned}\Delta W_{fus} =& \bar{K}_{fus} {W_{str}}^{ref} \left[\left(\frac{S_{wet}}{S_{wet}^{ref}} \right)^{1.086} \left(\frac{L}{L^{ref}} \right)^{-0.051} \right. \\ & \left. \left(\frac{l_{fus}/d_{fus}}{l_{fus}^{ref} d_{fus}^{ref}} \right)^{-0.072} \left(\frac{nW}{n^{ref} W^{ref}} \right)^{0.177} - 1 \right]\end{aligned} \tag{3.58}$$

$$\Delta W_{mlg} = \bar{K}_{mlg} {W_{str}}^{ref} \left[\left(\frac{N_l W_l}{{N_l}^{ref} {W_l}^{ref}} \right)^{0.768} \left(\frac{L_m}{{L_m}^{ref}} \right)^{0.409} \right] \tag{3.59}$$

$$\Delta W_{nlg} = \bar{K}_{nlg} {W_{str}}^{ref} \left[\left(\frac{N_l W_l}{{N_l}^{ref} {W_l}^{ref}} \right)^{0.566} \left(\frac{L_n}{{L_n}^{ref}} \right)^{0.845} \right] \tag{3.60}$$

where ($\bar{K}_w$), ($\bar{K}_{ht}$), ($\bar{K}_{vt}$), ($\bar{K}_{fus}$), ($\bar{K}_{mlg}$), and ($\bar{K}_{nlg}$) are the ratios of the reference component weight to the reference vehicle's structural weight, for the wing, horizontal tail, vertical tail, fuselage, MLG and NLG, respectively.

As it can be seen from Equations (3.55)-(3.60), not only does the structural weight estimate depend on the aircraft geometry, but also on its design take-off weight. Since these quantities evolve with the number of iterations, this structural weight estimate is iterative and regardless of the optimization methodology in use, the final result is not obtained before there is an actual weight estimate convergence.

The wing weight estimates presented in Equation (3.55) refer to a fixed-wing. As the assessment of morphing technologies is one of the goals of this thesis, it is important to address the structural weight changes due to a variable span wing (VSW), and to correct the wing structural weight estimate accordingly. Cunha (2014) [101] has developed a parametric study in the ANSYS software to find a correlation for estimating the weight of a variable span wing which used a telescopic wing-box with a given chord (c). It has been shown that the wing weight varies with the wingspan (b) and span extension factor ($\bar{p}$), which is defined by ($\bar{p} = (b_{max} - b_{min})/b_{max}$), according to:

$$\begin{aligned} W(\bar{p},b) =& 1.4522724 + 9.666774\bar{p} - 0.001604916b + \\ & 0.0003566916\bar{p}^2 + 0.859356b^2 + 1.718712\bar{p}b \end{aligned} \tag{3.61}$$

From the fixed-wing weight (W_w), it is straightforward to estimate the weight of the corresponding VSW, using:

$$W_w{}^{VSW} = W_w \frac{W(\bar{p},b)}{W(0,b)} \tag{3.62}$$

3.3.3.2 Energy Weight

The two computational codes presented in Chapters 3 and 4 enable the user to choose either a combustion engine or an electric motor. Depending on the user defined settings, the energy weight can either be an input or an output in both codes. If it is not a user input, the energy weight is computed from:

$$W_{ene} = \frac{E}{e_{spec}}.g \tag{3.63}$$

, where E is the energy consumption, g is the acceleration due to gravity and e_{spec} is either the specific energy, which can be computed from Equation (3.64) and (3.65) for depending on the usage of batteries or fuel, respectively.

$$e_{spec} = \frac{E}{m_{bat}} \tag{3.64}$$

$$e_{spec} = \frac{E}{SFC.P.t} \tag{3.65}$$

3.3.4 Static Stability

One of the goals of the MDO methodologies developed for this thesis is that the optimized configuration meets the user requirements in terms of longitudinal and lateral static stability margins. The first can be met via an appropriate sizing of the horizontal tail (S_{ht}), whereas the latter relies on sizing the vertical tail (S_{vt}) and the wing dihedral angle (Γ). Hence, every time a new wing layout configuration is being considered, the codes run a static stability subroutine for each different mission stage, selecting the most critical requirement for each of these three variables, in such a way that the following iteration takes into account these critical values for the stabilizers' planform areas and wing dihedral angle. Hence, the values of these three variables will be updated at each system level iteration.

3.3.4.1 Longitudinal Static Stability

As it is known, the pressure distribution about any lifting surface can be represented by a lift force (L), a drag force (D) and a pitching moment (M). For the pitching moment equilibrium about the aircraft CG, it is common to neglect the horizontal tail pitching moment (M_{ht}), since it is commonly one order of magnitude lower than the wing pitching moment and the lift forces' moments. It is also common practice to disregard the horizontal tail pitching moment (M_{ht}) and the pitching moment due to the propulsion (C_{m_p}) generated by the fact that their vertical offset with respect to the CG. Not only are the aerodynamic resistances significantly lower than the respective lift forces, but also the moment arm - in this case, vertical distance from the place where the force acts to the CG - is usually significantly smaller.

From Figure 3.1, and taking into consideration the aforementioned assumptions, it is possible to write the equation with the pitching moment equilibrium around the CG:

$$L_{ht}l_{ht} = L_w(\bar{h} - h_{n_{wb}})\bar{c} - M_w \tag{3.66}$$

As a result, the horizontal tail lift coefficient ($C_{L_{ht}}$) required for balancing the remaining aerodynamic forces can be determined. Firstly, by replacing the forces and moments of Equation (3.66) by the respective coefficients, dynamic pressure and reference areas and assuming that the velocity vector's magnitude is the same for the wing and for the horizontal stabilizer - which is an acceptable assumption given that the expected wing down-wash angle is low and since this angle has got little impact on the velocity vector magnitude - it is possible to obtain:

$$C_{L_{ht}}\frac{1}{2}\rho S_{ht}V^2 l_{ht} = C_{L_w}\frac{1}{2}\rho S V^2(\bar{h} - h_{n_{wb}})\bar{c} - C_{M_w}\frac{1}{2}\rho S V^2\bar{c} \tag{3.67}$$

Finally, after some simplifications and solving Equation (3.67) for the horizontal tail lift coefficient ($C_{L_{ht}}$), it is possible to obtain:

$$C_{L_{ht}} = \frac{[C_{L_w}(\bar{h} - h_{n_{wb}}) - C_{M_w}]S\bar{c}}{S_{ht}l_{ht}} \tag{3.68}$$

One can define the horizontal tail volume ratios ($\bar{v}_{ht}$ and v_{ht}), the first using the distance between the aerodynamic centers of the wing and the horizontal tail ($\bar{l}_{ht}$) and the second using the distance between the CG and the horizontal tail aerodynamic center (l_{ht}):

$$\begin{cases} \bar{v}_{ht} = \frac{\bar{l}_{ht}S_{ht}}{\bar{c}S} \\ v_{ht} = \frac{l_{ht}S_{ht}}{\bar{c}S} = \bar{v}_{ht} - \frac{S_{ht}}{S}(\bar{h} - h_{n_{wb}}) \end{cases} \tag{3.69}$$

At this point, it is possible to know the partial pitching moment contributions of each of the two lifting surfaces under consideration ($C_{m_{wb}}$ and $C_{m_{ht}}$):

$$\begin{cases} C_{m_{wb}} = C_{m_{ac_{wb}}} + C_{L_{wb}}(\bar{h} - h_{n_{wb}}) \\ C_{m_{ht}} = -\bar{v}_{ht}C_{L_{ht}} + C_{L_{ht}}\frac{S_{ht}}{S}(\bar{h} - h_{n_{wb}}) \end{cases} \tag{3.70}$$

, so that the total pitching moment around the aircraft's gravity center (C_m) is given by:

$$C_m = C_{m_{ac_{wb}}} + C_{L_{wb}}(\bar{h} - h_{n_{wb}}) - \bar{v}_{ht}C_{L_{ht}} + C_{L_{ht}}\frac{S_{ht}}{S}(\bar{h} - h_{n_{wb}}) \tag{3.71}$$

After some manipulation, it is possible to obtain:

$$C_m = C_{m_{ac_{wb}}} + C_L(\bar{h} - h_{n_{wb}}) - \bar{v}_{ht}C_{L_{ht}} \tag{3.72}$$

The CG rearmost position depends on the static margin (K_n) which is defined as the distance between the aircraft aerodynamic center (AC) and neutral point (NP):

$$K_n = (h_n - h) \tag{3.73}$$

To avoid any confusion between the terms AC and NP, the former will be used to refer to the wing and horizontal stabilizer chord-wise position where the pitching moment is independent of the angle-of-attack (AOA), whilst the latter will be used to refer to the aircraft lengthwise point where its overall pitching moment is independent of the pitching moment coefficient.

Conversely, the CG forward-most position (h_{limit}) depends on the maximum wing pitching moment that the horizontal tail is able to counteract. This point can be found by making ($C_m = 0$) and ($C_L = C_{L_{max}}$) on Equation (3.72) and ($C_{L_{ht}}$) computed as per Equation (3.68). After some manipulation, the CG forward-most position leads to:

$$h_{limit} = h_{n_{wb}} - \frac{1}{C_{L_{max}}}(C_{m_{ac_{wb}}} - \bar{v}_{ht}C_{L_{ht}}) \tag{3.74}$$

From the aerodynamics viewpoint, the relevant angles are represented in Figure 3.7. These include the aircraft AOA (α), the wing AOA and incidence (α_w and i_w) and the horizontal tail AOA and incidence (α_{ht} and i_{ht}), the wing down-wash angle (ε) and the elevator deflection angle (δ_e).

From Figure 3.7, it is possible to derive the relation between the different angles:

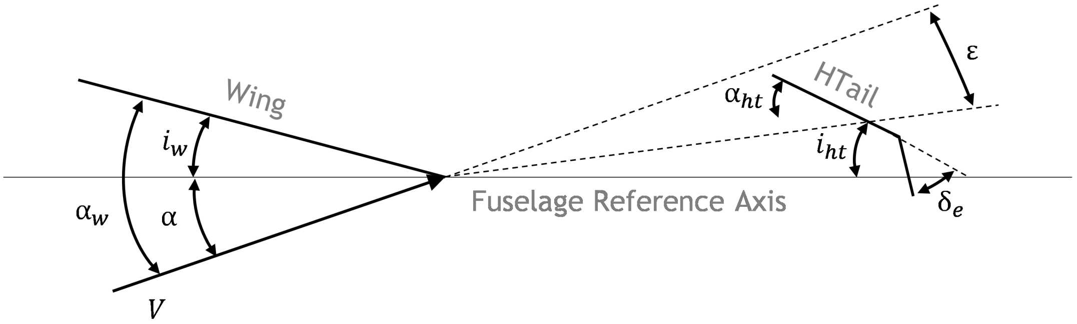


Figure 3.7: Diagram of forces

$$\begin{cases} \alpha_w = \alpha + i_w \\ \alpha_{ht} = \alpha + i_{ht} - \varepsilon \end{cases} \tag{3.75}$$

The lifting surfaces' lift coefficients (C_{L_w} and $C_{L_{ht}}$) can be estimated via Equations (3.76). The wing lift coefficient at zero angle-of-attack ($C_{L_{0_w}}$) and lift curve slope ($C_{L_{\alpha_w}}$) and on its AOA (α_w), neglecting the contribution of high-lift devices. Likewise, the horizontal tail lift coefficient depends on its zero lift coefficient ($C_{L_{0_{ht}}}$) and lift slope ($C_{L_{\alpha_{ht}}}$), AOA (α_{ht}) and elevator deflection (δ_e).

$$\begin{cases} C_{L_w} = C_{L_{0_w}} + C_{L_{\alpha_w}} \alpha_w \\ C_{L_{ht}} = C_{L_{0_{ht}}} + C_{L_{\alpha_{ht}}} \alpha_{ht} + C_{L_\delta} \delta_e \end{cases} \tag{3.76}$$

The nomenclature and sign convention used for the stability analysis is the one adopted by Etkin (1996) [102], as per Figure 3.8.

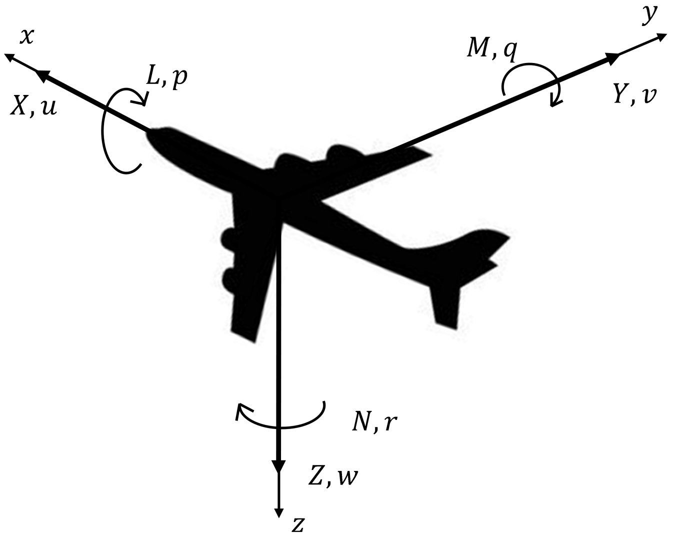


Figure 3.8: Notation of aircraft axes.

The CG position and the required static margin will be a user input, with the horizontal tail sizing using the user specified distance between the aerodynamic centers of the wing and of the horizontal tail to make sure that the aircraft's NP will stand in the right place.

$$K_n = h_n - \bar{h} \tag{3.77}$$

3.3.4.2 Lateral and Directional Static Stability

The lateral static stability requirements must make sure that, if the sideslip (β) with respect to both the rolling and yawing movements is not null, the airplane will naturally tend to counteract such tendencies returning to its original undisturbed position.

Application of the static stability principle to rotation about the z-axis suggests that the airplane should have weathercock stability. That is, when the airplane is at an angle of sideslip β relative to its flight path, its yawing moment should tend to restore it to symmetric flight. The yawing moment N is considered positive as shown in Figure 3.9. Hence the derivative $\partial N/\partial\beta$ must be positive.

The non-dimensional N coefficient is given by:

$$C_n = \frac{N}{\frac{1}{2}\rho V^2 S b} \tag{3.78}$$

Accordingly, for yawing stiffness, the yawing moment coefficient derivative with respect to the sideslip angle must respect the inequality ($C_{n_\beta} > 0$). According to Raymer (1989) [1] and Torenbeek (1982) [103], the common range of values for this parameter is the following (3.79).

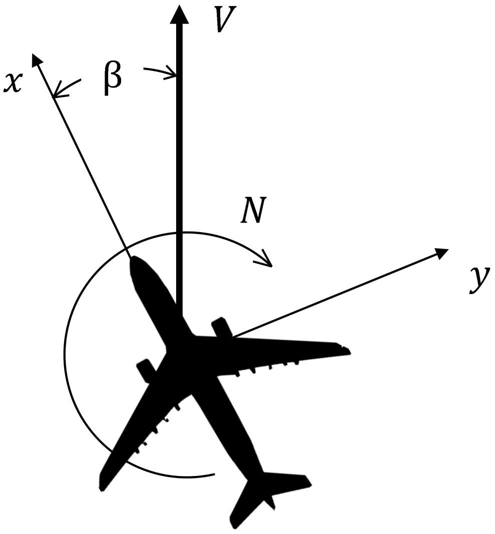


Figure 3.9: Sideslip angle and yawing moment.

$$0.04 < C_{n_\beta} < 0.1 \tag{3.79}$$

According to Etkin (1996) [102], the C_{n_β} can be estimated from:

$$C_{n_\beta} = \frac{1}{57.3}\left[\frac{1}{4\pi\Lambda} - \frac{tan(\bar{\Lambda}_{c/4})}{\pi\Lambda(\Lambda + 4cos(\bar{\Lambda}_{c/4}))}\left(cos(\bar{\Lambda}_{c/4}) - \frac{\Lambda}{2} - \frac{\Lambda^2}{8cos(\bar{\Lambda}_{c/4})} + 6(h_{n_w} - h)\frac{sin(\bar{\Lambda}_{c/4})}{\Lambda}\right)\right] \tag{3.80}$$

$$(per\ deg) \tag{3.81}$$

, where the different terms can be obtained from the USAF Datcom (1979) [104]. To make sure that the weathercock stability is achieved, the vertical stabilizer area may be adjusted. This is due to the fact that the vertical stabilizer area is the most relevant parameter in defining the aircraft C_{n_β}.

Application of the static stability principle to rotation about the airplane x-axis should assure rolling stability. That is, when the airplane is at an angle of bank ϕ relative to its flight path, its rolling moment should tend to restore it to symmetric flight. The rolling moment L is considered positive as shown in Figure 3.10. Hence the derivative $\partial L/\partial\beta$ must be negative.

The non-dimensional L coefficient is given by:

$$C_l = \frac{L}{\frac{1}{2}\rho V^2 Sb} \tag{3.82}$$

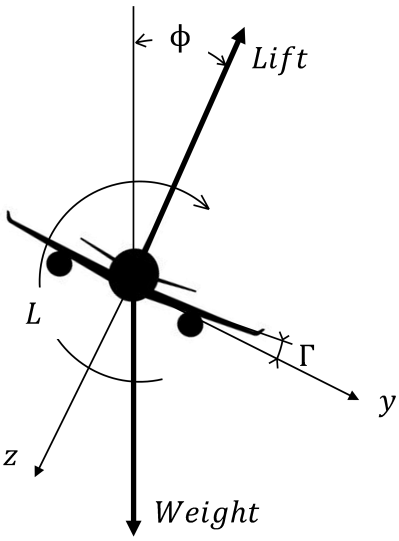


Figure 3.10: Dihedral angle and rolling moment.

In line with the sign and axis conventions adopted by Etkin (1996) [102], it must be assured that, for the rolling maneuver - the rolling moment coefficient derivative with respect to the sideslip angle respects the following inequality ($C_{l_\beta} < 0$). According to Raymer (1989) [1], the common range of values for this parameter is:

$$-0.05 < C_{l_\beta} < -0.02 \tag{3.83}$$

According to Etkin (1996) [102], the C_{l_β} can be estimated from:

$$C_{l_\beta} = C_L \left[\left(\frac{C_{l_\beta}}{C_L} \right)_{\bar{\Lambda}_{c/2}} K_{M_{\bar{\Lambda}}} + \left(\frac{C_{l_\beta}}{C_L} \right) \right]_{\Lambda} + \Gamma \left(\frac{C_{l_\beta}}{\Gamma} K_{M_\Gamma} \right) + \theta tan(\bar{\Lambda}_{c/4}) \frac{\Delta C_{l_\beta}}{\theta tan(\bar{\Lambda}_{c/4})} (per\ deg) \tag{3.84}$$

, where the different terms can be obtained from the USAF Datcom (1979) [104]. To make sure the rolling static stability margin is attained, some wing dihedral might be required. This is because the wing dihedral is the most relevant parameter in defining the aircraft C_{l_β}.

So as to make sure that the longitudinal static margin ($K_n > 0$) and the lateral static margins ($C_{l_\beta} < 0$) and ($C_{n_\beta} > 0$) result in a statically stable aircraft configuration, the iterative procedure depicted in Figure 3.11 has been implemented to determine the horizontal tail area (S_{ht}), vertical tail area (S_{vt}) and wing dihedral angle (Γ).

3.3.5 Dynamic Stability

In what regards to the dynamic stability, no optimization procedure has been implemented. However, the longitudinal and lateral dynamic stability matrices are estimated. In

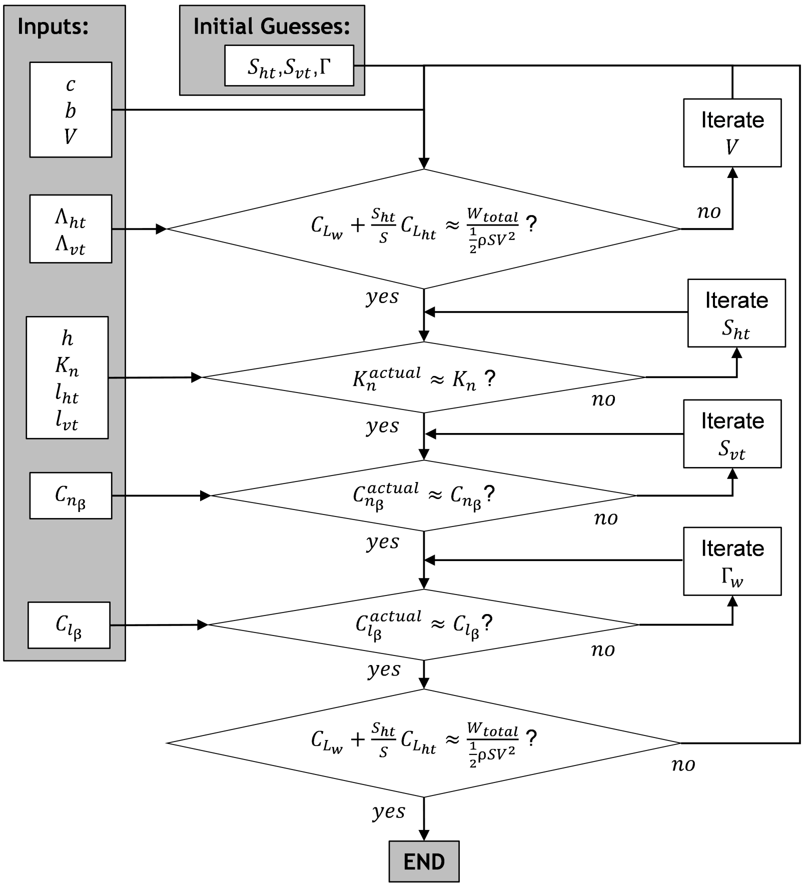

Inputs:
c
b
V
Λ_{ht}
Λ_{vt}
h
K_n
l_{ht}
l_{vt}
C_{n_β}
C_{l_β}
Initial Guesses:
S_{ht}, S_{vt}, Γ
Iterate V
$C_{L_w} + \frac{S_{ht}}{S} C_{L_{ht}} \approx \frac{W_{total}}{\frac{1}{2}\rho S V^2}$?
no
yes
Iterate S_{ht}
$K_n^{actual} \approx K_n$?
no
yes
Iterate S_{vt}
$C_{n_\beta}^{actual} \approx C_{n_\beta}$?
no
yes
Iterate Γ_w
$C_{l_\beta}^{actual} \approx C_{l_\beta}$?
no
yes
$C_{L_w} + \frac{S_{ht}}{S} C_{L_{ht}} \approx \frac{W_{total}}{\frac{1}{2}\rho S V^2}$?
no
yes
END

Figure 3.11: Static stability methodology.

order to calculate the stability derivatives that allow the computation of the matrices' eigenvalues, several theoretical and experimental correlations have been used [102]. These eigenvalues provide data on the phugoid and short-period for the longitudinal modes and rolling, spiral and dutch-roll for the lateral ones. These data enable the designer to conclude about the dynamic response of the aircraft.

The position and orientation is usually given relative to an Earth-fixed frame F_E. The airplane CG has coordinates (x_E, y_E, z_E). As for the orientation, this is given by a series of three consecutive rotations, called *Euler angles*. The airplane is imagined first to be oriented so that its axes are parallel to the axes of F_E. The following rotations are then applied:

- A rotation Ψ about oz_1, carrying the axes to $Cx_2y_2z_2$ (bringing Cx to its final azimuth);
- A rotation Θ about oy_2, carrying the axes to $Cx_3y_3z_3$ (bringing Cx to its final elevation);
- A rotation Φ about ox_3, carrying the axes to $Cxyz$.

The original and most complete equations of motion are commonly linearized for stability and control analysis. This theory assumes that the motion of the airplane consists of small deviations from the from a steady flight condition. The small disturbance theory has been found to provide good results [102]. This theory makes it possible to decouple the longitudinal and lateral modes of motion, as per System of Equations (3.85) and (3.86), respectively.

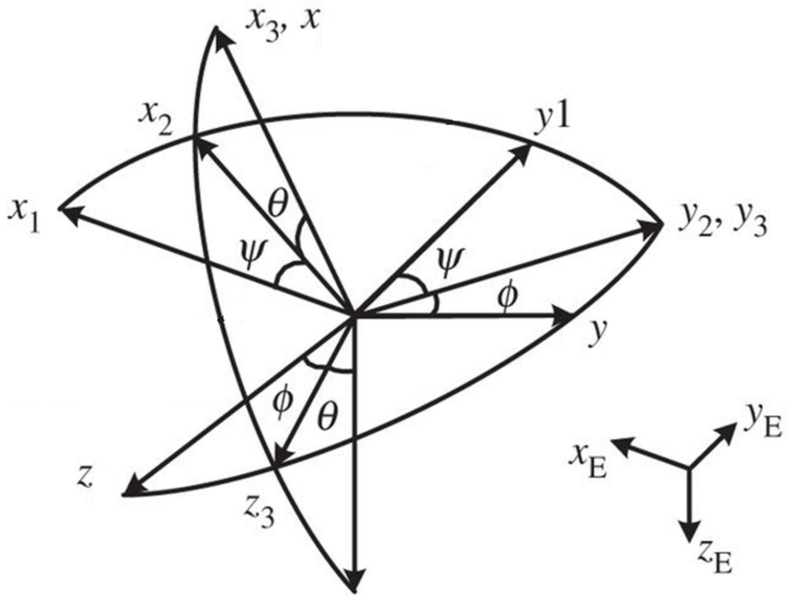


Figure 3.12: Airplane orientation axes.

$$\begin{bmatrix} \Delta\dot{u} \\ \dot{w} \\ \dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{X_u}{m} & \frac{X_w}{m} & 0 & -g\cos\theta_0 \\ \frac{Z_u}{m-Z_{\dot{w}}} & \frac{Z_w}{m-Z_{\dot{w}}} & \frac{Z_q+mu_0}{m-Z_{\dot{w}}} & \frac{-mg\sin\theta_0}{m-Z_{\dot{w}}} \\ \frac{1}{I_y}\left[M_u+\frac{M_{\dot{w}}Z_u}{m-Z_{\dot{w}}}\right] & \frac{1}{I_y}\left[M_w+\frac{M_{\dot{w}}Z_w}{m-Z_{\dot{w}}}\right] & \frac{1}{I_y}\left[M_q+\frac{M_{\dot{w}}(Z_q+mu_0)}{m-Z_{\dot{w}}}\right] & -\frac{1}{I_y}\left[\frac{M_{\dot{w}}mg\sin\theta_0}{m-Z_{\dot{w}}}\right] \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ w \\ q \\ \Delta\theta \end{bmatrix} \tag{3.85}$$

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_v}{m} & \frac{Y_p}{m} & \frac{Y_r}{m}-u_0 & g\cos\theta_0 \\ \frac{L_v}{I'_x}+I'_{zx}+N_v & \frac{L_p}{I'_x}+I'_{zx}N_p & \frac{L_r}{I'_x}+I'_{zx}N_r & 0 \\ I'_{zx}L_v+\frac{N_v}{I'_z} & I'_{zx}L_p+\frac{N_p}{I'_z} & I'_{zx}L_r+\frac{N_r}{I'_z} & 0 \\ 0 & 1 & \tan\theta_0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} \tag{3.86}$$

The doted quantities correspond to derivatives with respect to the time and the quantities with subscripts correspond to derivatives with respect to their subscripts. Conversely, (p, q, r) are the scalar components of the angular velocity vector, (u, v, w) are the scalar components of the velocity vector, (L, M, N) are the scalar components of the moment vector, (X, Y, Z) are the scalar components of the force vector. The symbols I_x, I_y, I_z represent the moments of inertia about the (x,y,z) axes, respectively.

Let us highlight the specific assumptions based upon which this small-disturbance theory holds. The existence of pure longitudinal motion depends on only two assumptions:

- the existence of a plane of symmetry;
- the absence of rotor gyroscopic effects.

The existence of pure lateral motion depends on a more restrictive set of assumptions:

- the linearization of the equations;
- the absence of rotor gyroscopic effects;
- the neglect of all aerodynamic cross-coupling.

$$\begin{cases} I'_x = (I_x I_z - {I_{zx}}^2)/I_z \\ I'_z = (I_x I_z - {I_{zx}}^2)/I_x \\ I'_z x = I_{zx}/(I_x I_z - {I_{zx}}^2) \end{cases} \tag{3.87}$$

3.4 Performance

Each mission stage will be simulated considering the approach described in the forthcoming subsections. Prior to that, it is important to refer that whenever the term velocity (V) is used, it refers to the airspeed, therefore, at zero ground speed (GS) the airspeed will equal the wind speed (V_{wind}), in accordance with Figure 3.13.

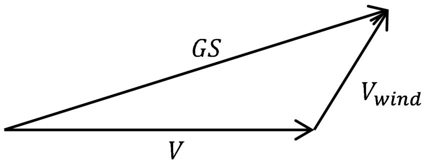


Figure 3.13: Velocity triangle.

Furthermore, it is worthwhile to mention that in all of the main mission stages described in subsections 3.4.1 through 3.4.2.2 - except for the take-off - the aircraft is trimmed, which means that its pitching moment relative to the aircraft gravity center is null. As for the take-off run, the vertical reaction forces that the landing gears make sure the aircraft is trimmed.

Before addressing the different mission stages, some relevant angular definitions must be made, namely:

- **aircraft angle-of-attack:** (α) - angle between the fuselage reference axis and the airspeed;
- **wing angle-of-attack:** (α_w) - angle between the wing mean chord and the airspeed;
- **climb/descent angle:** (γ) - angle between the horizontal plane and the airspeed;
- **wing incidence:** (i_w) - angle between the wing mean chord and the fuselage axis;
- **propulsion angle:** (τ) - angle between the fuselage axis and the thrust force vector;
- **pitch angle:** (θ) - angle between the fuselage axis and the horizontal plane;

All of these are graphically represented in Figure 3.14. It should be noted that the cruise stage is a particular case of climb at zero climb angle whereas the descent is a particular case of climb at negative climb angle. Nevertheless, the mission stages' models presented next are independent so that the reader can analyze each stage in isolation.

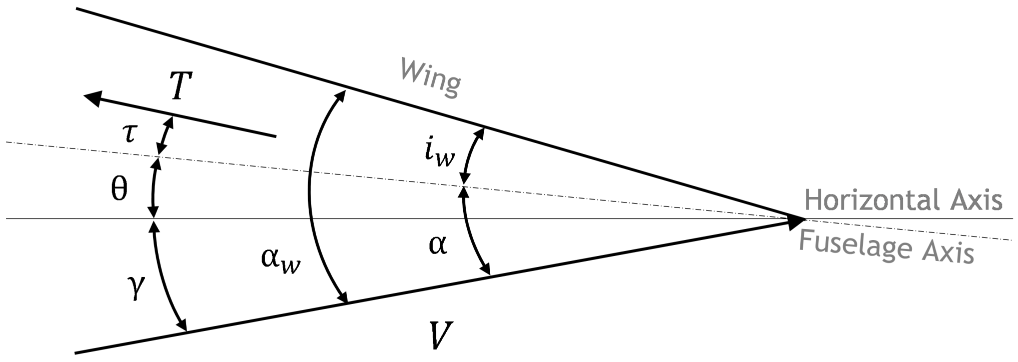


Figure 3.14: Angular definitions representation.

The mission stages representation described in the forthcoming sections relies on a discretization approach. This means that each mission stage is discretized in a number of time and/or velocity steps, during which the physical quantities are assumed constant. This is a good approximation when dealing with steady state flight and/or also when each mission stage is discretized in a large number of small time or velocity intervals, thus reducing the numerical error.

3.4.1 Take-off

The take-off is the first stage of every mission - with the exception of gliders which are towed and start their mission up in the air. Since this vehicle category is out of the scope of the present research, it is understood that the take-off is the first stage of every mission and the only stage that does not repeat itself in any mission profile.

The first approximation which will be used is to discretize the take-off run in velocity steps (ΔV_{step}), starting at zero ground speed, where the velocity will equal the wind speed (${V_1}^{initial} = V_{wind}$) and ending at the lift off velocity (${V_n}^{final} = V_{lo}$). The velocity at each of the ($n_{steps} + 1$) instants will thus be known, since the number of steps is predefined and both the wind velocity (V_{wind}) and lift off velocity (V_{lo}) shall be known upfront. Accordingly, the velocity step will correspond to the difference between two consecutive velocity instants, in accordance with Equation (3.88), which will be constant throughout the take-off run.

$$V_i^{final} = V_i^{initial} + \Delta V_{step} \tag{3.88}$$

The velocity considered at each take-off step or interval is a root mean square (RMS) (V) of the instant velocity at the begining of that step and the instant velocity at the end of that same interval:

$$V_i = \sqrt{\frac{\left(V_i^{initial}\right)^2 + \left(V_i^{final}\right)^2}{2}} \tag{3.89}$$

This is a RMS because the aerodynamic quantities will be a function of the square of the velocity at each time interval and therefore, it will provide better results. All the physical quantities described below will be dependent on the take-off step average velocity (V_i).

There will thus be (n_{steps}) which will correspond to the same number of velocity intervals. Let the generic variable (i) denote a generic velocity interval. Figure 3.15 provides a schematic representation of this procedure.

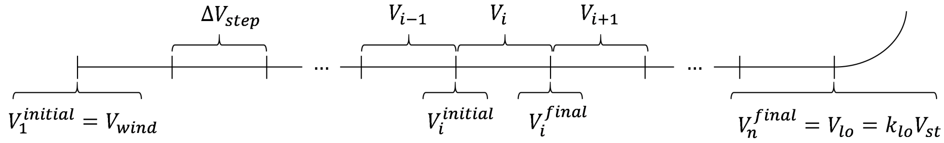
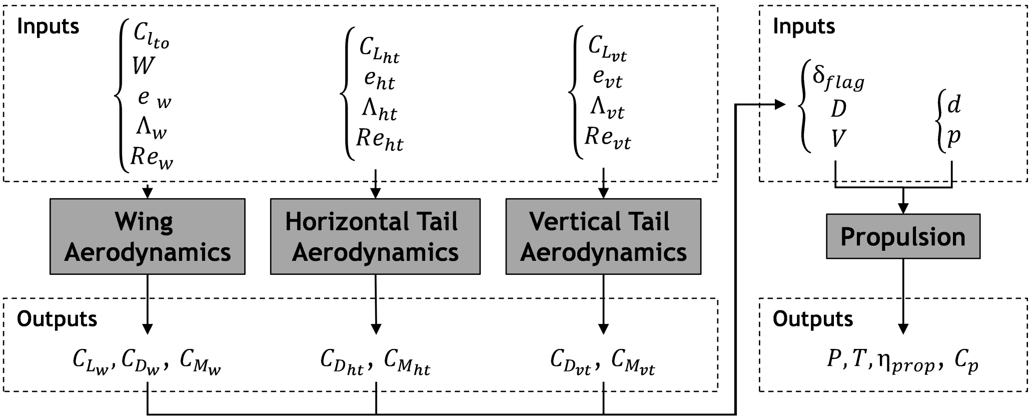


Figure 3.15: Take-off run velocity steps.

Generically, the take-off run is different from all the other mission stages in the way it does not require that the horizontal tail trim the aircraft. Because of that, the disciplinary models that compute the aerodynamic coefficients of the different lifting surfaces (wing, horizontal stabilizer and vertical stabilizer) and the propulsion can be independently called - depending on the optimization methodology in use. As the goal of this section is to describe the take-off run simulation approach, Figure 3.16 shows the most relevant inputs and outputs of the aerodynamics and propulsion disciplinary routines.



Inputs
$C_{l_{to}}$, W, e_w, Λ_w, Re_w
$C_{L_{ht}}$, e_{ht}, Λ_{ht}, Re_{ht}
$C_{L_{vt}}$, e_{vt}, Λ_{vt}, Re_{vt}
Wing Aerodynamics
Horizontal Tail Aerodynamics
Vertical Tail Aerodynamics
Outputs
$C_{L_w}, C_{D_w}, C_{M_w}$
$C_{D_{ht}}, C_{M_{ht}}$
$C_{D_{vt}}, C_{M_{vt}}$
Inputs
δ_{flag}, D, V
d, p
Propulsion
Outputs
P, T, η_{prop}, C_p

Figure 3.16: Physical models interactions for the take-off stage.

The lift force (L) will thus be the sum of the wing and horizontal tail contributions:

$$L_i = 0.5\rho V_i^2 (S_w C_{L_w} + S_{ht} C_{L_{ht}}) \tag{3.90}$$

As for the friction drag (D_{fric}) it will depend on the vehicles weight (W), lift force (L) and dynamic friction coefficient (μ). Conversely, the aerodynamic drag force (D_{aero}) will be a sum of the wing, horizontal tail, vertical tail, fuselage and miscellaneous contributions, being that the landing gear contributions are part of the miscellaneous term. The total drag at each step is then estimated as:

$$D_i = \mu(W_i - L_i) + 0.5\rho {V_i}^2(SC_{D_w} + S_{ht}C_{D_{ht}} + S_{vt}C_{D_{vt}} + SC_{D_{misc}}) + D_{fus} \tag{3.91}$$

Once the thrust force (T) is known, it is possible to know what is the acceleration (a) to which the aircraft will be subject at each step:

$$a_i = \frac{T_i - D_i}{W/g} \tag{3.92}$$

Finally, the amount of time spent at each velocity step can be computed as:

$$\Delta t_i = \frac{\Delta V_{step}}{a_i} \tag{3.93}$$

Now that it is clear what happens at each take-off step, it is important to understand what happens beyond it. Firstly, some initial guesses for the take-off weight and airfoil lift coefficients will start the take-off run. After running it once, the iterative cycle implemented will make it follow different directions depending on the objective function chosen – either maximizing the take-off weight (**Option 1**) or minimizing the energy weight for a surveillance mission (**Option 2**), as shown in Figure 3.17.

If Option 1 is chosen, the weight (W) will be iterated to use all the take-off distance available to lift the maximum payload and an external iterative cycle will iterate the airfoil lift coefficient (C_l) – varying aircraft ground attitude – to further increase the lifted payload so as to maximize the take-off run acceleration. Conversely, if Option 2 is chosen, the payload weight will be fixed and airfoil lift coefficient will be iterated in order to minimize the take-off distance (x_{to}) for that same weight. In both cases however, the DTOW will be updated at each step of the take-off run if the combustion engine is chosen. The pitch angle (θ) will depend on both the optimum AOA for take-off (α_w^{to}) and on the wing incidence for cruise (i_w^{cz}).

In Figure 3.17, the symbols (ξ^l), (ξ^m) and (ξ^n) refer to non-dimensional damping coefficients. These have been defined in Subsection 3.3.2.1.

3.4.2 Flight Stages

As it is known, from Newton's Second Law in the particular case where the mass does not vary (Equation (3.94)):

$$m.a = \sum F \tag{3.94}$$

, where the (a) refers to the vehicle's acceleration. By replacing the force by the force components acting on the aircraft along the flight path (see Figure 3.19a), one can obtain (Equation (3.95)).

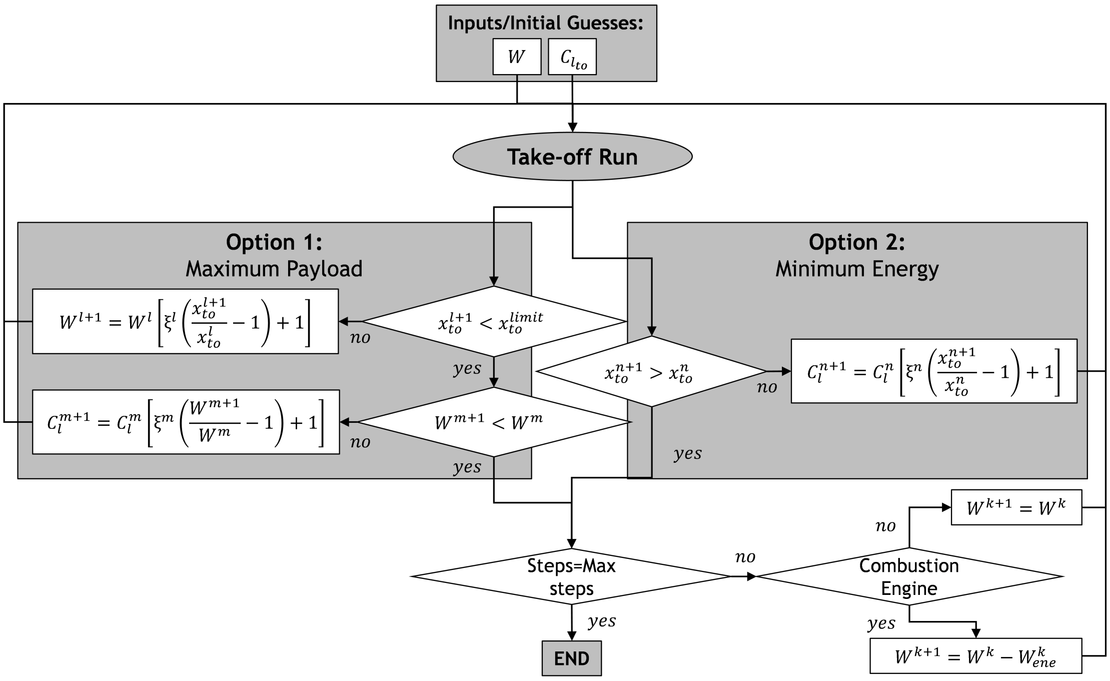

Inputs/Initial Guesses:
W
$C_{l_{to}}$
Take-off Run
Option 1:
Maximum Payload
Option 2:
Minimum Energy
$W^{l+1} = W^{l}\left[\xi^{l}\left(\frac{x_{to}^{l+1}}{x_{to}^{l}} - 1\right) + 1\right]$
$x_{to}^{l+1} < x_{to}^{limit}$
no
yes
$x_{to}^{n+1} > x_{to}^{n}$
no
$C_{l}^{n+1} = C_{l}^{n}\left[\xi^{n}\left(\frac{x_{to}^{n+1}}{x_{to}^{n}} - 1\right) + 1\right]$
$C_{l}^{m+1} = C_{l}^{m}\left[\xi^{m}\left(\frac{W^{m+1}}{W^{m}} - 1\right) + 1\right]$
$W^{m+1} < W^{m}$
no
yes
yes
$W^{k+1} = W^{k}$
no
no
Steps=Max steps
Combustion Engine
yes
yes
END
$W^{k+1} = W^{k} - W_{ene}^{k}$

Figure 3.17: Block diagram of the take-off subroutine.

$$\frac{W}{g}\frac{dV}{dt} = T.cos(\tau + \alpha) - D - W.sin(\gamma) \tag{3.95}$$

One can thus obtain Equation (3.96), where thrust must match the sum of these forces, the one due to the aerodynamic drag (D), the one related to the altitude change ($W.sin(\gamma)$) and the one due to velocity change ($W/g.dV/dt$).

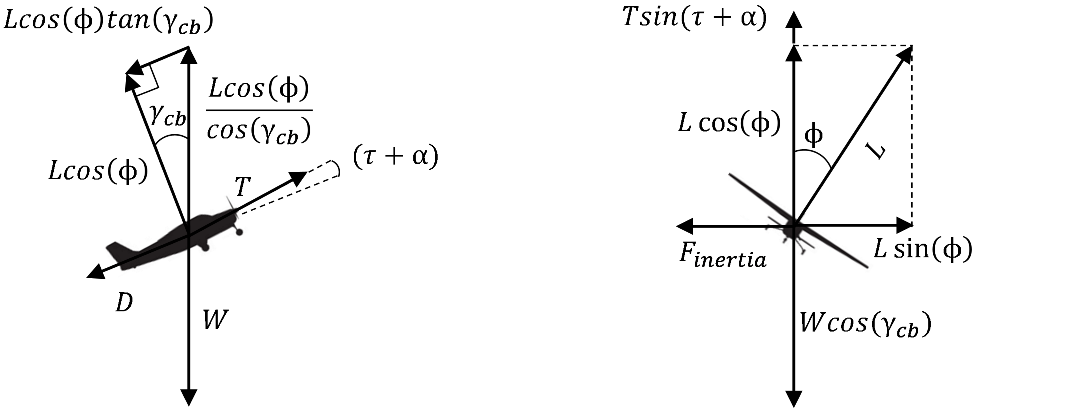
$$T = \frac{1}{cos(\tau + \alpha)}\left(D + W.sin(\gamma) + \frac{W}{g}.\frac{dV}{dt}\right) \tag{3.96}$$

Multiplying by the velocity one can obtain the propulsive power required for propelling the aircraft:

$$P = \frac{1}{cos(\tau + \alpha)}\left(DV + W.w + \frac{W}{g}V\frac{dV}{dt}\right) \tag{3.97}$$

, where ($w = dh/dt$) which corresponds to the vehicle RoC.

The forthcoming sub subsections show the particular Equations for the different flight stages (climb, cruise, loiter and descent), which equilibrium equations will be particular cases of the ones presented above, e.g. zero and negative climb angle for the cruise and descent stages, respectively.



(a) Side view.

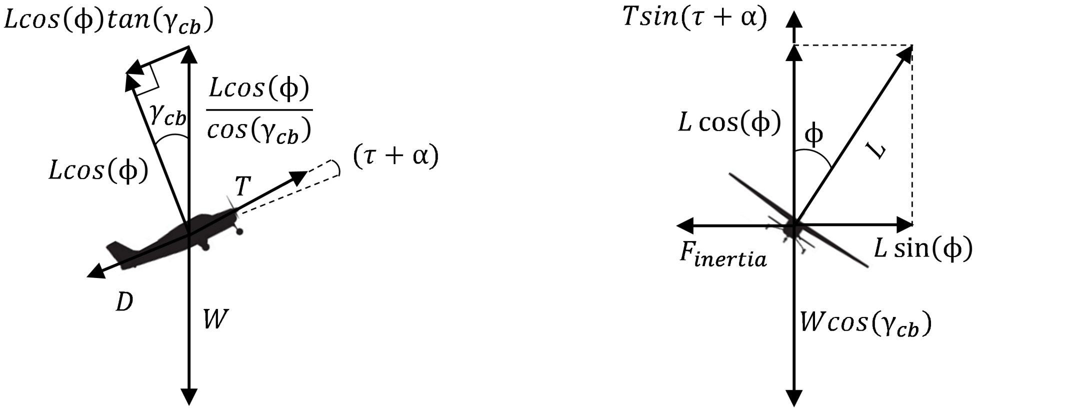
(b) Plane perpendicular to the aircraft trajectory.

Figure 3.18: Vertical diagram of forces of a generic flight stage.

3.4.2.1 Climb and Descent

The climb is a flight phase that corresponds to a positive change in the aircraft's flying altitude. Generically, the aircraft may be turning at a predefined bank angle while climbing, which results in an increased load factor with respect to the straight climb condition. Figures 3.19a and 3.19b show the side and front view of the vertical forces diagrams, respectively. Let (γ_{cb}) be the climb angle, (ϕ) be the bank angle, (θ) be the pitch angle, (L) be the aircraft lift, (W) be the aircraft weight, (D) be the aircraft drag, (T) be the aircraft thrust and ($F_{inertia}$) be an inertial centrifugal force.

The same Equations apply for the Descent mission stage. The only difference lies on the climb angle becoming negative.



(a) Side view.

(b) Plane perpendicular to the aircraft trajectory.

Figure 3.19: Vertical diagram of forces of the climb stage

Equations (3.98) show the balance of forces depicted in Figures 3.19a and 3.19b, respectively.

$$\begin{cases} T = \frac{D + W sin(\gamma_{cb}) + \frac{W}{g} \cdot \frac{dV}{dt}}{cos(\tau + \alpha)} \\ L = \frac{W cos(\gamma_{cb}) - T sin(\tau + \alpha)}{cos(\phi)} \\ \frac{W}{g}\frac{V^2}{R} = L sin(\phi) \end{cases} \tag{3.98}$$

From Equations (3.98), it is possible to determine the load factor (n) of the climb turn as a function of the bank angle (ϕ) and of the climb angle (γ_{cb}). For small ($\tau + \alpha$), the load factor can be calculated via:

$$n = \frac{cos(\gamma_{cb})}{cos(\phi)} \tag{3.99}$$

The methodology adopted for modeling the climb phase consisted of discretizing it in a number of different height intervals. Let the generic variable (i) denote a generic height interval. Figures 3.20 and 3.21 provide a schematic representation of this procedure for both the straight and turn climb conditions.

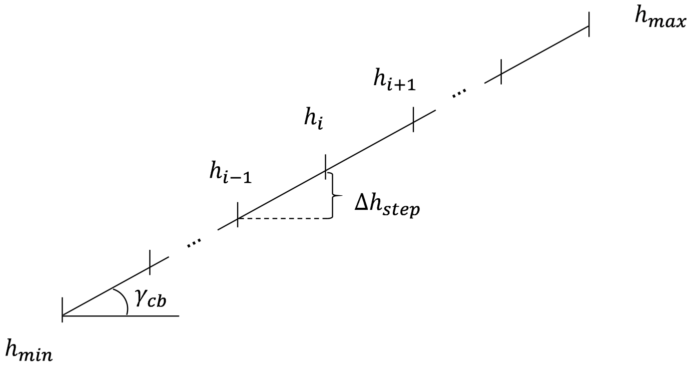


Figure 3.20: Schematic representation of the climb stage height steps (side view).

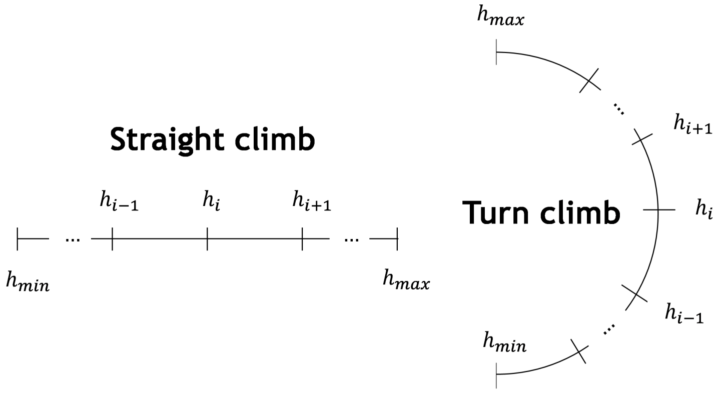


Figure 3.21: Schematic representation of the climb stage height steps in the straight and turn conditions (top view).

For the aircraft to be trimmed and statically stable, the static stability routine must be run to make sure that the horizontal and vertical stabilizer are duly sized, as explained in Section 3.3.4. Figure 3.22 depicts the most relevant inputs and outputs of the aerodynamics, static stability and propulsion disciplinary routines. The required lift (L) is the product of the vehicle's weight (W) by the load factor (n). Likewise, the actual airfoil lift coefficient ($C_{l_{cb}}$) is the product of the user-defined airfoil lift coefficient by the load factor (n), to make sure that in a non-straight flight condition there is a balance of forces, as shown in Figure 3.22.

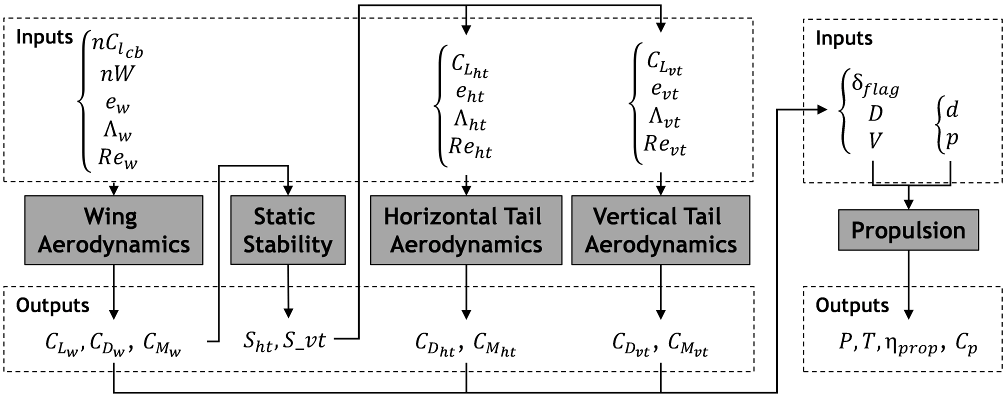


Figure 3.22: Physical models interactions for the climb stage.

The lift force (L) will thus be the sum of the wing and horizontal tail contributions:

$$L_i = 0.5\rho V_i^2(S_w C_{L_w} + S_{ht} C_{L_{ht}}) \tag{3.100}$$

The aerodynamic drag at each time step (D_i^{aero}) is:

$$D_i^{aero} = 0.5\rho {V_i}^2(SC_{D_w} + S_{ht}C_{D_{ht}} + S_{vt}C_{D_{vt}} + SC_{D_{misc}}) + D_{fus} \tag{3.101}$$

, which is the sum of the wing (C_{D_w}), horizontal tail ($C_{D_{ht}}$), vertical tail ($C_{D_{vt}}$), fuselage (D_{fus}) and miscellaneous ($C_{D_{misc}}$) contributions, since the landing gears contribution is included in the miscellaneous term.

From the drag computation it is possible to calculate the power required for climbing at constant airspeed (P_{req}), Equation (3.102).

$$P_{req_i} = \left[\frac{D_i^{aero} + W_i sin(\gamma_{cb}) + \frac{W}{g}\cdot\frac{dV}{dt}}{cos(\tau + \alpha)}\right] V_i \tag{3.102}$$

At each climb step, the overall power (P_i) is the sum of the three aforementioned contributions (aerodynamic drag, potential energy change and kinetic energy change):

$$P_i = \left[\frac{D_i^{aero} + W_i sin(\gamma_{cb})}{cos(\tau + \alpha)}\right] V_i + \frac{W_i}{g.cos(\tau + \alpha)} V_i a_i \tag{3.103}$$

, where the local acceleration is represented by ($a_i = dV/dt$). Its value is computed using the finite differences approach ($a_i = (V_i - V_{i-1})/\Delta t_{step}$).

The generic thrust force required at each time step can be obtained by dividing Equation (3.103) by the velocity (V_i), which results in:

$$T_i = \frac{D_i^{aero} + W_i sin(\gamma_{cb})}{cos(\tau + \alpha)} + \frac{W_i}{g.cos(\tau + \alpha)} a_i \tag{3.104}$$

Once the required thrust at each height step (T_i) is known, the propulsion module can be run to estimate the actual energy consumption and all the remaining propulsion-related data. The air density (ρ) changes with altitude, and so does the vehicle's weight (W_i) in the combustion engine case, which means that it is relevant to have a fair number of discretization steps. In the case of an accelerated climb, it is also important to update the velocity at each step, further reinforcing the relevance of the number of discretization steps.

The greater the number of steps, the lower will obviously be the approximation errors and the highest the computational cost will be. It is important to note that the load factor due to the climb angle (γ_{cb}) and/or due to the bank angle (ϕ) will result in multiplying the actual weight (W) by the respective load factor (n).

Now that it is clear what happens at each climb step, it is important to understand what happens beyond each climb step. Firstly, an initial guess for the airfoil lift coefficient ($C_{l_{cb}}$) will start the climb subroutine.

After running each altitude increment, this subroutine will also act differently based on the user objective function's option, as shown in Figure 3.23. Under **Option A**, which maximizes the power setting with the Rate-of-climb (RoC) being a dependent variable, the velocity will be iterated to make sure the forces are in equilibrium. The minimum safe velocity for flight, which is the product of the stall velocity (V_{st}) by the stall velocity coefficient (k_{st}), has to be respected, and therefore, if the iterated velocity does not respect this lower limit, the climb angle (γ_{cb}) is iterated with the airfoil lift coefficient ($C_{l_{cb}}$) and velocity (V) remaining unchanged, as shown in Figure 3.23.

Once the equilibrium of forces is attained, the RoC will be maximized. If the final result does not exceed a RoC threshold – minimum RoC admissible by the user – it means that the most conservative mission stage for lifting payload is climbing for a fixed maximum power setting and the take-off weight needs to be corrected accordingly. Alternatively the engine and propeller can be changed.

In **Option B**, the (RoC) is an input and the velocity will be iterated to make sure the forces are in equilibrium. If the converged thrust force is greater than the maximum thrust the engine or motor plus propeller combination can deliver at that velocity, the routine will iterate the airfoil climb lift coefficient until the thrust delivered is below its maximum. If after the climb airfoil lift coefficient ($C_{l_{cb}}$) is corrected, the thrust still exceeds the one available, the weight (W) needs to be corrected and the take-off simulation run again.

In the diagram of Figure 3.23, the symbols (δ^{l1}), (δ^{l2}), (δ^{m}) and (δ^{n}) refer to non-dimensional increments which are used in their respective iterative convergence procedure. These have been defined in Subsection 3.3.2.1.

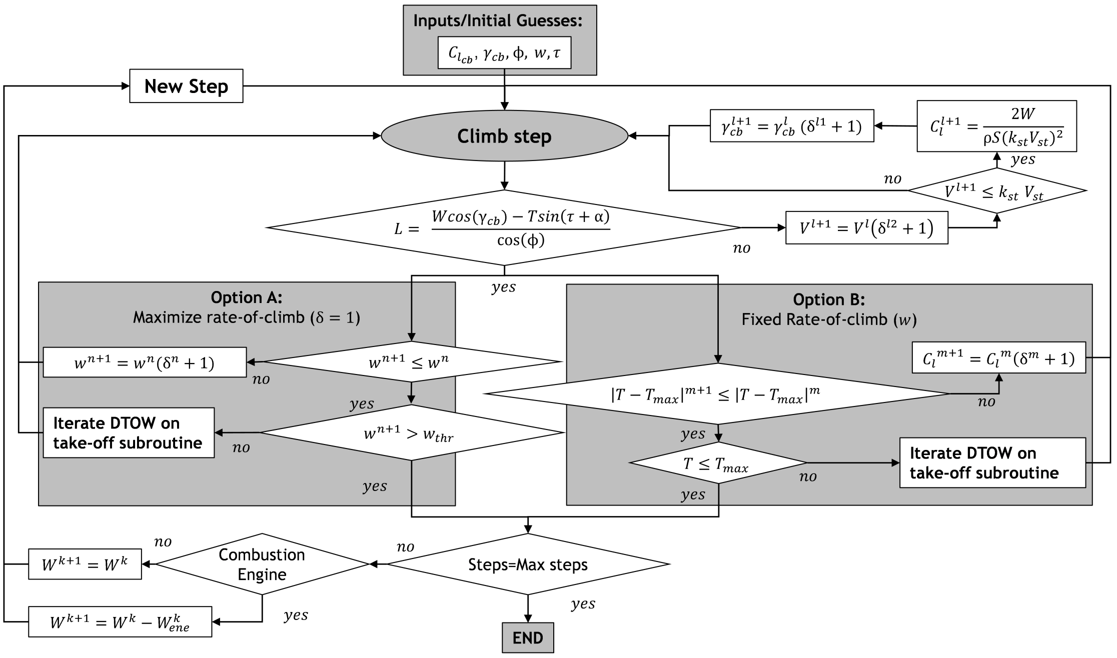

Inputs/Initial Guesses:
$C_{l_{cb}}, \gamma_{cb}, \phi, w, \tau$
New Step
Climb step
$\gamma_{cb}^{l+1} = \gamma_{cb}^{l}\,(\delta^{l1}+1)$
$C_l^{l+1} = \frac{2W}{\rho S (k_{st} V_{st})^2}$
yes
no
$V^{l+1} \le k_{st}\, V_{st}$
$L = \frac{W\cos(\gamma_{cb}) - T\sin(\tau+\alpha)}{\cos(\phi)}$
$V^{l+1} = V^{l}(\delta^{l2}+1)$
no
yes
Option A:
Maximize rate-of-climb ($\delta = 1$)
$w^{n+1} = w^{n}(\delta^{n}+1)$
$w^{n+1} \le w^{n}$
no
yes
Iterate DTOW on take-off subroutine
$w^{n+1} > w_{thr}$
no
yes
Option B:
Fixed Rate-of-climb (w)
$C_l^{m+1} = C_l^{m}(\delta^{m}+1)$
$|T - T_{max}|^{m+1} \le |T - T_{max}|^{m}$
no
yes
$T \le T_{max}$
no
Iterate DTOW on take-off subroutine
yes
no
$W^{k+1} = W^{k}$
Combustion Engine
no
Steps=Max steps
$W^{k+1} = W^{k} - W_{ene}^{k}$
yes
yes
END

Figure 3.23: Block diagram of the climb subroutine.

As for the Descent, stage, an initial guess for the airfoil lift coefficient – through the input data file – will get the descent subroutine started. To make sure the vertical forces balance themselves, the velocity will be iterated (Fig. 3.24). The minimum safe velocity for flight, which is the product of the stall velocity (V_{st}) by the stall velocity coefficient (k_{st}), has to be respected, and therefore, if the iterated velocity does not respect this lower limit, the descent angle (γ_{dt}) is iterated with the airfoil lift coefficient ($C_{l_{dt}}$) and velocity (V) remaining unchanged, as shown in Figure 3.24.

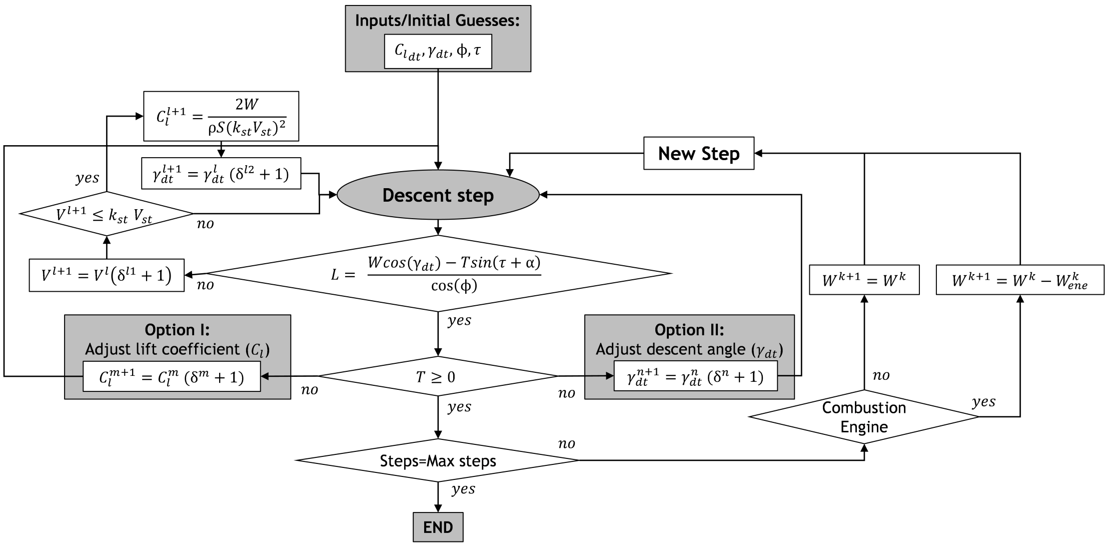

Inputs/Initial Guesses:
$C_{l_{dt}}, \gamma_{dt}, \phi, \tau$
$C_l^{l+1} = \frac{2W}{\rho S (k_{st} V_{st})^2}$
$\gamma_{dt}^{l+1} = \gamma_{dt}^{l}\,(\delta^{l2}+1)$
yes
$V^{l+1} \le k_{st}\, V_{st}$
no
Descent step
New Step
$V^{l+1} = V^{l}(\delta^{l1}+1)$
no
$L = \frac{W\cos(\gamma_{dt}) - T\sin(\tau+\alpha)}{\cos(\phi)}$
$W^{k+1} = W^{k}$
$W^{k+1} = W^{k} - W_{ene}^{k}$
yes
Option I:
Adjust lift coefficient (C_l)
$C_l^{m+1} = C_l^{m}\,(\delta^{m}+1)$
no
$T \ge 0$
no
Option II:
Adjust descent angle (γ_{dt})
$\gamma_{dt}^{n+1} = \gamma_{dt}^{n}\,(\delta^{n}+1)$
no
yes
Combustion Engine
yes
no
Steps=Max steps
yes
END

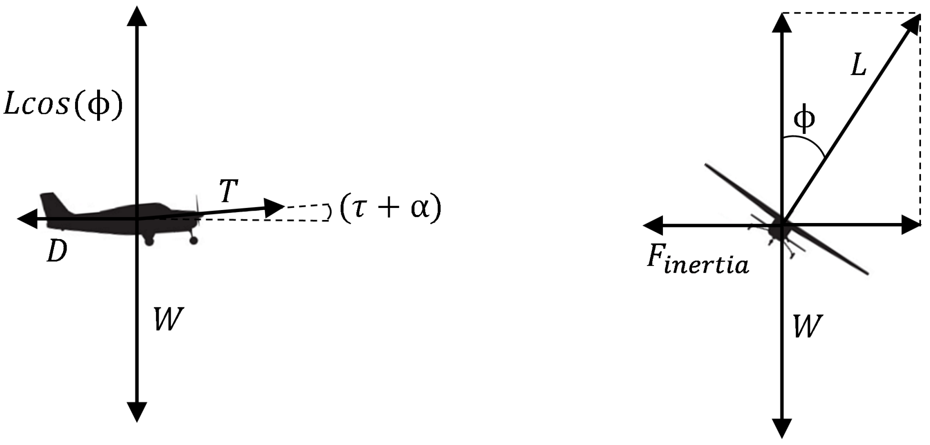
Figure 3.24: Block diagram of the descent subroutine.

It should be noted that if a negative thrust force is obtained, that is only physically possible using thrust reversal. As that is not common in small to medium sized UAVs, that possibility has not been considered. As such, to avoid it, either the airfoil lift coefficient **Option I** or the descent angle **Option II** will be iterated, in accordance with the user settings, as shown in Figure 3.24.

In the diagram of Figure 3.24, the symbols (δ^{l1}), (δ^{l2}) and (δ^{m}) refer to non-dimensional increments. These have been defined in Subsection 3.3.2.1.

3.4.2.2 Leveled Flight - Cruise and Loiter

The cruise and loiter correspond to either straight flight or a sustained turn, respectively. Figures 3.25a and 3.25b show the generic side and front view of the vertical forces diagrams, respectively. In the first condition, the load factor (n) is equal to unity, whereas in the second its value is computed as follows from Figure 3.25b and the system of Equations (3.105). Let (L) be the lift, (W) be the weight, ($F_{inertial}$) be the inertial centrifugal force, (R) be the turn radius, (ϕ) be the bank angle, (α) be the aircraft angle-of-attack and (τ) be the propulsive force angle with respect to the fuselage axis.



(a) Side view.

(b) Front view.

Figure 3.25: Vertical diagram of forces of the cruise stage

$$\begin{cases} Lcos(\phi) = W - Tsin(\tau + \alpha) \\ D = Tcos(\tau + \alpha) \end{cases} \tag{3.105}$$

Assuming small ($\tau + \alpha$):

$$\begin{cases} W = Lcos(\phi) \\ \frac{W}{g}\frac{V^2}{R} = Lsin(\phi) \end{cases} \tag{3.106}$$

From the system of Equations (3.106), it is possible to determine the load factor (n) of the sustained turn as a function of the bank angle (ϕ):

$$n = \frac{1}{cos(\phi)} \tag{3.107}$$

The methodology adopted consisted of discretizing the cruise stage in a number of different time intervals (n_{steps}). Let the generic variable (i) denote a generic time interval. Figure 3.26 provides a schematic representation of this procedure for both the leveled and sustained turn flight conditions.

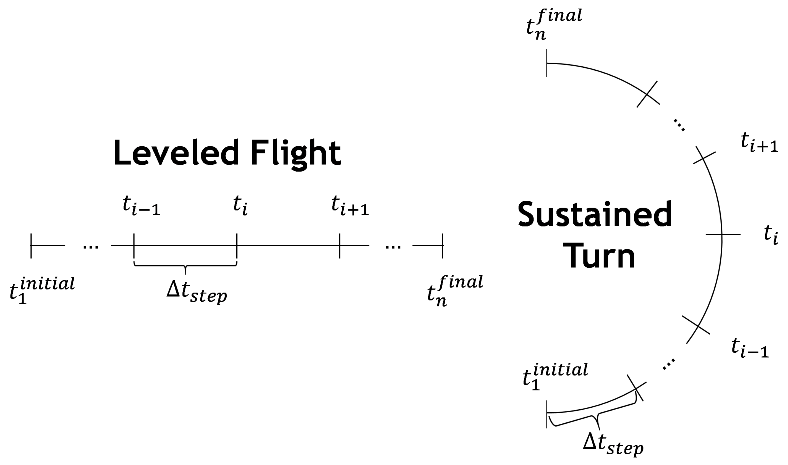


Figure 3.26: Schematic representation of the cruise stage time steps in the leveled flight and sustained turn conditions (top view).

For the aircraft to be trimmed and statically stable, the static stability routine must be run to make sure that the horizontal and vertical stabilizer are duly sized, as explained in Section 3.3.4. Figure 3.27 depicts the most important inputs and outputs of the aerodynamics, static stability and propulsion disciplinary routines. The required lift (L) is the product of the vehicle's weight (W) by the load factor (n), whereas the actual airfoil lift coefficient ($C_{l_{cz}}$) is the product of the user defined airfoil lift coefficient by the load factor (n), to make sure that in a banked flight condition there is a balance of forces, as shown in Figure 3.27.

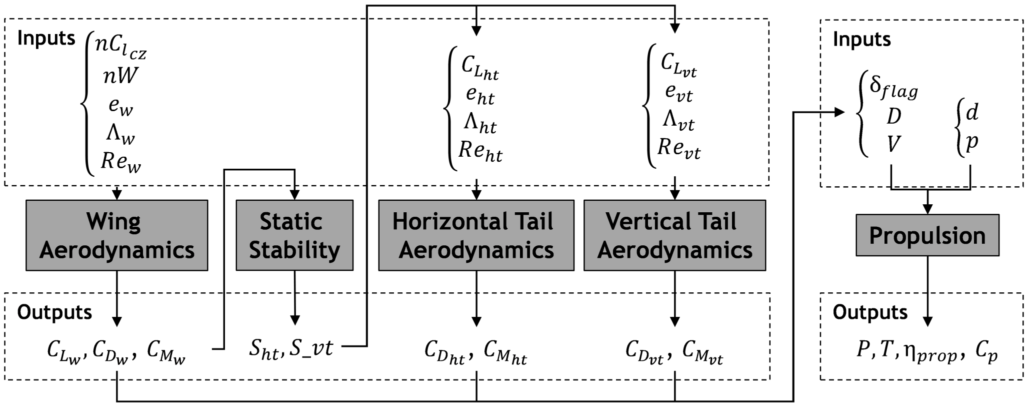


Figure 3.27: Physical models interactions for the cruise stage.

The lift force (L) will thus be the sum of the wing and horizontal tail contributions:

$$L_i = 0.5\rho V_i^2(S_w C_{L_w} + S_{ht} C_{L_{ht}}) \tag{3.108}$$

The total drag (D) is equal to the aerodynamic drag (D_i^{aero}), which is the sum of the wing, horizontal tail, vertical tail, fuselage and miscellaneous contributions, being that the landing gear contributions are part of the miscellaneous term:

$$D_i^{aero} = 0.5\rho V_i{}^2(SC_{D_w} + S_{ht} C_{D_{ht}} + S_{vt} C_{D_{vt}} + SC_{D_{misc}}) + D_{fus} \tag{3.109}$$

Following the diagram of forces of Figure 3.25a, it is possible to obtain the thrust from the total drag force (D), the aircraft angle-of-attack and the thrust force angle (α) with respect to the fuselage horizontal angle (τ):

$$T_i = \frac{0.5\rho V_i{}^2(SC_{D_w} + S_{ht} C_{D_{ht}} + S_{vt} C_{D_{vt}} + SC_{D_{misc}}) + D_{fus}}{cos(\tau + \alpha)} \tag{3.110}$$

Therefore, the propulsion module can be run to estimate the energy consumption and all the remaining propulsion-related data. Assuming that the wing airfoil lift coefficient is a user input, the velocity at each time step will only change with a change in the vehicle's weight (W), which will only happen in the case of a combustion engine, as fuel is being spent. Therefore, in the electric motor case, the number of steps in which each cruise stage is discretized is not relevant - since it is assumed that the air density remains constant throughout this mission stage.

Beyond each cruise step, there is an initial guess for the airfoil lift coefficient ($C_{l_{cz}}$), as shown in Figure 3.28. In case the lift does not equal weight, the velocity is iterated and the power setting is adjusted accordingly. In case the lift (L) equals weight (W) before the velocity (V) goes out of the boundaries defined by the user, the wing airfoil lift coefficient will not be changed.

However, if this is not the case, the airfoil lift coefficient will be iterated to make sure the cruise velocity is within the user defined range. Furthermore, the minimum safe velocity for flight, which is the product of the stall velocity (V_{st}) by the stall velocity coefficient (k_{st}) overrules the user defined minimum velocity if it happens to be greater than the user defined minimum, as shown in Figure 3.28. This is to make sure that the aircraft never flies outside its flight envelope. Likewise the former subroutines, the weight shall be updated at each cruise step if a combustion engine is in use.

In the diagram of Figure 3.28, the symbols (δ^m) and (δ^n) refer to non-dimensional increments. These have been defined in Subsection 3.3.2.1.

3.4.3 Metrics

The PARROT analysis methodology (Chapter 4) enables an ad-hoc design optimization through an informal process via the multiple parametric plots that are generated. Its primary

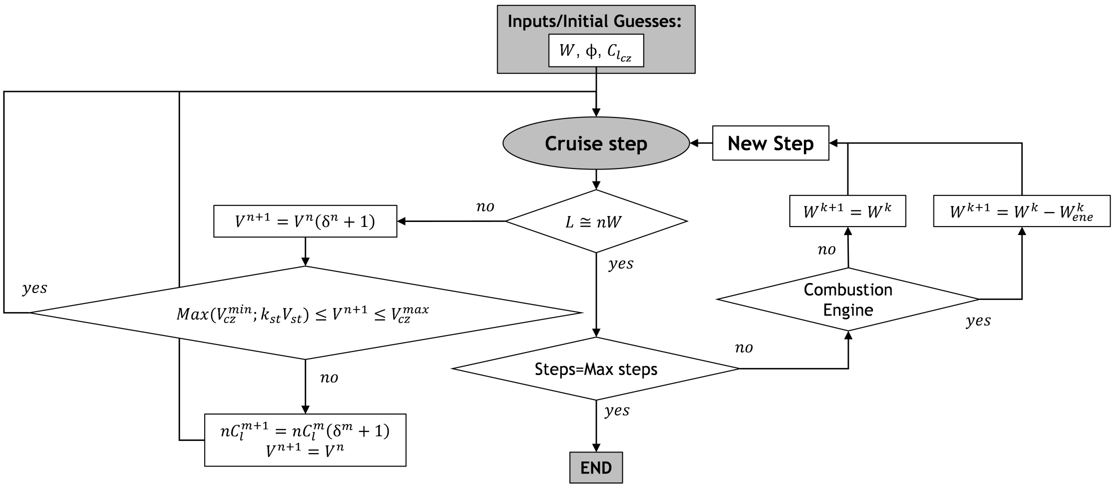

Inputs/Initial Guesses:
W, ϕ, $C_{l_{cz}}$
Cruise step
New Step
no
$V^{n+1} = V^{n}(\delta^{n} + 1)$
$L \cong nW$
$W^{k+1} = W^{k}$
$W^{k+1} = W^{k} - W_{ene}^{k}$
no
yes
yes
$Max(V_{cz}^{min}; k_{st}V_{st}) \leq V^{n+1} \leq V_{cz}^{max}$
Combustion Engine
yes
no
Steps=Max steps
no
yes
$nC_{l}^{m+1} = nC_{l}^{m}(\delta^{m} + 1)$
$V^{n+1} = V^{n}$
END

Figure 3.28: Block diagram of the cruise subroutine.

design variables are the wing mean chord and the wingspan. However, other parameters can be studied, just by keeping all the others constant and evaluating the impact of changing them. The PARROT code features two analysis modes. In the first working mode, the goal is to select the best wing layout for minimum energy consumption (for a typical surveillance mission when one usually aim at maximizing the aircraft range and/or endurance). Contrarily, on the second working mode, the code enables the user to find the best wing layout to maximize the payload weight lifted for a particular mission profile.

The MTOP methodology (Chapter 5) includes a formal mission-based design optimization methodology which objective function is the overall energy consumption. Its design variables may include the win mean chord, wingspan, propeller diameter, propeller pitch, flap chord fraction and flap deflection angle. While the wing mean chord, propeller diameter and flap chord fraction are always global variables, the wingspan and propeller pitch may be either global or local variables, whereas the flap deflection angle is always a local variable.

It is also worthwhile to discuss that in both methodologies, the user defined mission profile is discretized and the aircraft performance is computed at each point, so that the aircraft is permanently trimmed. The refined propulsion model, which builds on propeller performance data and on the combustion engine and electric motor models evaluates the energetic consumption at each step.

3.5 Chapter Summary

This chapter has presented the low-fidelity physical models for the most relevant aircraft preliminary design disciplines: aerodynamics, propulsion, performance, weight and stability (static and dynamic). The devised models can be used separately (for disciplinary analyses) or together (for multidisciplinary analyses).

For the airfoils analysis – and given the low Reynolds number expected for UAV operations – the XFOIL software has been chosen. The user selects a number of different airfoils which are run in XFOIL for a set of Reynolds numbers with some approximate analytical methods being used to estimate the respective 3D aerodynamic coefficients from the XFOIL's 2D data.

Experimental correlations have been used for estimating the fuselage and miscellaneous drag components.

Based on the propulsion source - either combustion engine or an electrical motor - one of two iterative procedures is run to determine all the relevant propulsive figures of merit. To feed these procedures, the user has to load the propeller performance data from either an external software or from experimental measurements.

The structural weight estimates have been most prominently based on reference's experimental correlations, a parametric study undertaken using a FEM software for the VSW weight model and also structural weight records from previously built aircraft structures using similar materials and manufacturing techniques as the one under analysis.

The stability data is based on several analytical computation methods, combined with a number of experimental correlations to feed some stability derivatives which can not be appropriately analytically estimated.

The disciplinary analysis models presented herein have been presented with the goal of feeding the design analysis and optimization methodologies presented in Chapters 3 and 4.

Chapter 4

Parametric Design Study

4.1 Introduction

The first of the two methodologies developed in this thesis is a mission-based parametric multidisciplinary design analysis tailored for UAVs, which has been implemented with the goal of feeding the designer with relevant information to guide his/her decision-making process at the beginning of the preliminary design stage (Section 1.2.2).

An adequate aircraft geometric sizing is paramount to obtain a high performance design. Size and mass also have a close correlation with costs. The design methodology presented in this chapter is based on an extensive parametric study developed in-house in a spreadsheet by Gamboa et al (2013) [105] which has subsequently been implemented in a Fortran code for the sake of efficiency and easiness of access. In addition, it is easier to enhance it in the future as it has been developed in a modular way. This code is named **P**arametric **A**i**R**c**R**aft **O**p**T**imization (PARROT) and includes a graphical user interface which further enhances its user friendliness.

Parametric studies have been used for several decades and are an interesting approach based upon which traditional graphical methods can be used to find the maximum or minimum of a multivariable function. However, despite the advantages of the graphical method, which enables a full view over the design domain preventing the local extrema problem, this approach becomes inadequate for problems with more than 3-4 design variables.

Nonetheless, one of the greatest assets of the parametric approach is to feed the designer with a number of different plots that show how the variation of the mainstream design variables impact the objective function(s). Parametric studies enable the designer to understand the influence of the design variables on other parameters and on the figures of merit. It is thus believed that this methodology will help the designer get closer to the optimum solution early in the design process than it would perhaps happen without such comprehensive overview. The outcome of the parametric design can thus be used as a starting point for an optimization procedure. Furthermore, it is expected to contribute to a more efficient design process fostering a faster preliminary design stage than traditional design procedures would.

This Chapter includes the UAV design analysis methodology (Section 4.2), named PARROT. It also features a brief overview on its graphical user interface (Section 4.3) as well as two case studies (Section 4.4) which attest the worthiness of this computational tool. The chapter ends with some concluding remarks (Section 4.5).

4.2 Methodology

The UAV design analysis methodology presented in this Chapter is based on an extensive parametric study developed in-house in a Fortran code (PARROT). The most distinguishing feature of this code is that it enables a mission-based design study, contrarily to some conventional design approaches which tend to optimize the aircraft for a specific flight condition. Its primary design parameters are the wingspan (b) and the wing mean chord ($\bar{c}$), although the number of design parameters can be easily extended to other important design parameters such as the wing airfoil lift coefficient (C_l), the wing planform shape, through the Oswald efficiency factor (e_0), among others.

Accordingly, the mission analysis is run for a number of scattered combinations of wingspan and wing mean chord with a significant number of performance metrics and objective functions being estimated for each different wing geometry. These results are stored in matrix form and can thereafter be used for generating plots which will not only inform the designer of the best wing geometry ($\bar{c}$,b) for a specific purpose (objective function), but also make him/her understand the impact of such choice on other performance functions. The PARROT code makes use of the low fidelity disciplinary models presented in Chapter 2, namely for the aerodynamics (3.3.1), propulsion (3.3.2), weight (3.3.3), static stability (3.3.4) and dynamic stability (3.3.5). The design algorithm is based on the mission profile definition, where all the performance requirements can be found.

Table 4.1 lists the mission related performance inputs of each of the four possible stages: take-off, climb, cruise and descent. The models used on each of these four stages have been described and explained in Section 3.4.

Table 4.1: Mission profile performance inputs.

Stage	Input Variables							
Take-off	$steps$	h	x_{to}	μ_{dyn}	k_{st}	V_w	–	–	–
Climb	$steps$	C_{l_i}	C_{l_f}	h^{alt}_{min}	h^{alt}_{max}	V_w	RoC / δ	n	$Option_{cb}$
Cruise/Loiter	$steps$	C_l	h	V_{min}	V_{max}	V_w	n	$Range$	$Endurance$
Descent	$steps$	C_{l_i}	C_{l_f}	h^{alt}_{max}	h^{alt}_{min}	V_w	γ_{dt}	n	$Option_{dt}$

For the take-off, the input variables include the number of discretization steps, the take-off altitude (h), the available take-off distance (x_{to}), the dynamic friction coefficient (μ_{dyn}) the stall velocity coefficient (k_{st}), (which is the quotient between lift-off velocity (V_{lo}) and stall velocity (V_{st})), and the wind velocity (V_w).

For the climb stage, the input variables include the number of discretization steps, the initial wing airfoil lift coefficient (C_{l_i}), the final wing airfoil lift coefficient (C_{l_f}), the initial altitude (h^{alt}_{min}), the final altitude (h^{alt}_{max}), the wind velocity (V_w), the rate-of-climb (RoC) or the thrust setting (δ), the load factor due to bank (n) and an option for choosing between the aforementioned rate-of-climb (RoC) or thrust setting (δ).

In what concerns to the cruise stage, input variables include the number of discretization steps, the wing airfoil lift coefficient (C_l), the cruising altitude (h), the minimum aircraft velocity (V_{min}), the maximum aircraft velocity (V_{max}), the wind velocity (V_w), the load factor (n), the aircraft range and the aircraft endurance. The PARROT code will internally use the most energy consuming requirement between endurance and range and thus either a conventional cruise phase or a loiter phase can be addressed.

Finally, in what regards to the descent stage, the input variables include the number of discretization steps, the initial wing airfoil lift coefficient (C_{l_i}), the final airfoil lift coefficient (C_{l_f}), the initial altitude (h^{alt}_{max}), the final altitude (h^{alt}_{min}), the wind velocity (V_w), the descent angle (γ_{dt}), the load factor due to bank (n) and an option between either iterating the airfoil lift coefficient (C_l) or the descent angle (γ_{dt}) to make sure the energy consumption is minimized.

This code has two working modes, depending on the primary objective function chosen, which either aims at minimizing the energy mass for a maximize range/endurance (4.2.1) or to maximize the payload weight (4.2.2). In both cases, one of the most relevant design variables is unknown – the aircraft's Design Take-off Weight (DTOW). Table 4.2 summarizes the different weight philosophies depending on the working mode under consideration.

Table 4.2: Known and unknown weight fractions in the two PARROT working modes.

Mission Mode	Knowns	Unknowns
Maximum Payload	W_{sys},W_{ene}	W_{pay},W_{str}
Surveillance	W_{sys},W_{pay}	W_{ene},W_{str}

Before addressing the specific features of each of these two different working modes, their common features will be discussed.

The aerodynamics model estimates the lifting surfaces aerodynamic coefficients for every combination of Reynolds number (Re) and angle-of-attack (α) by either interpolating or extrapolating as necessary from the scattered domain of Reynolds number and angle-of-attack combinations. This is of paramount relevance as in low Reynolds numbers flows - as are the ones expected to be explored while running the PARROT code - small changes in the Reynolds number result in significant differences in what concerns to the lifting surface's airfoils aerodynamic coefficients (C_l, C_d and C_m), which impacts all the subsequent performance analysis. This aerodynamics model, including the approximate methods to estimate the 3D aerodynamic coefficients (C_L, C_D and C_M) is fully explained in Section 3.3.1.

The propulsion model (Section 3.3.2) matches the propeller and the motor or engine for a given speed and throttle setting, for every point analyzed, and provides thrust force and power consumption, enabling the user to either choose a combustion engine or an electrical motor.

Lift and drag coefficients are computed considering trimmed conditions in each and every analysis point. The user defines the lateral and longitudinal static stability margins (C_{l_β}, C_{n_β} and K_n) and the static stability model, described in Section 3.3.4 sizes the horizontal and vertical empennages and wing dihedral angle in accordance with the aforementioned settings.

The dynamic stability is not optimized but simply analyzed. As such, its model, Section 3.3.5, works independently of the working mode chosen with its single goal being to inform the designer about the vehicle dynamic behavior by providing the eigenvalues of the lateral (roll, dutch roll and spiral) and longitudinal (phugoid and short period) dynamic stability matrices and their respective response frequency, damping coefficient and time to damp to half amplitude.

4.2.1 Maximum Range/Endurance Mission

The first of the two PARROT's working modes aims at optimizing an aircraft for minimum energy consumption in a maximum range/endurance mission, which means a kind of mission where the greater the vehicle's energetic efficiency the greater can its endurance and range be.

The user may define either a specific range and/or endurance for the cruise mission stage. From these inputs, the routine will determine which of the two requirements requires more energy, and will analyze the full mission profile considering that particular range and/or endurance.

Despite providing a very simplified schematic representation of this design working mode, Figure 4.1 enables the reader to understand how the iterative procedure starts and how the DTOW is updated.

Firstly, the user defines the payload and the systems weights (W_{pay}, W_{sys}), together with the intended mission profile and the batteries or fuel specific energy (e_{spec}). Then, a first guess for the DTOW will get the routine started. A number of wingspan (b) and mean wing chord ($\bar{c}$) combinations will be generated using the full factorial sampling method, in accordance with the user settings for the number of different mean wing chords (n_{chords}), minimum and maximum mean wing chord ($\bar{c}_{min}, \bar{c}_{max}$) and different wingspans (n_{spans}), minimum and maximum wingspan (b_{min}, b_{max}). The PARROT code will then start the sequential analysis of each mission stage according to the mission profile performance targets (Table 4.1).

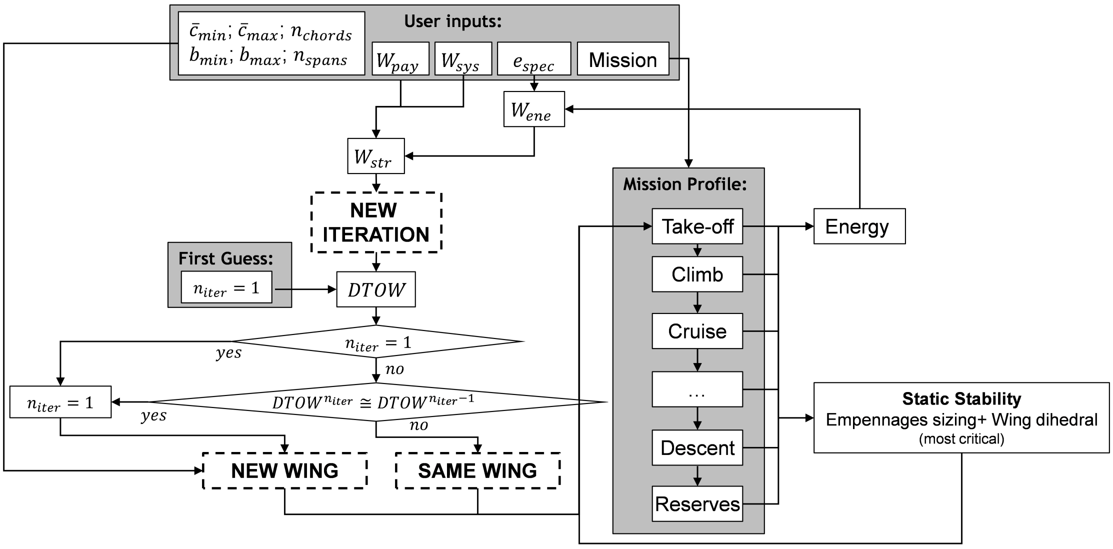


Figure 4.1: Scheme featuring the system level iterative procedure for the PARROT code in the surveillance mission mode.

Each iterative procedure starts with an initial guess for the vehicle's DTOW ($n_{iter} = 1$) and is run until the DTOW converges. Once the DTOW converges, another wing geometry (wingspan and wing mean chord combination) is analyzed.

At each iteration, the user defined mission profile is simulated and the total energy (E) required for performing such mission is known, which means that the energy weight (W_{ene}) can be computed from Equation (3.63). At this point, it is possible to re-run the structural weight model and consequently updating the DTOW accordingly so that the following iteration starts.

The static stability model (Section 3.3.4) is run for each mission stage in order to size the optimum horizontal and vertical empennages (S_{ht}, S_{vt}) for each wing geometry in accordance with the user defined static margins. If required, some wing dihedral (Γ) might be added. Finally, the empennages and dihedral considered for each wing geometry on the following iteration will correspond to the most critical solution amongst all mission stages, to make sure that the actual static stability margins are at least as good as the respective user defined requirements.

The wing incidence (i_w) will be an energy-weighted average of the optimum wing incidence of each stage - horizontal fuselage - which corresponds to the wing angle-of-attack in the respective mission stage (${\alpha_w}^{stage}$). This wing incidence calculation further contributes for reducing the overall energy consumption, by diminishing the fuselage induced drag. The weight coefficients are proportional to their energy consumption relative to the total mission energy (E^{stage}/E^{total}), as per Equation (4.1).

$$i_w = \frac{E^{to}}{E^{total}}{\alpha_w}^{to} + \frac{E^{cb}}{E^{total}}{\alpha_w}^{cb} + ... + \frac{E^{dt}}{E^{total}}{\alpha_w}^{dt} \tag{4.1}$$

Once either the DTOW converges or the maximum number of iterations is reached, the iterative procedure comes to an end and the designer can evaluate all the analysis outputs to determine the impact of varying the wing mean chord ($\bar{c}$) and/or the wingspan (b) on the most relevant design functions.

4.2.2 Maximum Payload

The second of PARROT's working modes aims at maximizing the aircraft payload for a specific mission profile and given propulsive power.

Figure 4.2 provides a simplified schematic representation of the design layout in this mode. It is possible to understand how the iterative procedure starts and how the design take-off weight (DTOW) is updated.

Firstly, the user defines the energy and the systems weights (W_{ene}, W_{sys}), together with the intended mission profile and specific energy (e_{spec}). Then, a first guess for the DTOW will get the routine started.

A number of wingspan (b) and mean wing chord ($\bar{c}$) combinations is generated using the full factorial sampling approach explained in Section 4.2.1.

Since this working mode aims at lifting the maximum possible payload, the iterative procedure is significantly different from the minimize energy working mode as far as the DTOW convergence is concerned.

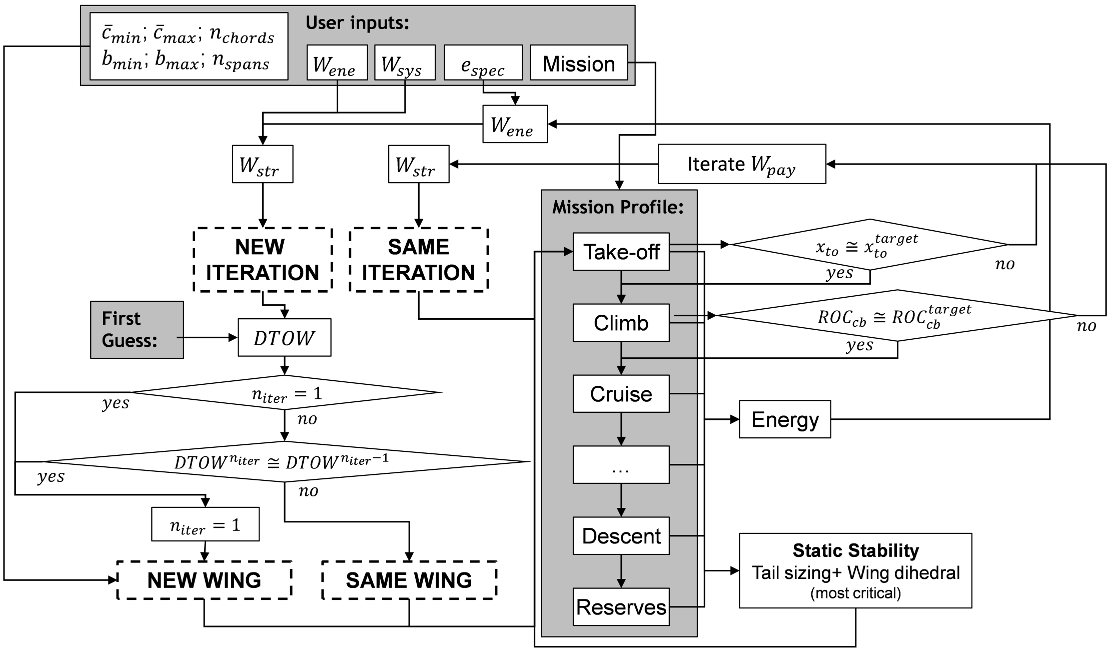


Figure 4.2: Scheme featuring the system level iterative procedure for the PARROT code in the maximize payload mode.

It is assumed that all mission profiles will start with a take-off stage followed by a climb stage, these being precisely the two most relevant stages in limiting the MTOW and consequently the maximum payload weight (W_{pay}), since the take-off stage will limit the available take-off distance, whereas the climb stage limits the rate-of-climb (RoC).

As for the take-off (Section 3.4.1), all the available distance will be used. If that happens not to be the case, the payload weight will be corrected accordingly, increased if the required take-off distance is lower than the one available and decreased otherwise.

For enhancing the payload lifted, the wing incidence (i_w) will be the optimum wing incidence for the take-off stage, which means that it equals the optimum wing angle-of-attack for the take-off run (${\alpha_w}^{to}$), as per Equation (4.2), which does however impact the fuselage drag and the energetic efficiency in the remaining mission stages.

$$i_w = {\alpha_w}^{to} \tag{4.2}$$

As for the first climb stage (Section 3.4.2.1), the user defined RoC shall be met - which can in some cases be mandatory as far as clearance of obstacles is concerned. However, if the available propulsive power is not enough to meet it, the payload weight is reduced accordingly.

If the take-off and/or the first climb stage user settings limit the vehicle DTOW, the DTOW is iterated without the overall weight convergence iteration number being incremented. As depicted in the scheme of Figure 4.2, once the DTOW meets the take-off and first climb stage requirements all the remaining mission stages are run.

Once all the mission stages have been analyzed, the energy spent (E) shall be known, which means that the energy weight (W_{ene}) can be computed from Equation (3.63). From these, it is possible to re-run the structural weight model and finally updating the DTOW accordingly so that the following weight convergence iteration can take place.

Also, after analyzing all mission stages, the static stability (Section 3.3.4) is run to size the optimum horizontal and vertical empennages (S_{ht}, S_{vt}) for each wing geometry in accordance with the user defined static margins. If required, some wing dihedral (Γ) might be added. Finally, the empennages and dihedral considered for each wing geometry on the following iteration will correspond to the most critical solution amongst all the mission profile stages, to make sure that the actual static margins are at least as good as the respective user defined settings.

Once either the DTOW converges or the maximum number of iterations is reached, the iterative procedure comes to an end and the designer can evaluate all the analysis outputs to determine the impact of varying the wing mean chord ($\bar{c}$) and/or the wingspan (b) on multiple relevant figures of merit.

4.3 Graphical User Interface

The current section presents a graphical user interface (GUI) tailored for widening the scope of possible users of the implemented PARROT methodology. While the PARROT methodology is one of the core developments and contributions of the current research work, its GUI is a side development despite enriching the overall research deliverables. Accordingly, this section provides an overlook on the most significant aspects of the developed GUI. Neither it includes significant details on the development itself nor particular instructions to guide its future users. These can be found in Appendix D, which features the PARROT GUI users' manual, where all the software GUI windows are shown and the user can see a list of all the required inputs and provided outputs.

The development of PARROT's GUI was made using the open source XFLR5 GUI, which is programmed in C++ language. The main reason for using the XFLR5 framework was the fact that it is an open source code, easy to handle and already having expedite methods for the aerodynamic analysis of airfoils (using XFOIL).

The first step was to create a new menu in XFLR5 called *Aircraft Optimization* (Figure 4.3). This is made to distinguish the mission-based aircraft design analysis from other standard XFLR5 analyses. By clicking on this new menu option, it can be found a new one called *Analysis* in which the user can choose the PARROT program.

Once the user has made the aforementioned selection, it is possible to have a general view of the PARROT code main menu, as shown in Figure 4.4. The first options are related with program inputs, with specific menus for the propulsion, systems, fuselage, aerodynamics and weight data as well as for the intended mission profile performance targets.

As the number of input parameters is relatively large, once the user has loaded all the data for the first time, it is possible to generate a ".txt" file which stores all the project data. This file can be loaded in subsequent analyses, avoiding the tiresome and repetitive task of loading all the required data each time the PARROT routine is called. It can be useful to load

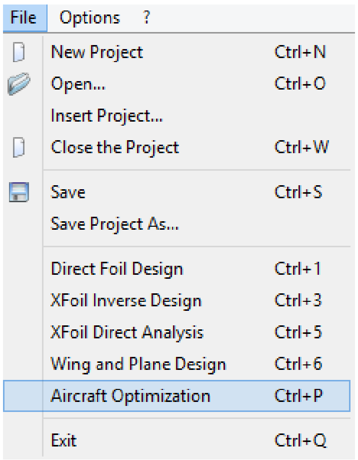
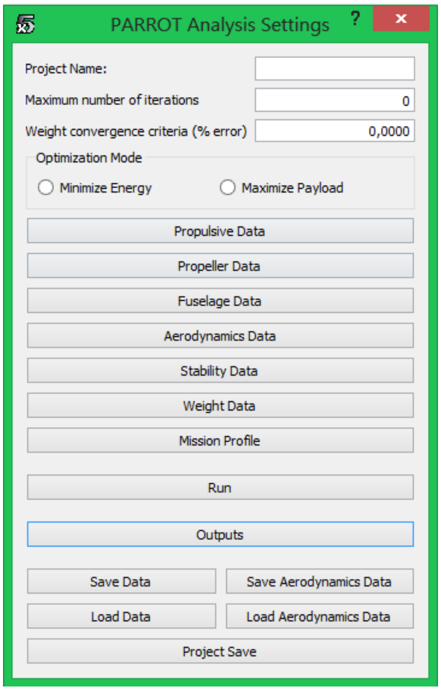

File
Options
?
New Project Ctrl+N
Open... Ctrl+O
Insert Project...
Close the Project Ctrl+W
Save Ctrl+S
Save Project As...
Direct Foil Design Ctrl+1
XFoil Inverse Design Ctrl+3
XFoil Direct Analysis Ctrl+5
Wing and Plane Design Ctrl+6
Aircraft Optimization Ctrl+P
Exit Ctrl+Q

Figure 4.3: *Aircraft Optimization* menu.


PARROT Analysis Settings

Project Name:

Maximum number of iterations 0

Weight convergence criteria (% error) 0,0000

Optimization Mode

○ Minimize Energy ○ Maximize Payload

Figure 4.4: Main interface window.

all the input data from the ".txt" file if the user wants to rerun a previously saved analysis or if only a few inputs are meant to be changed. Therefore, the user can also load the general input parameters by clicking on *Load Data*. To make this possible, every time a new analysis is made, a file named "input_parrot.txt"' is generated, which can later be run.

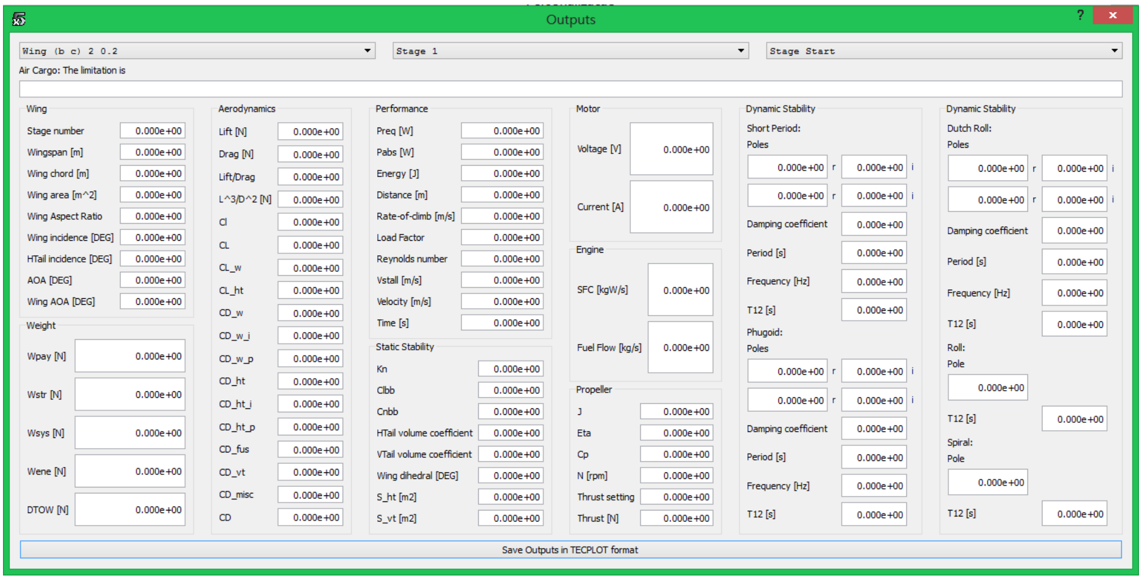
After clicking on the *Analysis* button - which will call PARROT's executable file - it is possible to visualize all the relevant outputs as functions of each flight phase and wingspan versus wing mean chord combination. The user can also save this output data in a .txt in matrix form to enable an easy generation of the respective parametric graphical representations. To have a more global view about all the inputs and outputs, it is also possible to save all the data in a specific folder with the respective project name.

On the *Aerodynamics Data* button it is possible to load the airfoils' aerodynamic coefficients, which shall be generated beforehand in XFLR5, using the *XFoil Direct Analysis* menu. For that, it is necessary to upload the airfoils coordinate files and then perform a *Batch Analysis*.

Finally, in the window *Aerodynamics Data*, the user needs to write the airfoils' names (according to the name used in the *Batch Analysis*) in the appropriate fields.

The user can also load the *Aerodynamics Data*, by directly clicking on *Load Aerodynamics Data*. The developed GUI will consecutively and respectively then ask for the wing, horizontal tail, and vertical tail airfoils' aerodynamics data files. This last option can be used provided that the files loaded follow the aerodynamics standard files layout.

The output window is shown in Figure 4.5. It enables the designer to see the numeric output of a large number of output variables for each wing geometry at the beginning and end of each mission stage. These include: the various weight fractions (systems, energy, structures and payload weight); the most relevant aerodynamic coefficients, like the wing airfoil lift coefficient, the lift as well as the total, induced and parasitic drag coefficient of the various lifting surfaces (wing, horizontal stabilizer and vertical stabilizer); the propeller performance indicators, like the propeller advance ratio, performance and power coefficient; the motor voltage (U) and electrical current (I) or the engine consumption and engine specific fuel consumption (SFC); and finally the longitudinal stability margin (K_n), the lateral stability margins (C_{l_β},C_{n_β}) and the dynamic stability poles, response frequency, damping coefficient and time to damp to half amplitude.



Wing (b c) 2 0.2 ▾
Stage 1 ▾
Stage Start ▾

Air Cargo: The limitation is

Wing
Stage number 0.000e+00
Wingspan [m] 0.000e+00
Wing chord [m] 0.000e+00
Wing area [m^2] 0.000e+00
Wing Aspect Ratio 0.000e+00
Wing incidence [DEG] 0.000e+00
HTail incidence [DEG] 0.000e+00
AOA [DEG] 0.000e+00
Wing AOA [DEG] 0.000e+00

Weight
Wpay [N] 0.000e+00
Wstr [N] 0.000e+00
Wsys [N] 0.000e+00
Wene [N] 0.000e+00
DTOW [N] 0.000e+00

Aerodynamics
Lift [N] 0.000e+00
Drag [N] 0.000e+00
Lift/Drag 0.000e+00
L^3/D^2 [N] 0.000e+00
Cl 0.000e+00
CL 0.000e+00
CL_w 0.000e+00
CL_ht 0.000e+00
CD_w 0.000e+00
CD_w_i 0.000e+00
CD_w_p 0.000e+00
CD_ht 0.000e+00
CD_ht_i 0.000e+00
CD_ht_p 0.000e+00
CD_fus 0.000e+00
CD_vt 0.000e+00
CD_misc 0.000e+00
CD 0.000e+00

Performance
Preq [W] 0.000e+00
Pabs [W] 0.000e+00
Energy [J] 0.000e+00
Distance [m] 0.000e+00
Rate-of-climb [m/s] 0.000e+00
Load Factor 0.000e+00
Reynolds number 0.000e+00
Vstall [m/s] 0.000e+00
Velocity [m/s] 0.000e+00
Time [s] 0.000e+00

Static Stability
Kn 0.000e+00
Clbb 0.000e+00
Cnbb 0.000e+00
HTail volume coefficient 0.000e+00
VTail volume coefficient 0.000e+00
Wing dihedral [DEG] 0.000e+00
S_ht [m2] 0.000e+00
S_vt [m2] 0.000e+00

Motor
Voltage [V] 0.000e+00
Current [A] 0.000e+00

Engine
SFC [kgW/s] 0.000e+00
Fuel Flow [kg/s] 0.000e+00

Propeller
J 0.000e+00
Eta 0.000e+00
Cp 0.000e+00
N [rpm] 0.000e+00
Thrust setting 0.000e+00
Thrust [N] 0.000e+00

Dynamic Stability
Short Period:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00
Phugoid:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00

Dynamic Stability
Dutch Roll:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00
Roll:
Pole
0.000e+00
T12 [s] 0.000e+00
Spiral:
Pole
0.000e+00
T12 [s] 0.000e+00

Figure 4.5: Output data interface window.

Using the PARROT graphical user interface, the output variables shown in Figure 4.5, (Section 4.3) can be stored in matrix form, as a function of the wing geometry, mission stage and the mission stage point (beginning or end) which can thereafter be used to generate the parametric plots using some external software. It is up to the user discretion to decide which are the most relevant plots to consider in each problem: either to analyze or to select or limit the design point. The generated plots can be of paramount relevance in aiding to get closer to the optimum design solution within a short time frame. Furthermore, and perhaps equally important, the assessment of the most relevant plots against each others, facilitate the designer task by helping him/her understand how the design point selection impacts other performance metrics and, in some cases, how design specifications can limit the design domain and hence the design point selection.

4.4 Case Studies

In order to illustrate the usefulness of the computational methodology presented in this chapter, two case studies are shown in subsections 4.4.1 and 4.4.2. The goal is to show the two analyses modes which the PARROT code enables - maximum payload and minimum energy, respectively. Accordingly, the first case study's goal is to lift the maximum possible payload weight within a limited runway and a predefined electric motor, propeller and battery, whereas the second aims at performing a predefined surveillance mission given a combustion engine and propeller.

4.4.1 Air Cargo Challenge 2015

The Air Cargo Challenge (ACC) is an international biannual competition targeted to the academic community. Each of the participating teams has the assignment of designing, building and flying a radio-controlled aircraft whose main goal is to lift the highest useful payload possible in a 60m runway. Furthermore, each group has to provide written and oral support to its decisions. They are ranked according to the number of points which depend on the design report, technical drawings, oral presentation and flight scores, with bonuses and penalties also being assigned based on the achievement of predefined performance metrics or the non-compliance with specific rules.

The ACC was created in 2003 by students from *Instituto Superior Técnico* - University of Lisbon, as a national competition, but its success in Portugal lead to the first European edition being held in 2007, with a continuous growth of the number of participating teams and represented countries ever since (Fig. 4.6). Since 2011, the competition is also open to contenders from outside of Europe.

Figure 4.6: Flight of University of Beira Interior's model (the winner of ACC 2011) in Stuttgart, Germany.

The regulations of the ACC 2015 ([106]) - hosted by the University of Stuttgart - have been the ones adopted in the current case study. The most relevant design specifications are summarized in the forthcoming paragraphs.

4.4.1.1 Design Specifications

The ACC 2015 competition regulations establishes that the flight competition objective function is dependent on both the payload mass lifted (m_{pay}) and the number of legs (n_{legs}) flown in ($120s$). For a regular flight, the flight competition score will be calculated in accordance with:

$$Points = 2m_{pay}(n_{legs} + 3) \tag{4.3}$$

In this context, a regular flight means a valid take-off (within the available take-off distance), and a valid landing (where the aircraft comes to a complete stop within the available distance in a single piece).

The most relevant design specifications of the ACC 2015 are summarized in Table 4.3.

Table 4.3: Air Cargo Challenge 2015 design specifications.

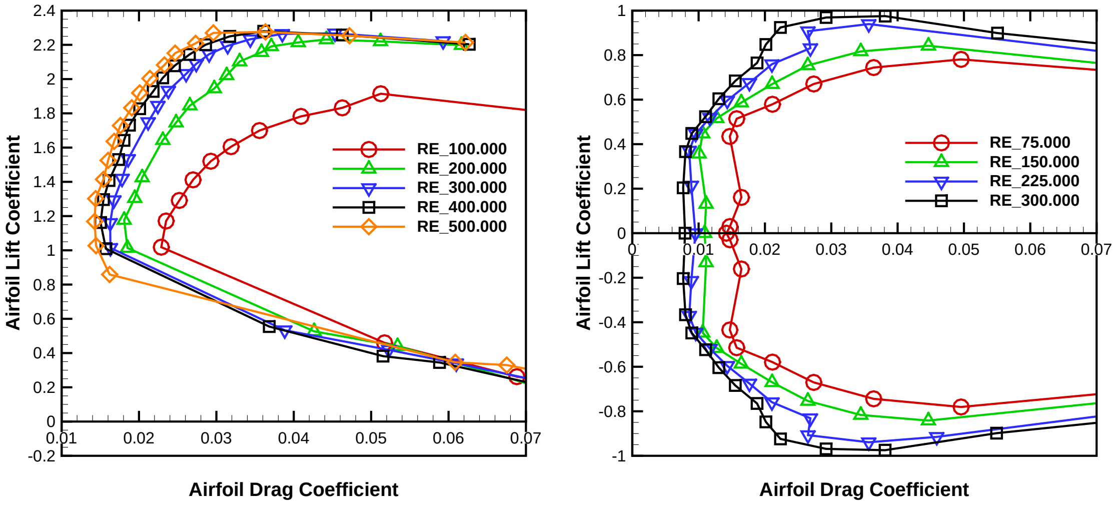
Constraint	Value
Motor	AXI Gold 2826/10
Propeller size	$13'' \times 7''$
Battery	Up to 3 cells in series and the product of maximum continuous discharge rate times the capacity has to be at least 45A
Maximum Take-off Distance	60m
Aircraft Dimensions	Limited to a 2.5 m side square
Transportation Box Outer Dimensions	1×0.5×0.4m

Besides the design specifications listed in Table 4.3, a significant number of inputs is required for feeding the parametric design analysis code, which means that it is up to the designer to define them.

One of the most important aspects in the study of aircraft flying at low Reynolds numbers is the wing airfoil choice. The wing airfoil chosen for this case study was the Selig 1223, which is the most widely used airfoil in former editions of the ACC, because of its high lift capabilities at low Reynolds numbers. The selected wing airfoil lift coefficient is ($C_l = 0.9$), because it is the lowest lift coefficient for which the airfoil performance is still not significantly jeopardized (adequate C_l/C_d). The airfoil chosen for both the horizontal and vertical stabilizers was the NACA 0009, which is also a common choice for the empennages. The airfoils drag polars are shown in Figure 4.7a and 4.7b.

It is worthwhile to mention that the user can also use the PARROT code to choose the most suitable airfoil for a given mission. Accordingly, once all the inputs have been defined, the user can run several simulations with two or more airfoils keeping all the remaining inputs constant in order to benchmark them in terms of a given objective function.

Once all the remaining disciplinary inputs are known, and before running the PARROT code the mission profile shall be defined. According to the architecture of the PARROT code, there are four generic mission stages: take-off, climb, cruise and descent.



(a) Selig 1223 drag polar.

(b) Naca 0009 drag polar.

Figure 4.7: Aircraft airfoils drag polar.

The current mission profile consists of a take-off, a climb from take-off altitude up to the cruise altitude, a cruise stage composed of the scoring legs and a descent back to the take-off altitude, as shown in Figure 4.8.

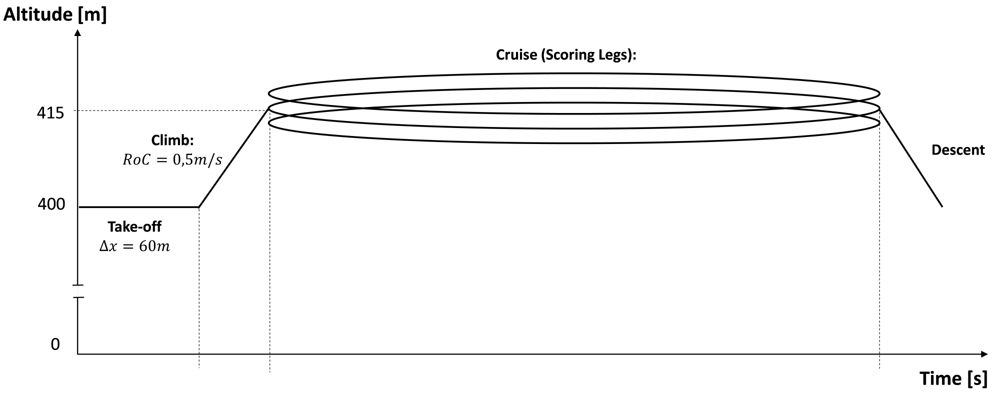


Figure 4.8: Air Cargo Challenge 2015 mission profile.

The cruise stage will alternate between straight and banked leveled flight, as shown in Figure 4.9 and finally a descent towards ground level.

It is assumed that one leg is composed of a leveled straight flight for $70m$ plus a leveled turn of $180°$ at a bank angle $\phi = 45°$.

It is assumed that each $(100m)$ is composed of $(70m)$ of straight leveled flight and two coordinated turns with a turn radius $(R > 15m)$ to make sure that each $(100m)$ leg is met. From Equation (3.106), it is possible to determine the minimum cruise velocity required for such turn radius.

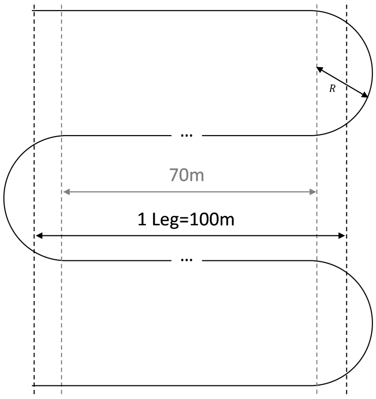


Figure 4.9: Top view schematic representation of the flight path during the 120s legs.

$$tan(\phi) = \frac{V^2}{gR} \tag{4.4}$$

Accordingly, the minimum velocity is $V_{min} = \sqrt{gR_{min}tan(\phi)}$. Once the acceleration of gravity $g = 9.81m/s$, the minimum radius $R_{min} = 15m$ and the bank angle $\phi = 45^\circ$ are known, it is possible to calculate the minimum velocity in the sustained turn cruise stage, $V_{min} = 12.13m/s$.

Another important variable is the air density, which is assumed to vary with altitude in accordance with the international standard atmosphere (ISA) [107]. Since the ACC 2015 has taken place near Stuttgart, Germany, where the local altitude above sea level varies in the following range $200m < h < 400m$, it has been assumed that the take-off and landing altitude equals $h = 400m$. Furthermore, it is assumed that the cruise flight will occur at an altitude of $15m$ above the take-off altitude. The most relevant mission profile inputs are summarized in Table 4.4.

Table 4.4: Air Cargo Challenge 2015 mission profile.

Stage	**Input Variables**								
Take-off	$steps$	h $[m]$	x_{to} $[m]$	μ_{dyn}	k_{st}	V_w $[m/s]$			
	200	400	60	0.08	1.2	0	-	-	-
Climb	$steps$	C_{l_i}	C_{l_f}	h^{alt}_{min} $[m]$	h^{alt}_{max} $[m]$	V_w $[m/s]$	RoC $[m/s]$	n	$Option_{cb}$
	100	1.1	0.9	400	415	0	0.5	1.0	2
Cruise 1	$steps$	C_l	h $[m]$	V_{min} $[m/s]$	V_{max} $[m/s]$	V_w $[m/s]$	n	$Range$ $[km]$	$Endurance$ $[h]$
	10	0.9	415	11.0	40.0	0	1	0.84	0
Cruise 2	$steps$	C_l	h $[m]$	V_{min} $[m/s]$	V_{max} $[m/s]$	V_w $[m/s]$	n	$Range$ $[km]$	$Endurance$ $[h]$
	10	0.9	415	11.0	40.0	0	1.414	0.876	0
Descent	$steps$	C_{l_i}	C_{l_f}	h^{alt}_{max} $[m]$	h^{alt}_{min} $[m]$	V_w $[m/s]$	n	γ_{dt} $[°]$	$Option_{dt}$
	50	1.2	1.2	415	400	0	1.0	4.0	2

Prior to running the PARROT code, besides all the already discussed inputs definition, it is important to choose an appropriate domain of analysis. Not only is this choice important to avoid unfeasible design solutions, but also to foster an improved computational efficiency by eliminating some variable combinations that are beforehand known not to meet the design constraints or to deliver poor performance. Considering the PARROT architecture, it is of particular relevance to decide the wing mean chord and wingspan ranges that aim to be considered in this parametric analysis.

Since the aircraft has to fit within a ($2.5 \times 2.5m$) square, its maximum wingspan is limited to about ($b_{max} = 3.5m$). As for the wing mean chord, it has been limited to ($\bar{c}_{max} = 0.45m$) because the wing planform shape is not duly optimized otherwise, since a very low taper ratio ($\lambda = \frac{c_{tip}}{c_{root}}$) would be required, which would impact the Oswald efficiency factor, and thereafter the wing aerodynamic performance. The Oswald factor considered is ($e_0 = 1.0$) - where an optimized planform shape and twist distribution is assumed. Furthermore, this limit allows the wing panels to fit in the transportation box. Finally, as the largest wingspans and wing mean chords were expected to deliver the best performances, it has been decided that the lower boundaries of these two variables would stand on ($\bar{c}_{min} = 0.30m$) and ($b_{min} = 3.0m$). For each of these variables 6 intervals have been analyzed, which means that a total of 36 wing layouts (wing mean chord versus wingspan) have been studied.

4.4.1.2 Results and Discussion

The most important results are summarized in the plots of Figures 4.10 through 4.11, where the variation of the most relevant performance metrics are plotted against the design variables (wing mean chord and wingspan). Other design variables can be studied, like the wing airfoil lift coefficient or the propeller pitch just by re-running the analysis and only changing one of these variables. It is then possible to compare the respective objective function parametric plots, thus giving a further step beyond the simply selecting the best wing planform for the mission profile under analysis.

Figure 4.10c shows how the structural weight varies with the wingspan versus wing mean chord combination. As expected, the structural weight increases with the wing area. Figure 4.10b shows that the best wing layout is $\bar{c} = 0.42m$; $b = 3.5m$. This is because the wings with the same wingspan and greater wing mean chord will not be able to meet the minimum RoC of $0.5m/s$ specified for climbing, although they could lift more payload in the available $60m$ runway. The same reasoning can be used to justify Figure 4.10a, which features the payload weight. This plot is the one that is more closely related with the ACC'15 objective function.

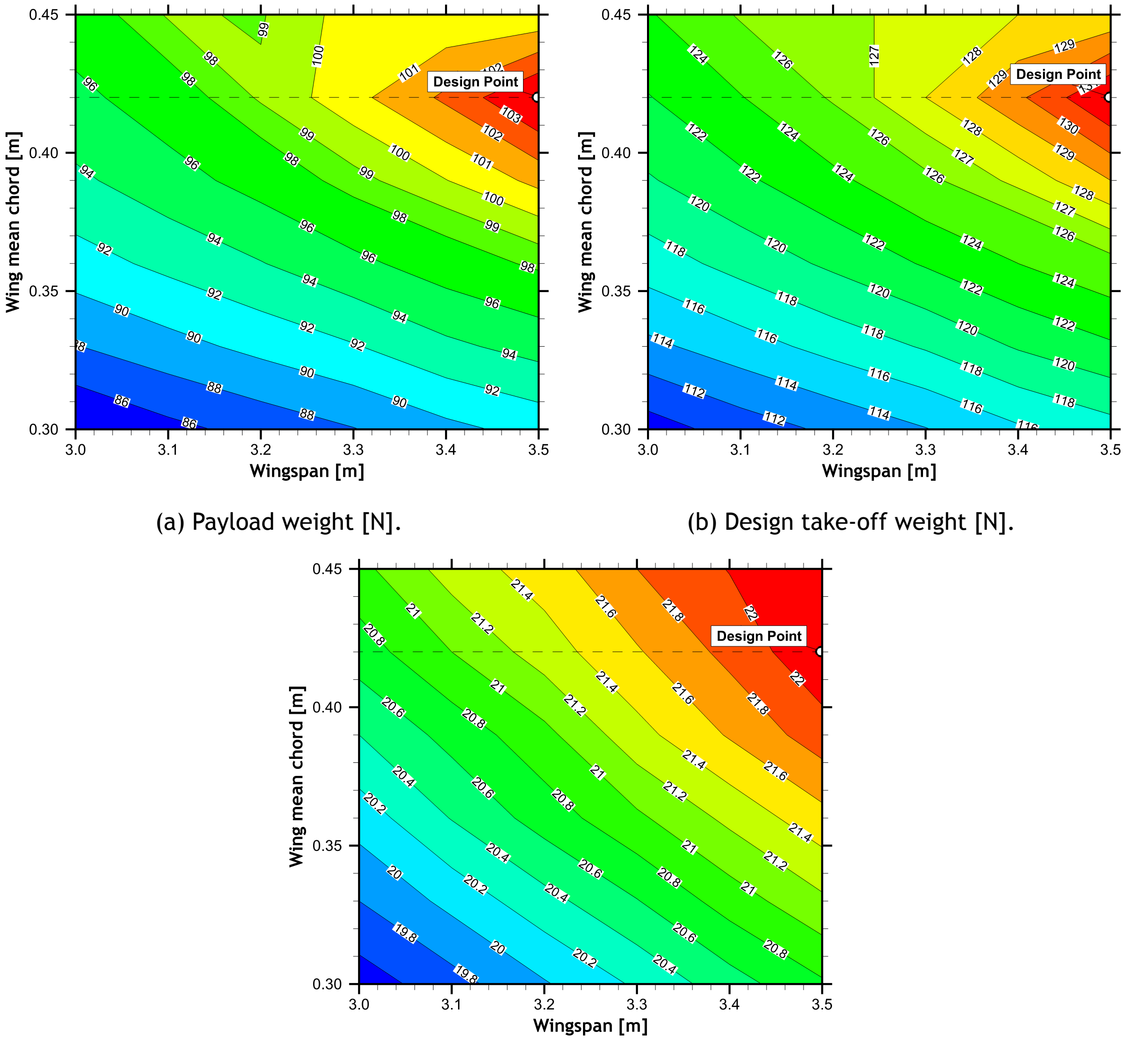
As a matter of fact, the competition's objective function is to lift the highest payload and perform the maximum number of legs in two minutes $120s$, as per Equation 4.3. If one fixes the airfoil lift coefficient of the cruise stage - as it has been done to reduce the wing's parasite drag coefficient without putting the wing airfoil performance at risk – the greater the vehicle's wing loading W/S, the greater will be the velocity and therefore the number of legs performed, which means that the two objectives (payload weight and number of legs) are slightly contradictory because the higher wing loadings occur for the smaller wings and the higher payloads tend to occur for the larger wings.

The most relevant weight components are shown in the plots of Figure 4.10. Figure 4.10a in particular features the payload weight W_{pay}, which is one of the objective function's components - along with the number of legs flown in $120s$. It can be seen that the best performing wing is the $\bar{c} = 0.42m$, $b = 3.5m$, which corresponds to the maximum possible wingspan, in accordance with the competition regulations. Larger wings are not able to meet the same rate-of-climb requirement without a decrease in their DTOW (Figure 4.10b), with their higher structural weight (Figure 4.10c) coming at the expense of the payload weight lifted. The design point is thus chosen to maximize the design payload weight lifted.

Furthermore, Figure 4.10c shows an increase of the structural weight with the wing area. Although higher aspect ratios are expected to provide greater energetic efficiency, in this analysis mode, the PARROT code single purpose is to make sure to maximize the payload lifted, and therefore, notwithstanding the aforementioned effect of the rate-of-climb requirement, the greater the wing area, the more useful payload it can lift.

Nevertheless, as it can be seen in Figure 4.11, the number of legs that are possible to perform within $120s$ in the domain of wing spans and mean chords selected is the same. Accordingly, for scoring purposes, only an integer number of completed legs can be considered. It should be noted that the computation of the number of legs has been made as if this was a continuous variable, which is not the case since only an integer discrete number of solutions is possible for scoring purposes. From the analysis of Figure 4.11, it is apparent that the maximum number of complete legs flown is the same regardless of the wing geometry - within the domain under consideration - and is equal to 12 legs.

It is clear that the payload weight will determine the best wing layout from a scoring viewpoint. The best wing layout $\bar{c} = 0.42m$; $b = 3.5m$ can also be seen in Figure 4.12, which shows the total flight score (Equation 4.3) as a function of the payload mass lifted and of the



(a) Payload weight [N].

(b) Design take-off weight [N].

(c) Structure weight [N].

Figure 4.10: Performance metrics as a function of wingspan and wing mean chord.

integer number of legs performed (12 for all the analyzed wings). Would the total number of complete legs flown be different among the wing geometries considered and Figure 4.11 would also be a decision driver plot in what concerns the design point selection.

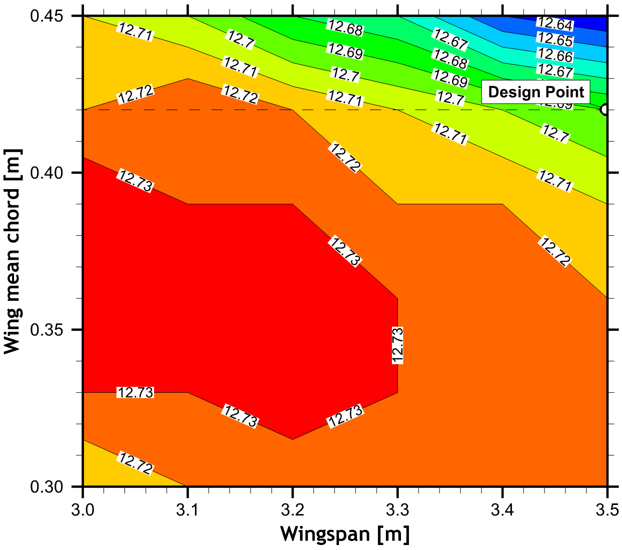


Figure 4.11: Number of legs flown in 120s.

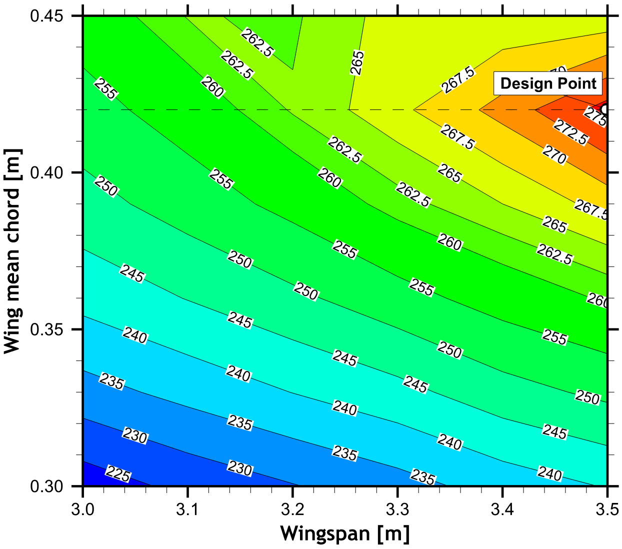
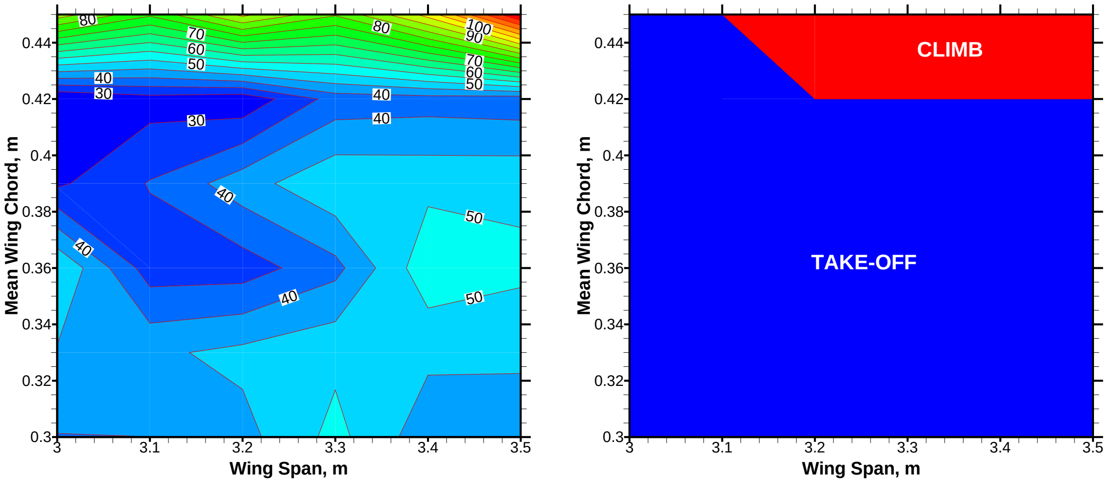


Figure 4.12: ACC score objective function as a function of wingspan and wing mean chord.

It has been shown how the parametric plots, namely the payload weight, can be used to identify the optimum design point. The selection of the design point was made solely by looking at the payload weight because the number of legs is the same within the domain analyzed. The other weight fraction plots can be used to understand how the design point affects each of them, which can help the designer in questing for minimizing possible negative impacts of a given choice, in situations different than the current case study, where there is more flexibility in terms of the design specifications.

The PARROT running time [s] in the Maximum Payload working mode is shown Figure 4.13a, while the payload limiting mission stage as a function of the wing mean chord and wingspan is shown in Figure 4.13b. It is interesting to understand that despite all wings having a running time of the same order of magnitude, the wings that have the climb stage as the payload weight as limiting condition are the ones with the highest running times, as expected, and can be easily



(a) PARROT running time [s] as a function of wingspan and wing mean chord.

(b) Maximum Payload limitation (take-off or climb) as a function of wingspan and wing mean chord.

Figure 4.13: Air Cargo Challenge analysis - PARROT Performance.

understood from Figure 4.2.

4.4.2 Maximum Range/Endurance Mission

The PARROT's second case study is defined in such a way that important features that have not been used on the Air Cargo Challenge 2015 study (Subsection 4.4.1) are duly appreciated. Accordingly, the analysis mode chosen is the minimize energy, which suits best a surveillance mission profile. Furthermore, contrarily to the already presented case study where an electric motor was used, this case study uses a combustion engine.

This case study's mission profile is composed of a take-off, a climb to cruise altitude, a high speed cruise stage with headwind, a low speed loiter stage, a high speed cruise stage with tailwind and a final descent towards ground level. This mission profile is shown in Figure 4.14.

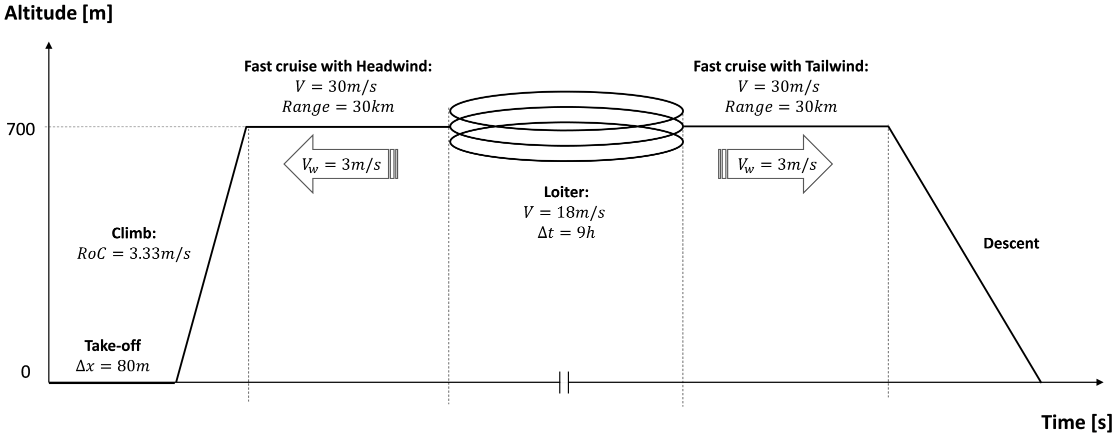


Figure 4.14: Surveillance mission profile.

4.4.2.1 Design Specifications

Contrarily to the former case study, this surveillance mission does not have previously defined mandatory design specifications. As such, the designer is entitled more flexibility, which means that it will eventually be possible to have a more enhanced design configuration. However, regardless of the design analysis problem at hands, it is important to understand that before reaching the optimum design configuration, the designer will always have to impose a number of constraints other than the mission profile performance targets. In the current study, these include the engine, the propeller and the fuel's specific energy selection, among others.

The most relevant design specifications for this surveillance mission are summarized in Table 4.5. The engine was selected based on the expected required power, whereas the propeller was carefully chosen to agree with the engine specifications and operating velocity. The other two mention-worth specifications refer to the imposed wing aspect ratio (Λ) and the wing stall velocity (V_{st}) limits.

The wing aspect ratio requirement ($\Lambda \leq 15$) is imposed to fall within the validity of the structural weight estimates, which have a limited range of applicability in terms of the wing aspect ratio, as the correlations presented in Section 3.3.3.1 have been developed for medium aspect ratio wings. In addition, the reference structure weight used for the structural weight estimates is itself a medium aspect ratio wing and therefore it would be unrealistic to provide estimates for high aspect-ratio wings. Lastly, it is worthwhile to note the structural weight problems that such high aspect ratio wing can cause. Indeed, such a wing could potentially contribute to aeroelastic problems, which the disciplinary physical models in use fail to consider.

Besides, the imposed wing aspect ratio limit, the stall velocity requirement ($V_{st} \leq 13.5m/s$) is imposed for flight safety reasons.

Table 4.5: Most relevant design specifications for the surveillance mission.

Constraint	Value
Engine	3W-55i
Propeller size	$22'' \times 12''$
Wing aspect ratio	≤ 15
Wing stall velocity	$\leq 13.5m/s$

In the PARROT code, the aforementioned specifications, most notably regarding the engine, propeller and fuel are considered parameters and not design variables. Therefore, it does not enable an assessment of such variables. Nonetheless, its intrinsic design analysis facilities can be used ad-hoc via, e.g. running several analysis with everything constant but the propeller diameter to determine the optimum propeller diameter for the current mission profile, which would consist of another parametric study. The same approach would apply for virtually all the remaining design parameters, which means that the user can actually leverage the PARROT code far beyond its standard analyses features. This can also be used for choosing the most suitable airfoil for a given mission.

In any case, it is up to the designer to define the initial parameters that enable the iterative procedure to get started at a feasible design point. As such, the propeller (pitch and diameter) should match the engine chosen, the engine should match the required propulsive force, the wing and stabilizers airfoils and so forth.

Besides the design specifications listed in Table 4.5, a significant number of inputs is required for feeding the parametric design analysis code. One of the most important aspects in the study of aircraft flying at low Reynolds numbers is the wing airfoil choice.

The wing airfoil chosen for this case study - OGIII - has been optimized in software specifically tailored for airfoil shape optimization, and it is a well performing airfoil in the range of Reynolds numbers and wing loadings it is expected to operate. The airfoil chosen for both the horizontal and vertical stabilizers was the NACA 0009, which is also a common choice for these lifting surfaces, likewise was made in the previous case study.

In accordance with the previous considerations, the most relevant mission profile inputs are summarized in Table 4.6.

Table 4.6: Surveillance mission profile requirements.

Stage	Input Variables								
Take-off	$steps$	h $[m]$	x_{to} $[m]$	[illegible]	[illegible]	V_w $[m/s]$			
	50	0	80	0.08	1.2	0	-	-	-
Climb	$steps$	C	C	h_{min}^{alt} $[m]$	h_{max}^{alt} $[m]$	V_w $[m/s]$	RoC $[m/s]$	n	$Option$
	25	1.1	0.9	0	700	0	0.5	1.0	2
Cruise 1	$steps$	C_l	h $[m]$	V_{min} $[m/s]$	V_{max} $[m/s]$	V_w $[m/s]$	[illegible]	$Range$ $[km]$	$Endurance$ $[h]$
	30	0.4	700	30	50	3	1	30	0
Cruise 2	$steps$	C_l	h $[m]$	V_{min} $[m/s]$	V_{max} $[m/s]$	V_w $[m/s]$	[illegible]	$Range$ $[km]$	$Endurance$ $[h]$
	30	0.9	700	18	50	0	1.0642	0	9
Cruise 3	$steps$	C	h $[m]$	V_{min} $[m/s]$	V_{max} $[m/s]$	V_w $[m/s]$	[illegible]	$Range$ $[km]$	$Endurance$ $[h]$
	30	0.4	700	30	50	-3	1.0	30	0
Descent	$steps$	C	C	h_{max}^{alt} $[m]$	h_{min}^{alt} $[m]$	V_w $[m/s]$	[illegible]	γ_{dt} $[°]$	$Option$
	50	1.2	1.2	700	0	0	1.0	4.0	2

Prior to running the PARROT code, by eliminating some variable combinations that are beforehand known not to meet the design constraints one can avoid unfeasible design solutions. In addition, these boundaries shall be chosen in such a way that the best solution does not lie outside them. If that would be the case, then the domain of analysis should be corrected and the analyses should be run again.

In light of the PARROT's architecture, it is of particular relevance to decide the wing mean chord ($\bar{c}$) and wingspan (b) ranges that aim to be considered in this parametric study. Contrarily to the former case study, whose design specifications were set by the Air Cargo Challenge regulations, in this case study the design constraints shall be established by the designer. The domain of wing mean chords and wingspans analyzed is ($0.20m \leq \bar{c} \leq 0.30m$) and ($2.0m \leq b \leq 5.0m$) respectively. These values have been based on prior experience for the range of vehicle operating velocities and also for the expected design take-off weight order of magnitude.

In addition to this domain of analysis, two additional design constraints have been added. The acceptable solutions are limited to wing geometries respecting a maximum stall velocity ($V_{st} \leq 13.5m/s$) and a maximum aspect ratio ($\Lambda \leq 15$). Whereas the maximum stall velocity requirement makes sure that the aircraft is able to fly safely at velocities higher than the considered threshold, the maximum admissible aspect ratio contributes to avoid potential aeroelastic problems, which the disciplinary models in use do not assess.

4.4.2.2 Results and Discussion

The PARROT's objective function in the minimize energy analysis mode is the energy weight (W_{ene}) or the energy (E). However, as these two variables are directly proportional, they can be interchangeably used. Figure 4.15 features the energy weight as a function of the wing mean chord ($\bar{c}$) and the wingspan (b). It is noticeable that the lowest energy consumptions correspond to the wings with the greatest aspect ratio, within the considered domain of wing mean chords ($0.2m < \bar{c} < 0.3m$) and wingspans ($2m < b < 5m$). It is obvious that the best solution from an energetic efficiency viewpoint is found somewhere close to the bottom right corner of the plot of Figure 4.15.

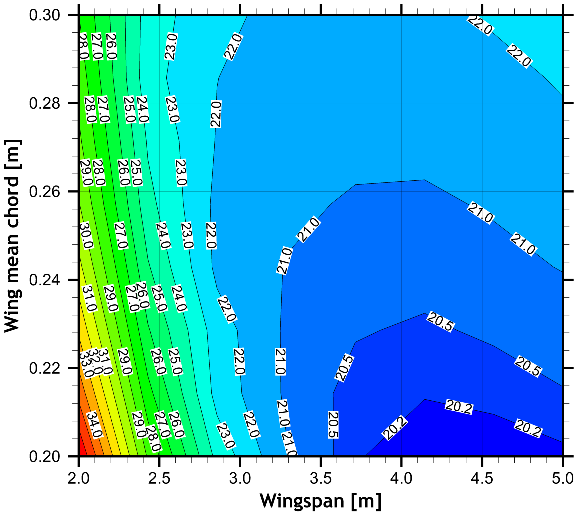
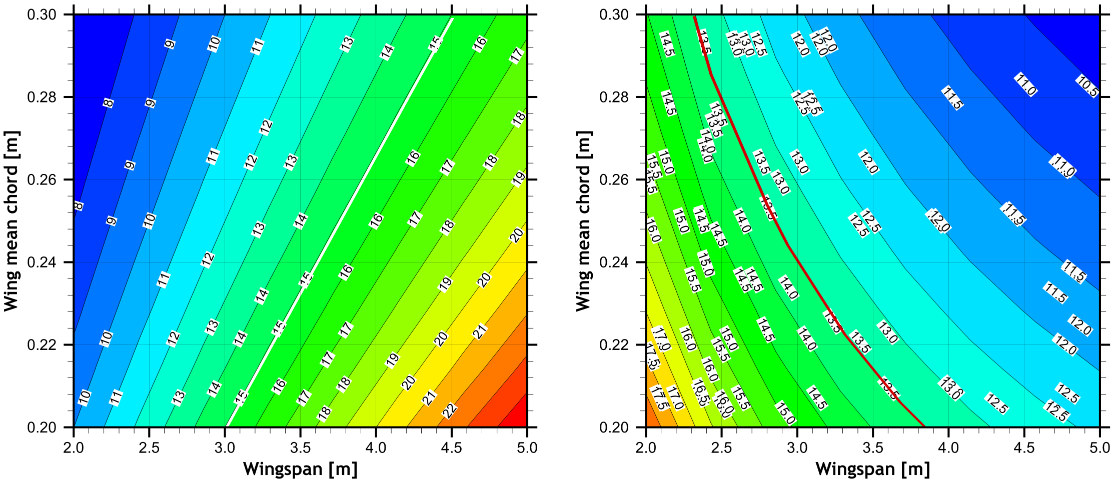


Figure 4.15: Energy weight [N] as a function of wingspan and wing mean chord.

However, the initial specifications limit the acceptable solutions to wing geometries respecting a maximum stall velocity ($V_{st} \leq 13.5m/s$) and a maximum aspect ratio ($\Lambda_w \leq 15$). These two limits are shown in Figures 4.16a and 4.16b, respectively. Figure 4.16a features the maximum acceptable aspect ratio highlighted in white color, whereas Figure 4.16b shows the maximum acceptable stall velocity highlighted in red color.



(a) Wing aspect ratio.

(b) Wing stall velocity [m/s].

Figure 4.16: Limits' specifications.

Under these requirements, the design point is the one identified on Figure 4.17, which is the one that corresponds to the lowest energy consumption respecting the maximum stall velocity and maximum aspect ratio limitations.

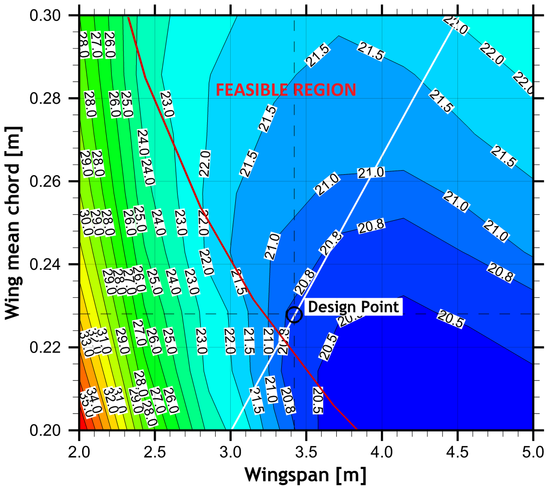
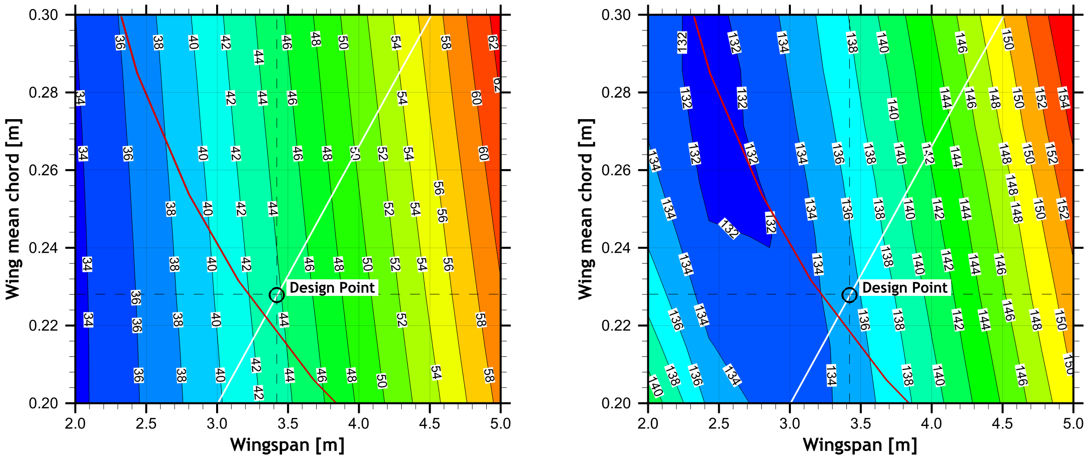


Figure 4.17: Energy weight [N] as a function of wingspan and wing mean chord, including the wing aspect ratio and the wing stall velocity boundaries and the respective design point.

In this analysis mode, the payload weight (W_{pay}) and the systems weight (W_{sys}) are user inputs, and therefore independent of the wing geometry under analysis. The structure weight (W_{str}) is shown in Figure 4.18a. It can be perceived that both larger wing areas and greater aspect ratios contribute to an increase in the vehicle's structural weight. The design take-off weight ($DTOW$) is shown in Figure 4.18b. It is interesting to see that while it is proportional do the structure weight at wingspans greater than ($3m$), for lower wingspans the energy weight becomes more relevant (W_{ene}), Figure 4.17.

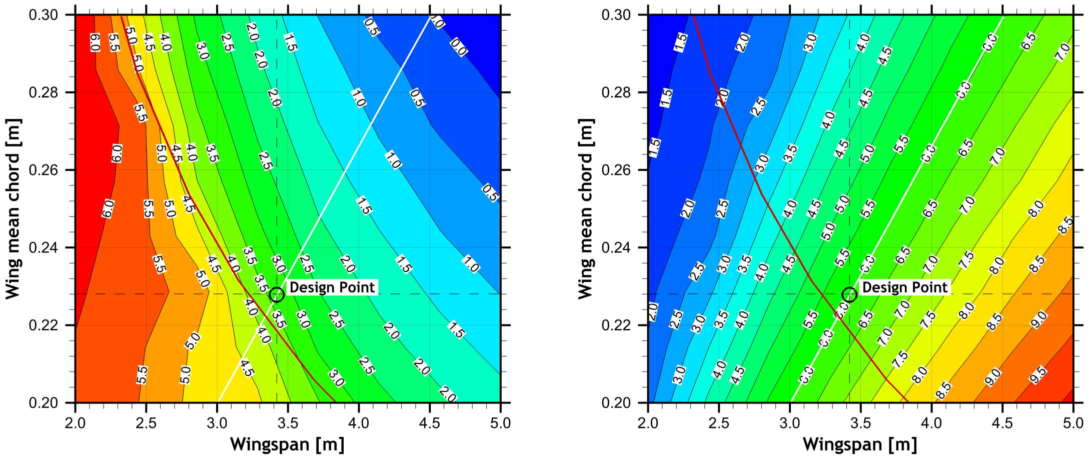


(a) Structure weight [N].

(b) Design take-off weight [N].

Figure 4.18: Structure weight and design take-off weight.

The wing incidence (i_w), computed as an energy weighted average of the wing angle of attack in each different mission stage is shown in Figure 4.19a. Conversely the wing dihedral, required for lateral static stability is shown in Figure 4.19b.

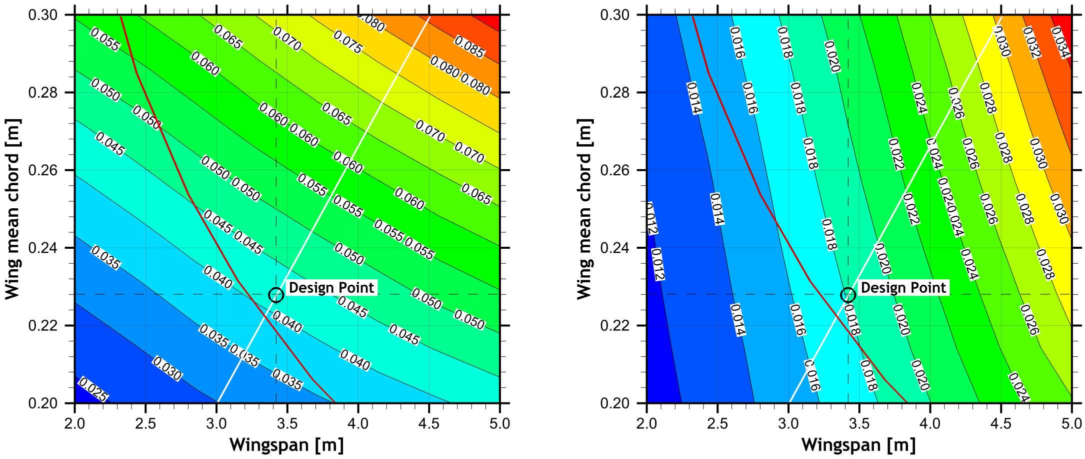


(a) Wing incidence [°].

(b) Wing dihedral angle [°].

Figure 4.19: Wing incidence and dihedral angles.

The horizontal stabilizer area (S_{ht}) is shown in Figure 4.20a, while the vertical stabilizer area (S_{vt}) is shown in Figure 4.20b. As expected, the greater the wing area, the greater are the empennages size for analogous static stability margins.



(a) Horizontal stabilizer area $[m^2]$.

(b) Vertical stabilizer area $[m^2]$.

Figure 4.20: Empennages surfaces' areas.

Lastly, the better adequacy of a combustion engine for a surveillance mission can be witnessed. With a similar DTOW to the one of case study 1 (Air Cargo Challenge 2015 4.2.2), this aircraft is able to perform a loiter of $(9h)$ and two cruise stages of $(30km)$ each.

It has been shown how the parametric plots, namely the energy weight plot, can be used to identifying the optimum design point, limiting the maximum acceptable aspect ratio and stall velocity. The other weight fraction plots can be used to understand how the design point will affect each of them, which can help the designer in questing for minimizing possible negative impacts of a given choice.

It is worthwhile to discuss that the PARROT analysis code has got a relevant limitation which is not being able to handle unfeasible solutions. While the tail is automatically sized for the given solution based on the user specified static margins, thus posing no problem, a badly selected engine/motor or propeller can produce unfeasible solutions and get the analysis stuck. It is therefore important that these parameter are somehow sized beforehand to avoid feasibility issues.

4.5 Concluding Remarks

This chapter has presented the PARROT methodology tailored for the preliminary design of UAVs. This approach uses the wingspan and the wing mean chord as its primary study parameters. This methodology makes use of the low fidelity models presented in Chapter 2 for the aerodynamics, weight, propulsion, and static and dynamic stability. As such, this MDO methodology makes sure that the aircraft configuration is statically stable, both laterally and longitudinally.

A large number of outputs featuring the most prominent performance metrics, (e.g. the payload weight (W_{pay}), the energy weight (W_{ene}) and the design take-off weight $(DTOW)$), among many others, can be plotted against the wing mean chord and wingspan. Thus, the PARROT code can not only guide the designer's decision early in the design process, but also help him/her understand how the wing geometry can impact performance metrics other than the objective function. The developed methodology enables the user to perform parametric studies involving variables other than the wing mean chord and the wingspan. To do this, all the user has to do is to successively re-run a previous simulation changing a single parameter to assess its performance impact.

The graphical user interface of the PARROT architecture makes this multidisciplinary code interesting for a significantly wider number of users, while making the interaction fairly more user friendly than otherwise. It profits from its interaction with the standard XFLR5 facilities, namely by using the aerodynamic analysis of airfoils to feed the PARROT studies. A free version of the PARROT code shall soon be released online to make it available for the academic and engineering community, along with a user's manual for dealing with its GUI, Appendix D.

The first case study presented in this chapter aimed at showcasing the PARROT methodology in the maximum payload analysis mode using an electric motor. The Air Cargo Challenge 2015 contest design regulations have been used for such purpose. The presented parametric plots are in fair agreement with the actual competition winner's performance - including the payload lifted and number of legs flown in $(120s)$, despite the low fidelity models in use, which means that both the physical models and the analysis methodology are working well. This is even more apparent if one considers that the teams with the highest flight scores have thoroughly optimized their aircrafts' design.

The second case study aimed at highlighting the PARROT code results in the surveillance mission mode. A second goal was to adopt a combustion engine propulsive source. The results have shown that the combustion engine suits better an extended surveillance mission than an electric solution. It has been shown how the optimum design point can be found by imposing several design requirements, including upper boundaries for the aspect ratio and stall velocity. It has been concluded that the minimum energy consumption neither corresponds to the lowest design take-off weight nor to the lowest structural weight.

An output file with all the computed data can be saved after the PARROT code is run. Therefore, the user can not only see them in the graphical user interface (GUI) choosing the wing, mission stage and whether the beginning or the end of the selected stage, but also generate plots like the ones presented for the two case studies featured - using some external software. This makes possible to evaluate the different variables behavior in all the design domain. It is up to the user discretion to decide which are the most relevant plots to consider in each problem: either to analyze or to select or limit the design point.

Chapter 5

Multilevel Design Optimization

5.1 Introduction

This PhD Thesis main goal is the development of two innovative mission-based design optimization methodologies, including the mainstream disciplines of aeronautical design: aerodynamics, propulsion, mass distribution, stability and performance. While the disciplinary models that underlie both of them have been thoroughly described in Chapter 2, the PARROT methodology has been presented in Chapter 3. The current Chapter is devoted to a second methodology, named **M**ul**T**ilevel design **OP**timization (MTOP). In contrast to PARROT, MTOP is not based on a parametric study optimization approach, but rather on a multilevel optimization architecture. Accordingly, the final output is not an analysis that can be used for optimization purposes but the optimization result itself, which is the combination of design variables that better suits the objective or objectives subjected to the design constraints. The actual optimum is reached provided the design constraints and objectives generate a feasible aircraft and either the design space is convex or the optimization algorithm obviates the local extrema problem.

MTOP makes use of the Enhanced Collaborative Optimization (ECO) multilevel architecture, together with a gradient-based optimization algorithm. Having the energy consumption minimization for a specified mission profile as its goal, this architecture results in an unconstrained system problem, whose single goal is to assure compatibility between subspaces and duly constrained subspace level problems with the goal of minimizing the energy consumption.

One of the most distinguishing features of this methodology is that instead of each subspace representing the traditional design disciplines (e.g. aerodynamics, structures, stability, etc), the author has chosen to make a different subspace out of each flight stage (e.g. take-off, climb, cruise, etc). The main reason behind this approach has been the inclusion of morphing technologies as part of the optimization process. Thus, as opposed to the PARROT code, the MTOP enables the assessment and optimization of Variable Span Wing (VSW), Variable Camber Flap (VCF) and Variable Pitch Propeller (VPP). In addition, the designer can benchmark these solutions against the baseline conventional configuration and indeed evaluate different morphing combinations for each different mission in a rather straightforward approach. Moreover, it also includes a tailored graphical user interface which further enhances its user friendliness.

The methodology design variables include the wing mean chord, wingspan, propeller diameter, pitch propeller, flap chord fraction and flap deflection.

5.2 Multilevel Architectures

5.2.1 Introduction

After a comprehensive performance benchmark study of the already developed optimization architectures (refer to Section 2.1), it has been decided to implement the most recent version of Collaborative Optimization (CO), which has been named as Enhanced Collaborative Optimization (ECO) - presented by Roth and Kroo (2008) [57, 58, 72].

The original version of Collaborative Optimization (CO) provides a significant degree of independence between subspaces. On the one hand, this enables disciplinary experts to run their own codes using discipline preferred optimization techniques. On the other hand, each subspace has very little information of the actions and preferences of the other subspaces. Information is only changed indirectly through the system level targets.

ECO has shown to be effective in reducing the number of discipline analyses when compared to CO [57, 72] and to compare favorably with the methods against which it has been benchmarked, after the works of Kodiyalam (1998) [59], Kodiyalam and Yuan (2000) [108], de Wit and van Keulen (2007) [44], Yi et al (2008) [42] and Roth and Kroo (2008) [58].

5.2.2 Enhanced Collaborative Optimization

The original ECO architecture has been presented in Subsection 2.1.3.4. In the current Subsection it is detailed including two modifications to the original formulation. The first resides on a weighting coefficient (λ^i_{weight}) which makes it possible to assign a different weight to each subspace as far as the system level compatibility is concerned, whilst the second refers to a dynamic compatibility parameter (λ^*_C), which avoids a premature compatibility among subspaces.

System Level Optimization

In ECO, the system level optimization is simply an unconstrained minimization problem. The global objective function (i.e. the ultimate design goal) is not present in the system level objective. The system level single goal is to achieve compatibility between subspaces, as formulated in Equation (5.1).

$$\text{Min } J_{sys} = \sum_{i=1}^{n} \left[\lambda^i_{weight} \sum_{j=1}^{n_{s\,i}} (z_j - x_{s_j}{}^{*(i)})^2 \right] \tag{5.1}$$

with respect to List of design variables
subject to No constraints

, where z are the system level targets for shared variables, x^* represents the subspace best attempt to match the system level targets, (subject to local constraints), λ^i_{weight} is a weighting coefficient which will be different for each subspace, n is the number of subspaces and n_{si} is the number of shared variables in subspace i.

In the ECO architecture, the subspaces are responsible for most of the design decisions. Their objective function includes three components: a local objective function (which generically depends on shared global and local variables), a compatibility term (with a quadratic measure of compatibility) and a feasibility term (with a set of slack variables), as formulated in Equation (5.2).

$$\text{Min } J_i = f(x_S, x_L) + \lambda_C^* \sum_{k=1}^{n_{s_i}} (x_{s_k} - z_k)^2 + \lambda_F \sum_{j=1}^{n} \left[\sum_{k=1}^{n_{g_j}} (s_{g_k}^{(j)}) + \sum_{l=1}^{n_{h_j}} (s_{h_l}^{(j)} + e_l^{(j)}) \right] \tag{5.2}$$

with respect to List of design variables

subject to

$$\begin{aligned} & g_k^{(i)}(x_S, x_L) \geq 0, k = 1...n_{g_i} \\ & h_l^{(i)}(x_S, x_L) = 0, l = 1...n_{h_i} \\ & \tilde{g}_k^{(j)}(x_s) + s_{g_k}^{(j)} \geq 0, j = 1...n, k = 1...n_{g_j}, j \neq i \\ & \tilde{h}_l^{(j)}(x_s) + s_{h_l}^{(j)} - e_l^{(j)} = 0, j = 1...n, l = 1...n_{h_j}, j \neq i \\ & s_g, s_h, e \geq 0 \end{aligned}$$

, where f is the local objective function, λ_C^* is a dynamic compatibility parameter, λ_F is a feasibility parameter, n_{g_j} is the number of inequality constraints in subspace j, n_{h_j} is the number of equality constraints in subspace j, $\tilde{g}_k^{(j)}$ is a linear model of the k^{th} inequality constraints in subspace j, $\tilde{h}_l^{(j)}$ is a linear model of the l^{th} equality constraints in subspace j.

The third term in Equation (5.2) is called the feasibility term and is used to make sure that the equality and inequality constraints imposed in each subspace are known in other subspaces. Hence, this term represents the single cross-flow of data between the subspaces. The first part of the term refers to inequality constraints whilst the second refers to the equality constraints.

There have been two significant modifications to the original ECO formulation by Roth and Kroo (2008) [57], the first being the use of a weighting coefficient (λ_{weight}^i) on the system level optimization problem, Equation (5.1) which enables weighting each subspace in a different way. As a result, the system level optimization objective function is not the arithmetic average of the difference between the subspaces best attempt to match the system level targets (x^*) and the system level targets themselves (z). Conversely, the system level optimization function will take into consideration the relative weight of each subspace for the minimization of the overall objective function, resulting in a weighted average of the squared difference between the subspaces best attempt to match the system level targets (x^*) and the system level targets themselves (z).

Another major difference from the works presented to date Roth and Kroo (2008) [57, 58] and Xiao (2010) [109], is that the compatibility parameter is a dynamic quantity. This is because it has been found that an early compatibility between subspaces may jeopardize the actual multilevel optimization problem. In order to mitigate this shortcoming on the first system level iterations each disciplinary (subspace) optimization is fully independent ($\lambda_C^* = 0$) and only after a threshold number of system level iterations does the compatibility term become active ($\lambda_C^* > 0$), with its magnitude growing linearly with the number of system level iterations. This means that in the first system level iterations the compatibility term is null while it gradually becomes the dominant term as compatibility among subspaces is sought. This means that on the last system level iterations the subspace objective function single goal is to make shared variables compatible.

As it is common in engineering applications, the design variables may have diverse ranges, with quantities having different orders of magnitude. In order to tackle such problem, a normalization method to avoid the fluctuations caused by sensitive design variables has been adopted, Equation (5.3).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.3}$$

For computing the numerical gradients, first-order forward differences have been used:

$$\frac{\partial f(x)}{\partial x} = \frac{f(x+h) - f(x)}{h} \tag{5.4}$$

, where h is an small increment of the design variable.

5.3 Optimization Algorithms

Given that the main goals of this research have been the multilevel optimization architecture implementation and the physical models development the authors have decided not to generate their own optimization algorithm, relying on some previous developments in the domain of gradient-based ones.

In the current research study, one aim at using an optimization algorithm with the goal to minimizing the energy consumption for an arbitrary mission profile with respect to several aircraft design variables. One expect the objective function (energy) to be well behaved and its absolute minimum to be easily captured using a gradient-based optimization strategy.

The first optimization algorithm implemented is called DONLP2 and is based on a sequential equality constrained quadratic programming method by Spellucci (1998) [110, 111], whereas the second optimization algorithm considered was the FFSQP, which consists of a set of FORTRAN subroutines for the minimization of the maximum of a set of smooth objective functions subject to general smooth constraints due to Zhou et al (1992) [112]. While the DONLP2 makes it possible not to respect the constraints on the earliest iterations, the FFSQP does not. Because most of the optimization problems solved are believed to have well behaved domains (largely convex), local minima were not an issue and therefore the comparison among the two optimization algorithms is less interesting and out of the scope of the current research. Hence, the FFSQP has been used to solve the benchmarking and the optimization case studies presented in the forthcoming sections.

It is important to note that the numerical models must comply with laws of physics. Thus, not only must the initial guess lead to a feasible design point, as the subsequent steps have to lead to physically realistic conditions. Hence, it is paramount to carefully select the design variables initial point and design variable ranges.

5.4 Benchmarking Case Studies

The current Section is devoted to the benchmark of the Enhanced Collaborative Optimization (ECO) multilevel optimization algorithm presented in Subsection 5.2.2. To do that, the results of two case studies have been compared with the ones available in the literature. In the first Subsection 5.4.1, the Rosenbrock problem has been implemented and compared against the results of Roth et al (2008) [58], whereas the second Subsection 5.4.2 featured an analytical test case whose results have been benchmarked with the ones presented by the same authors [57] for this particular problem.

5.4.1 Rosenbrock Problem

The Rosenbrock function is a classic function often used as a test case for optimization algorithms and is thus of obvious interest for the Enhanced Collaborative Optimization (ECO) multilevel architecture. Consider the solution of the Rosenbrock problem using ECO. The problem can be decomposed into a bi-level structure with a system level problem and two subspaces. This has been done by Roth et al [58] and is used hereinafter for benchmarking the ECO architecture. Accordingly, the function to be minimized at the system level optimization is the one shown in Equation 5.5.

$$\text{Min } f = 100x_2^2 + x_3^2 \tag{5.5}$$

$$\begin{array}{ll} \text{with respect to} & x_1, x_2, x_3, x_4 \\ \text{subject to} & h_1 = x_2 - (x_4 - x_1^2) = 0 \\ & h_2 = x_3 - (1 - x_1) = 0 \\ & g_1 = 0.575 - \sqrt{x_1^2 + x_4^2} \geq 0 \\ \text{bounds} & x_1 \geq 0.01 \end{array}$$

In line with the ECO architecture, the system level contains no local constraints and the minimization function is the difference between the subspaces best attempt to match the system level targets ($x^{(i)}$) and the system level targets themselves (z).

System Level

$$\begin{aligned} \text{Min } J_{sys} = & \left(z_1 - x_1^{(1)}\right)^2 + \left(z_2 - x_2^{(1)}\right)^2 + \\ & \left(z_1 - x_1^{(2)}\right)^2 + \left(z_2 - x_2^{(2)}\right)^2 + \left(z_3 - x_3^{(2)}\right)^2 \end{aligned} \tag{5.6}$$

$$\begin{array}{ll} \text{with respect to} & z_1, z_2, z_3 \\ \text{subject to} & \text{no constraints} \end{array}$$

Let (J_{sys}) be the system level objective function, (λ_C) be the compatibility parameter, (λ_F) the feasibility parameter, (s_1) and (e_1) be slack variables and (g_i) and (h_i) be equality and inequality constraints, respectively and ($\tilde{g}_i$) and ($\tilde{h}_i$) be equality and inequality constraint copies, respectively.

In ECO, the subspaces retain most of the responsibility for guiding the optimization process, seeking to minimize the global objective while ensuring their own compatibility with the other(s) subspace(s). This high level of control is enabled by modeling the global objective within the subspace objective, and by modeling the effect of constraints from other subspaces.

Subspace 1

$$\text{Min } J_{ss1} = 100{x_2}^2 + \lambda_C\left[(x_1 - z_1)^2 + (x_2 - z_2)^2\right] + \lambda_F(s_1 + e_1) \tag{5.7}$$

$$\begin{aligned}
\text{with respect to} \quad & x = [x_1, x_2, x_L, s_1, e_1] \\
\text{subject to} \quad & g_1 = 0.575 - \sqrt{{x_1}^2 + {x_L}^2} \geq 0 \\
& h_1 = x_2 - (x_L - {x_1}^2) = 0 \\
& \tilde{h}_1 = h_2(z) + \left(\frac{dh_2}{dx_1}\right)(x_1 - z_1) + \left(\frac{dh_2}{dx_2}\right)(x_2 - z_2) + s_1 - e_1 = 0 \\
& s_1, e_1 \geq 0
\end{aligned}$$

Subspace 2

$$\begin{aligned}
\text{Min } J_{ss2} =& 100{x_2}^2 + {x_3}^2 + \lambda_C\left[(x_1 - z_1)^2 + (x_2 - z_2)^2 + (x_3 - z_3)^2\right] \\
& + \lambda_F(s_1 + s_2 + e_1)
\end{aligned} \tag{5.8}$$

$$\begin{aligned}
\text{with respect to} \quad & x = [x_1, x_2, x_3, s_1, s_2, e_1] \\
\text{subject to} \quad & h_2 = x_3 - (1 - x_1) = 0 \\
& \tilde{h}_1 = h_1(z) + \left(\frac{dh_1}{dx_1}\right)(x_1 - z_1) + \left(\frac{dh_1}{dx_2}\right)(x_2 - z_2) + s_1 - e_1 = 0 \\
& \tilde{g}_1 = g_1(z) + \left(\frac{dg_1}{dx_1}\right)(x_1 - z_1) + \left(\frac{dg_1}{dx_2}\right)(x_2 - z_2) + s_2 \geq 0 \\
& s_1, s_2, e_1 \geq 0
\end{aligned}$$

As noted in the ECO method description, a wide range of penalty parameter values yield convergence. In the solution of the current case study, Roth et al have used the following values: ($\lambda_C = 0.1$), ($\lambda_F = 5.0$). These two values have been used for the corresponding variables in the current case study for the sake of reinforcing the benchmark between the multilevel optimization architectures in use.

Two convergence criteria have been implemented. One limiting the number of system level iterations and another establishing the maximum acceptable relative difference between the function on the $i - 1^{th}$ iteration and on the i^{th} iteration:

$$\frac{f^{i}_{global} - f^{i-1}_{global}}{f^{i-1}_{global}} < \tilde{e} \tag{5.9}$$

, where $\tilde{e}$ is the relative acceptable error, which is user set and f_{global} is the global objective function. The code stops when the first of these two conditions is met. In the current case study the maximum number of system level iterations was set to 100, whereas the relative error was set to $\tilde{e} = 10 \times 10^{-5}$.

Accordingly, the concept of dynamic compatibility parameter introduced by Albuquerque et al (2015) [100] has not been used in this validation of the ECO methodology. The solution results can be benchmarked with the ones obtained by Roth and Kroo for the same case study and multilevel optimization architectures. These are summarized in Table 5.1.

Table 5.1: Solutions of the Rosenbrock Problem.

Variable	Implemented ECO	Benchmarking ECO [58]	Absolute deviation [%]
x_1	0.5140	0.5126	0.273
x_2	-0.0066	-0.0023	187
x_3	0.4861	0.4874	0.267
Objective	0.241	0.238	1.26

The results shown in Table 5.1 show that the absolute minimum of the Rosenbrock function was found and there is a very good agreement between the different optimized variables - exhibiting a maximum absolute percentage deviation of (1.11%) with the exception of variable (x_2), which can be partially justified by the different optimization algorithms in use. Furthermore, the little relevance of this variable for the whole objective function can be witnessed by the small deviation between the objective function value (1.26%), for such a large deviation of the referred variable. Figure 5.1 shows how the design variables and objective function evolve with the number of system level iterations.

One could argue that the implemented formulation results should be exact since the function under analysis and the respective gradients are analytically computed. The difference between the obtained results and the benchmarking ECO is possibly due to the use of numerical derivatives by Roth and Kroo (2008) [58].

5.4.2 Analytical Test Case

Roth et al (2008) [57] have presented an analytical test case solved using the ECO formulation. For that, they have decomposed the problem into a bi-level structure with a system level problem and two subspaces. Their results have also been used for benchmarking the implemented ECO architecture. Accordingly, optimization problem to be solved is formulated in standard form as shown in Equation 5.10.

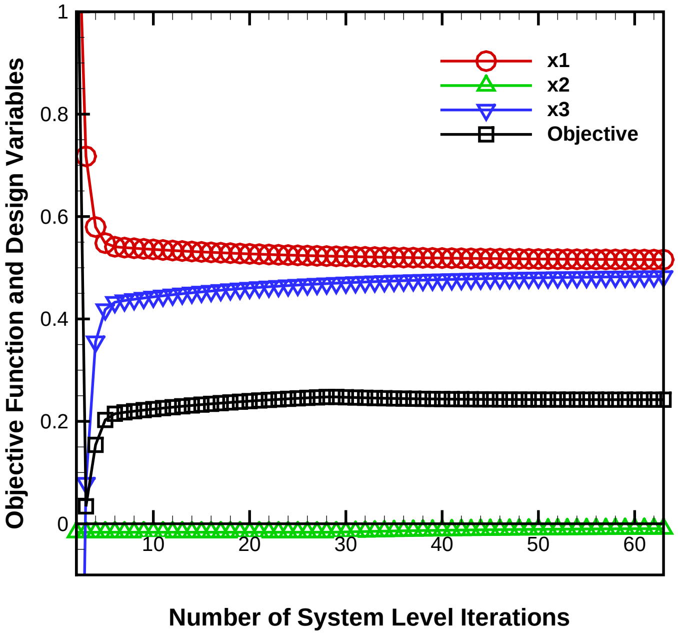


Figure 5.1: Iteration history up to convergence (Rosenbrock case study).

$$\text{Min } f = x_2^2 + x_3 + x_4 + e^{-x_5} \tag{5.10}$$

$$\begin{array}{cl}
\text{with respect to} & x_1, x_2, x_3, x_4, x_5 \\
\text{subject to} & x_4/8 - 1 \geq 0 \\
 & 1 - x_5/24 \geq 0 \\
\text{bounds} & 0 \leq x \leq 10 \\
\text{dependent variables} & x_4 = x_1^2 + x_2 + x_3 - 0.2x_5 \\
 & x_5 = \sqrt{x_4} + x_1 + x_3
\end{array}$$

Given the generic problem formulation, the Enhanced Collaborative Optimization multi-level architecture problem formulation can now be presented, starting from the system level optimization.

System Level

$$J_{sys} = \left[(z_1 - x_1^{(1)})^2 + (z_2 - x_2^{(1)})^2 + (z_3 - x_3^{(1)})^2 + (z_4 - x_4^{(1)})^2 + (z_5 - x_5^{(1)})^2\right] + \tag{5.11}$$

$$+ \left[(z_1 - x_1^{(2)})^2 + (z_3 - x_3^{(2)})^2 + (z_4 - x_4^{(2)})^2 + (z_5 - x_5^{(2)})^2\right] \tag{5.12}$$

$$\begin{array}{cl}
\text{with respect to} & x = [z_1, z_2, z_3, z_4, z_5] \\
\text{subject to} & \text{No constraints}
\end{array}$$

Subspace 1

Let (J_{ss1}) be the subspace 1 objective function.

$$\text{Min } J_{ss1} = \left[x_2^2 + x_3 + x_4 + e^{-x_5}\right] + \tag{5.13}$$

$$+\lambda_C\left[(x_1-z_1)^2+(x_2-z_2)^2+(x_3-z_3)^2+(x_4-z_4)^2+(x_5-z_5)^2\right]+\lambda_F[e_1]$$

$$\begin{aligned}
\text{with respect to}\quad & x=[x_1,x_2,x_3,x_4,x_5,e_1,e_2]\\
\text{subject to}\quad & g_1=\tfrac{x_4}{8}-1\geq 0\\
& h_1=x_4-y_1=0\\
& \tilde{g}_2=g_2(z)+\textstyle\sum_{i=1}^{5}\left[\tfrac{dg_2}{dx_i}(x_i-z_i)\right]+e_1\geq 0, i\neq 2\\
& \tilde{h}_2=h_2(z)+\textstyle\sum_{i=1}^{5}\left[\tfrac{dh_2}{dx_i}(x_i-z_i)\right]+e_2\geq 0, i\neq 2
\end{aligned}$$

Subspace 2

Let (J_{ss2}) be the subspace 2 objective function.

$$\text{Min } J_{ss2} = \left[x_3 + x_4 + e^{-x_5}\right] + \tag{5.14}$$
$$+\lambda_C\left[(x_1-z_1)^2+(x_2-z_2)^2+(x_3-z_3)^2+(x_4-z_4)^2+(x_5-z_5)^2\right]+\lambda_F[e_1+e_2]$$

$$\begin{aligned}
\text{with respect to}\quad & x=[x_1,x_3,x_4,x_5,e_1,e_2]\\
\text{subject to}\quad & g_2=1-\tfrac{x_5}{24}\geq 0\\
& h_2=x_5-y_2=0\\
& \tilde{g}_1=g_1(z)+\textstyle\sum_{i=1}^{5}\left[\tfrac{dg_1}{dx_i}(x_i-z_i)\right]+e_1\geq 0, i\neq 2\\
& \tilde{h}_1=h_1(z)+\textstyle\sum_{i=1}^{5}\left[\tfrac{dh_1}{dx_i}(x_i-z_i)\right]+e_2\geq 0, i\neq 2
\end{aligned}$$

As noted in the ECO architecture description, a wide range of penalty parameter values yield convergence. In the solution of the current case study, Roth et al have used the following values: ($\lambda_C = 0.1$), ($\lambda_F = 10.0$). These two values have been used for the corresponding variables in the current case study for the sake of reinforcing the benchmark between the multilevel optimization architectures is use.

The same convergence criteria used on the Rosenbrock benchmark study have been used in the current case study. The maximum number of system level iterations was set to 100, whereas the relative error was set to $\tilde{e} = 10 \times 10^{-8}$.

Accordingly, the concept of dynamic compatibility parameter introduced by Albuquerque et al (2015) [100] has not been used in this validation of the ECO methodology, because it had not been used by the benchmarking publication. The solution results can be benchmarked with the ones obtained by Roth et al (2008) [57] for the same case study and multilevel optimization architectures. These are summarized in Table 5.2.

The results shown in Table 5.2 show that the absolute minimum of the analytical test

Table 5.2: Solutions of the Analytical Test Case.

Variable	Implemented ECO	Analytical Solution [57]
x_1	3.0284	3.0284
x_2	0.0000	0.0000
x_3	0.0000	0.0000
x_4	8.0000	8.0000
x_5	5.8569	5.8569
Objective	8.00286	8.00286

case under analysis has been determined. Not only did the algorithm obviate local minima, but there is a perfect agreement between the obtained values and the references', for the numeric precision in use. This can probably be apportioned to the fact that this analytical study function exhibits a spatial behavior which makes it more easily captured by the optimization algorithm used. Figure 5.2 shows how the design variables and objective function evolve with the number of system level iterations.

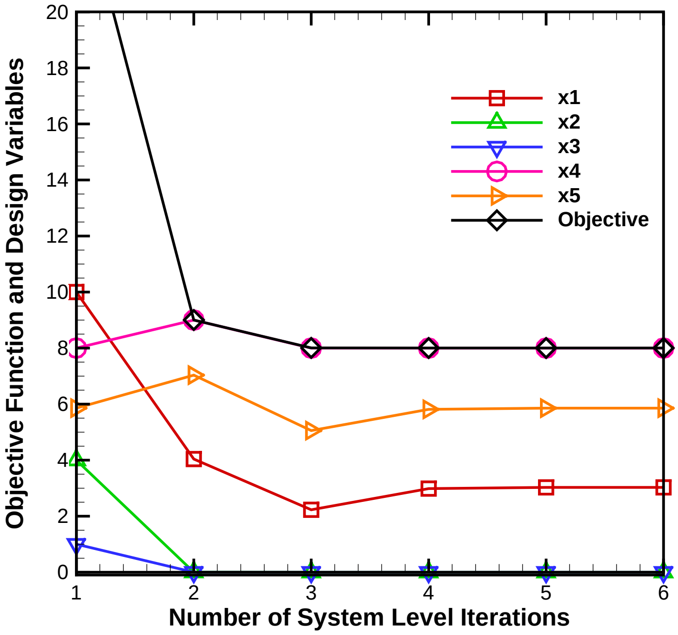


Figure 5.2: Iteration history up to convergence (analytical case study).

5.5 Optimization Methodology

The first question that may arise when a multilevel optimization architecture methodology is implemented lies in defining the disciplines (subspaces) under consideration. As a matter of fact, a discipline can represent nearly any part, sub-part, partition or sub-domain of the overall system. In the case of aircraft design, it is common to consider the traditional design disciplines, like: aerodynamics, structures, performance, stability, controls, etc, as the optimization disciplines of the multilevel subspaces. However that is not necessarily the case.

As aforementioned, one of the core goals of the current research project is the assessment of morphing technologies, resulting that some design variables will change at each mission stage. Therefore, assigning a different subspace (i.e. disciplinary optimization) to each different mission stage (e.g. take-off, climb, cruise, etc) came as a corollary.

In order to avoid any possible confusion between the concepts of discipline and subspace, which are commonly interchangeably used within the context of multilevel optimization, our definition of subspace refers to our subspace optimizations (e.g. take-off, climb, cruise, etc), whereas our definition of discipline refers to the traditional disciplinary models (e.g. aerodynamics, performance, propulsion, etc.).

A schematic representation of the MTOP framework devised in the current study is presented in Figure 5.3. It makes it possible to understand how the system level, subspace level, optimization algorithm and disciplinary analysis relate with each other.

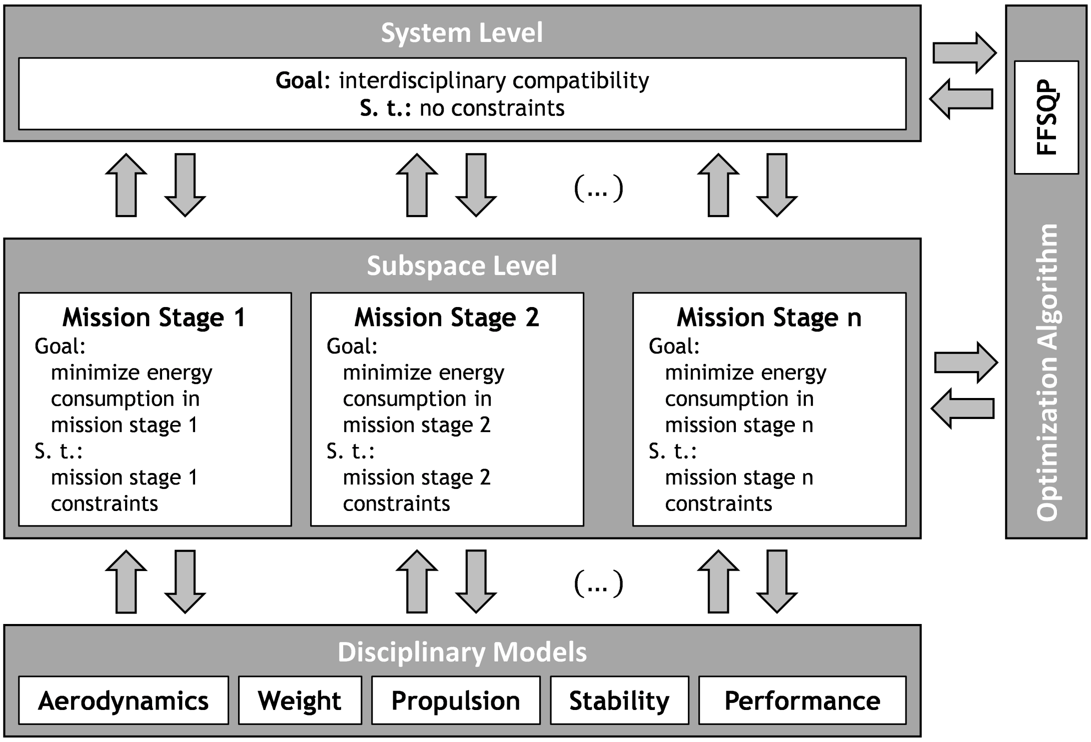

System Level
Goal: interdisciplinary compatibility
S. t.: no constraints
(...)
Subspace Level
Mission Stage 1
Goal:
minimize energy consumption in mission stage 1
S. t.:
mission stage 1 constraints
Mission Stage 2
Goal:
minimize energy consumption in mission stage 2
S. t.:
mission stage 2 constraints
Mission Stage n
Goal:
minimize energy consumption in mission stage n
S. t.:
mission stage n constraints
(...)
Disciplinary Models
Aerodynamics
Weight
Propulsion
Stability
Performance
FFSQP
Optimization Algorithm

Figure 5.3: Schematic representation of the MTOP framework.

Since part of this research work has been developed within the European 7^{th} Framework Programme CHANGE consortium [113], the evaluation of the profitability of morphing solutions is one of its main goals. In order to assess morphing concepts in the context of MDO - and among the many morphing concepts developed to date Barbarino et al (2008) [78] - the assessment of variable span wing (VSW), variable camber flap (VCF) and variable pitch propeller (VPP) concepts have been selected. These choices have been primarily based on the fact that these are amongst the morphing solutions which deliver the greatest performance impacts. Likewise, these morphing technologies have been used in the CHANGE consortium prototype and it is therefore of particular relevance to provide a numerical assessment of their worthiness.

Furthermore, a successful in-house development of a variable span wing using a telescopic mechanism by Metrinho et al (2011) [114] and by Felicio et al (2011) citefelicio and the recent implementation of a weight estimate correlation Cunha (2014) [101] for variable span wings have further fostered the interest in assessing the VSW solution.

The other morphing concept considered allows for the wing airfoil camber variation and it is achieved by mounting a flap on the inboard wing section using a continuous flap. This will enable the setting of an optimized wing flap deflection (δ_f) for each mission stage, further enhancing the vehicle's aerodynamic performance or reducing energy requirements.

The setting of an optimum wingspan and/or variable camber flap for each mission stage is expected to enlarge the aircraft's flight envelope as well as to enable an enhanced aerodynamic performance, depending on the mission profile under consideration.

These morphing concepts can either work in isolation or together - with this research goal including the performance assessment of all three possibilities, i.e. VSW, VCF and both VSW and VCF. However, it should be noted that if the two concepts are implemented on the same wing, they will limit each other in the way that the variable camber wing airfoil can only be built on the inboard portion of the wing that is fully independent from the variable span wing telescopic mechanism retraction. Accordingly, the maximum span variation would happen for a flap-less wing, whereas the maximum flap span would mean that a variable span wing could not be built. Thus, this should be kept in mind in the design of such a wing. Figure 5.4 features a graphical representation of the implied geometrical constraints.

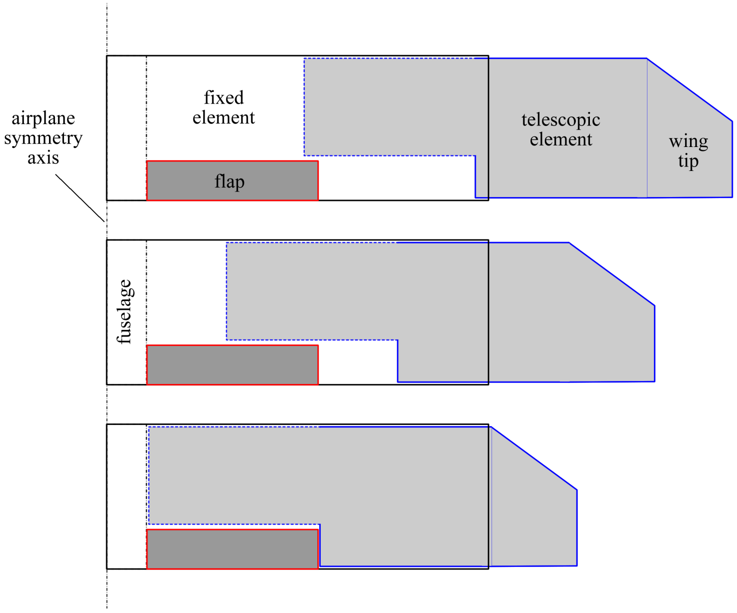


Figure 5.4: Graphical representation of a telescopic VSW with a VCF wingspan section which uses a continuous flap for different telescopic wing extensions.

The implementation of morphing solutions, which are expected - by definition - to allow a different and optimized wing shape for each mission stage, has supported the decision of adopting a novel multilevel optimization endeavor with respect to the traditional approaches which use the aircraft design disciplines, like aerodynamics, propulsion, structures, etc, as their multilevel optimization subspaces. Accordingly, instead of each subspace representing a single discipline, the implemented methodology looks at each mission stage as a different subspace. There are four typical mission stages: take-off, climb, cruise/loiter and descent, as detailed in Section 3.4. In each mission profile, the user can define an arbitrary number of stages, other than take-off, from the aforementioned four.

Each subspace will thus be associated to one mission stages. However, and following the need to consider different sets of inputs and optimization variables for the climb and cruise stages, two different climbs and two different cruises are possible, as shown in Table 5.3. Accordingly, the weighting coefficient (${\lambda^i}_{weight}$) of the ECO architecture is defined as per Equation (5.15), where (E^i) is the subspace (i) energy and (E_{total}) is the total energy (sum of every subspace energy).

$$ {\lambda^i}_{weight} = \frac{E^i}{E_{total}} \tag{5.15} $$

This coefficient is updated at each system level iteration based on the energy values of the previous iterations. In the first iteration, the weight coefficient equals unity for all subspaces.

Table 5.3 features a list of the mission stage options, each having its own set of: inputs - rate-of-climb (RoC), velocity (V), range (Δx) and time (Δt); shared variables - wing mean chord ($\bar{c}$), propeller diameter (d_{prop}) and flap relative chord ($c_f/\bar{c}$); variables that can either be shared or local - wingspan (b) and propeller pitch (p_{prop}); and local variables that can be different amongst the various subspaces - flap deflection (δ_f); and objectives - take-off, climb, cruise and descent energy, respectively (E_{to}, E_{cb}, E_{cz}, E_{dt}).

Table 5.3: Distributed optimization problem formulation (LV - local variable; GV - global variable).

	Stage	Constraints	GV	GV / LV	LV	Objective
1	Take-off	-	$\bar{c}, d_{prop}, c_f/\bar{c}$	b, p_{prop}	δ_f	E_{to}
2	Climb	RoC				E_{cb}
3		-				
4	Cruise	V, Δx				E_{cz}
5		V, Δt				
6	Descent	-				E_{dt}

Some variables can only be local (LV) or global (GV), whereas some others can be both local or global (LV/GV). As for the wingspan (b), it this is a GV if a FW is being analyzed, whereas it is a LV if a VSW is considered. Likewise, the propeller pitch (p_{prop}) may be a GV, when using a fixed pitch propeller, whereas it is a LV if a VPP is adopted. Contrarily, if a VCF is in use, it makes no sense to force the same flap deflection in all mission stages, and therefore the flap deflection (δ_f) is always a LV. As for the wing chord ($\bar{c}$), the propeller diameter (d_{prop}) and the flap chord fraction ($c_f/\bar{c}$) these can only be GV since no morphing mechanisms interfering with these variables have been devised in the current study.

As the weight (W_{total}) is one of the most impactful variables, it is very important to define which weight fractions are the inputs and the outputs of each optimization. As per definition the total weight is as follows (5.16):

$$W_{total} = W_{pay} + W_{sys} + W_{str} + W_{ene} \tag{5.16}$$

However, since it is possible to adopt different strategies in terms of weight optimization, several different working modes of the multilevel optimization routine are possible. They have been summarized in Table 5.4.

Table 5.4: Weight fractions depending on the optimization mode selected.

Mode	Parameters (Inputs)	Variables (Outputs)
1	W_{sys}, W_{pay}	$W_{str}, W_{ene}, W_{total}$
2	$W_{sys}, W_{pay}, W_{ene}$	W_{str}, W_{total}
3	W_{sys}, W_{total}	$W_{str}, W_{ene}, W_{pay}$
4	$W_{sys}, W_{ene}, W_{total}$	W_{str}, W_{pay}

All the results presented in hereinafter refer to mode 1, as defined in Table 5.4. The payload and systems weight (W_{sys}, W_{pay}) are being user inputs, whereas the structural, energy and total weights ($W_{str}, W_{ene}, W_{total}$) are outputs.

In the first two mission profiles presented below an electrical motor has been used. The energy weight estimates are based upon a typical LiPo battery's specific energy [J/kg]. Both the motor and the systems energy consumption have been considered, with the latter referring to the energy by other components, like the servomechanisms, the Electronic Speed Control, Cameras, etc.

Further details on the Disciplinary Models used can be obtained in Chapter (2), where all the Analysis Models are thoroughly explained.

5.5.1 Morphing Technologies

Amongst the morphing technologies that could have been employed in the current study, three of the most impactful have been chosen. These are two morphing wing technologies, variable span wing (VSW) and variable camber flap (VCF), as well as one non-wing morphing technology, a variable pitch propeller (VPP) solution.

The VSW choice has been primarily based on the fact that it is clearly one of the morphing wing solutions with the greatest performance impact since the lifting surface area and aspect ratio are parameters of paramount relevance. The fact that in-house works on such a concept have already proven successful [114, 115] along with the recent development of a weight estimate correlation [101] for variable span wings have further supported this choice.

The VCF choice can be largely justified on the fact that it is clearly one of the morphing wing solutions with the greatest performance impact since the actual wing airfoil camber is of utmost aerodynamic relevance. In addition to this, this choice brought the additional benefit of encompassing two of the morphing mechanisms developed for the CHANGE project [113].

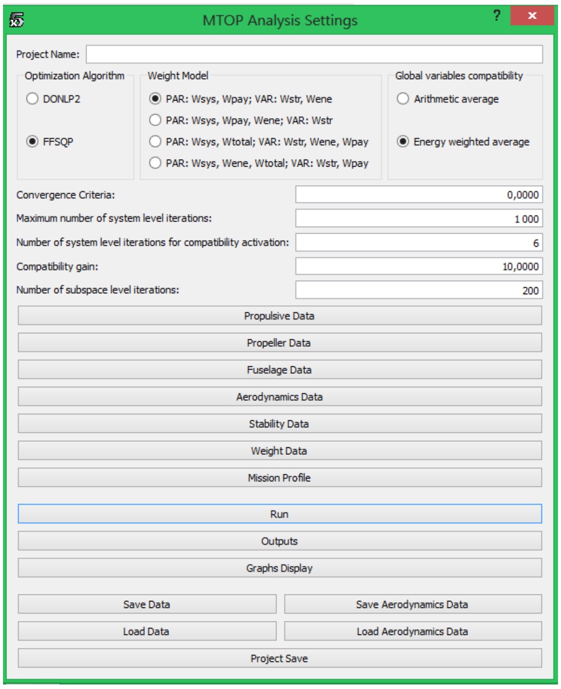
Lastly, the VPP has been chosen despite representing a more conventional morphing solution, on the basis it can provide meaningful performance gains without a particularly relevant systems weight penalty. This morphing solution has been the most used as well as the most commercially adopted [78] amongst the three studied as it can enlarge the aircraft flight envelope and diminish its energy consumption.

The computational tool developed is expected to provide some quantitative insight on the actual gains of these solutions. Nevertheless, it should be kept in mind that conversely to what happens for the VSW, the structural and systems weight penalty of the VCF and VPP have not been considered, due to a lack of historical records or dedicated studies on the subject.

5.6 Graphical User Interface

The current section presents a graphical user interface (GUI) tailored for widening the scope of possible users of the implemented MTOP methodology. The approach followed in its development was largely the same of the PARROT GUI. Therefore, much of the considerations made in Section 4.3 still hold. A more detailed description of the MTOP GUI can be found in Appendix E, which features the MTOP GUI users' manual, where all the software GUI windows are shown and the user can see a list of all the required inputs and provided outputs.

The MTOP code main menu is shown in Figure 5.5. The first options are related to the loading of inputs, with specific menus for the propulsion, systems, fuselage, aerodynamics and weight data as well as for the intended mission profile performance targets. Despite broadly working in the same way, this window is significantly different from the PARROT main menu, in the way it requires further inputs, namely regarding optimization settings, like the optimization algorithm, optimization mode, weighting approach, among others.


MTOP Analysis Settings

Project Name:

Optimization Algorithm
○ DONLP2
● FFSQP

Weight Model
● PAR: Wsys, Wpay; VAR: Wstr, Wene
○ PAR: Wsys, Wpay, Wene; VAR: Wstr
○ PAR: Wsys, Wtotal; VAR: Wstr, Wene, Wpay
○ PAR: Wsys, Wene, Wtotal; VAR: Wstr, Wpay

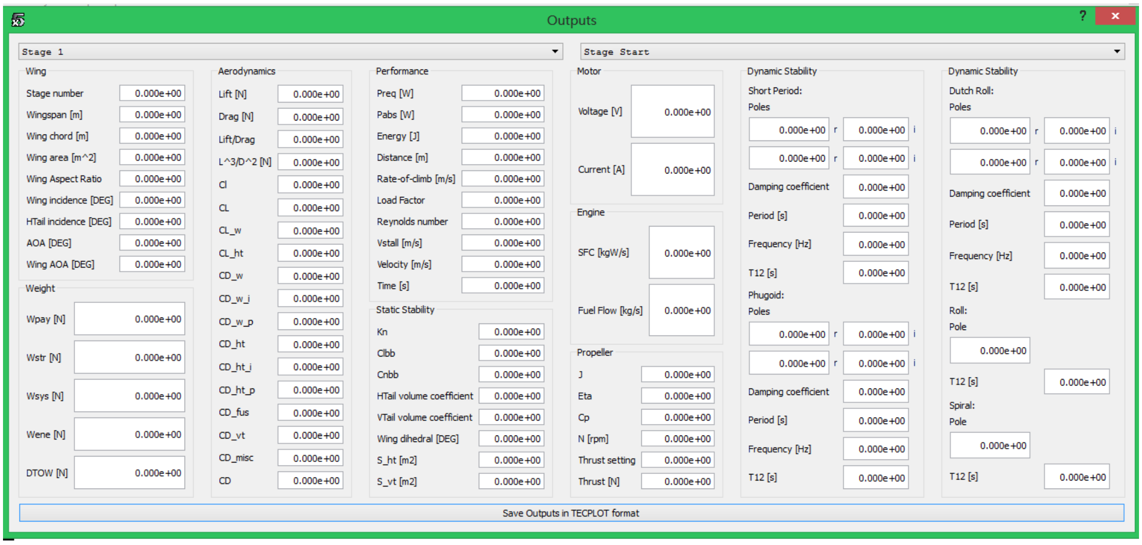
Global variables compatibility
○ Arithmetic average
● Energy weighted average

Convergence Criteria: 0,0000
Maximum number of system level iterations: 1 000
Number of system level iterations for compatibility activation: 6
Compatibility gain: 10,0000
Number of subspace level iterations: 200

Figure 5.5: Main interface window.

As the number of input parameters is relatively large, once the user has loaded all the data for the first time, it is possible to generate a ".txt" file which will store all the project data, exactly in the same way it is done in the PARROT GUI. The way it works is shown in Section 4.3.

The output window is shown in Figure 5.6. It enables the designer to see the numeric output of a large number of output variables for each wing geometry at the beginning and end of each mission stage for the optimal design solution achieved, providing the same outputs of the PARROT GUI.



Stage 1

Wing
Stage number 0.000e+00
Wingspan [m] 0.000e+00
Wing chord [m] 0.000e+00
Wing area [m^2] 0.000e+00
Wing Aspect Ratio 0.000e+00
Wing incidence [DEG] 0.000e+00
HTail incidence [DEG] 0.000e+00
AOA [DEG] 0.000e+00
Wing AOA [DEG] 0.000e+00

Weight
Wpay [N] 0.000e+00
Wstr [N] 0.000e+00
Wsys [N] 0.000e+00
Wene [N] 0.000e+00
DTOW [N] 0.000e+00

Aerodynamics
Lift [N] 0.000e+00
Drag [N] 0.000e+00
Lift/Drag 0.000e+00
L^3/D^2 [N] 0.000e+00
Cl 0.000e+00
CL 0.000e+00
CL_w 0.000e+00
CL_ht 0.000e+00
CD_w 0.000e+00
CD_w_i 0.000e+00
CD_w_p 0.000e+00
CD_ht 0.000e+00
CD_ht_i 0.000e+00
CD_ht_p 0.000e+00
CD_fus 0.000e+00
CD_vt 0.000e+00
CD_misc 0.000e+00
CD 0.000e+00

Performance
Preq [W] 0.000e+00
Pabs [W] 0.000e+00
Energy [J] 0.000e+00
Distance [m] 0.000e+00
Rate-of-climb [m/s] 0.000e+00
Load Factor 0.000e+00
Reynolds number 0.000e+00
Vstall [m/s] 0.000e+00
Velocity [m/s] 0.000e+00
Time [s] 0.000e+00

Static Stability
Kn 0.000e+00
Clbb 0.000e+00
Cnbb 0.000e+00
HTail volume coefficient 0.000e+00
VTail volume coefficient 0.000e+00
Wing dihedral [DEG] 0.000e+00
S_ht [m2] 0.000e+00
S_vt [m2] 0.000e+00

Stage Start

Motor
Voltage [V] 0.000e+00
Current [A] 0.000e+00

Engine
SFC [kgW/s] 0.000e+00
Fuel Flow [kg/s] 0.000e+00

Propeller
J 0.000e+00
Eta 0.000e+00
Cp 0.000e+00
N [rpm] 0.000e+00
Thrust setting 0.000e+00
Thrust [N] 0.000e+00

Dynamic Stability
Short Period:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00
Phugoid:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00

Dynamic Stability
Dutch Roll:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00
Roll:
Pole
0.000e+00
T12 [s] 0.000e+00
Spiral:
Pole
0.000e+00
T12 [s] 0.000e+00

Figure 5.6: Output data interface window.

5.7 Case Studies

In order to point out the advantages and disadvantages of the optimization scheme implemented three different mission profiles will be considered. Their mission patterns are depicted in Figures 5.7 through 5.15. These have been chosen because although they require an amount of energy of the same order of magnitude, their profiles are clearly different. This is particularly useful to show how the morphing mechanisms profitability depends on the mission profile. In all three missions, the weight optimization method 1 was used (as defined in Table 5.4), with the goal of minimizing the energy consumption.

In order to provide a fair comparison between a fixed-wing (FW) - a wing with a constant geometry - and a variable span wing (VSW), the span efficiency factor (e) used in the two conditions has been different. Concerning 3D aerodynamics, it is worthwhile to mention that the span efficiency factor was considered a function of the wing planform shape and operating conditions. In order to provide a reliable estimate for its value, the following approach has been devised. A FW with a straight trailing edge, three trapezoidal parts and twist at the tip and a VSW with a rectangular planform and twist at the tip were optimized using a dedicated computational tool running a non-linear vortex lattice method (VLM) formulation based on Katz (2001) [116] and Mukherjee and Gopalarathnam (2003) [117]. Three different wing aspect ratios (Λ) have been analyzed for both wing configurations (FW and VSW), with the aerodynamic coefficients being plotted against the wing angle-of-attack. From these, it became possible to estimate the span efficiency factor functions, as per Equation 3.7.

Due to the geometric limitations of a VSW telescopic mechanism, the wing planform is probably going to have the approximate shape of a rectangle and limited twist confined to a well designed wing tip, and therefore the span efficiency factor considered for the wing alone was ($e^{VSW} = 1.015$), (Fig. 5.4). Conversely, the FW planform shape shall be optimized to achieve better 3D aerodynamic efficiency, hence, the span efficiency factor considered for the wing alone was ($e^{FW} = 1.05$).

For mission profiles I and II, the motor chosen was the Scorpion SII-4025-520KV. Conversely, since mission profile III is expected to be significantly more energy consuming, a combustion engine system has been adopted - the 3W-55i engine. The motor and engine specifications can be found in Appendix F. In the optimization cases when the propeller pitch and diameter are design variables their values may obviously vary with the user setting values corresponding to their initial guesses. This choice of motor, engine and propellers have been arbitrated thinking of the UAV design take-off weight (DTOW) forecast and operating range of speeds. In all three mission profiles, the lifting surfaces' airfoils chosen were the same. For estimating the energy weight - either batteries or fuel depending on the propulsive source in use - a residual energy fraction of 30% and a safety margin of 10% have been considered.

Table 5.5 lists the different optimization cases considered, each with its own set of constants, local variables and global variables with respect to each mission stage. By benchmarking the different optimization cases for each mission, it is possible to conclude about the most energy efficient solution and also about the profitability of morphing solutions, namely VSW, VCF and VPP. Eighteen different optimization cases are presented (six per mission).

Optimizations B, C and D can be used to assess the profitability of each device - variable span wing (VSW), variable pitch propeller (VPP) and variable camber flap (VCF), respectively -

Table 5.5: Optimization cases for the three mission profiles considered, (P => Parameter; GV => Global Variable; LV => Local Variable).

Case	$\bar{c}$	b	d_{prop}	p_{prop}	$c_f/\bar{c}$	δ_f	Notes
A	GV	GV	GV	GV	P	P	Baseline
B	GV	LV	GV	GV	P	P	VSW
C	GV	GV	GV	LV	P	P	VPP
D	GV	GV	GV	GV	GV	LV	VCF
E	GV	GV	GV	LV	GV	LV	VPP & VCF
F	GV	LV	GV	LV	GV	LV	VSW, VPP & VCF

when compared to the benchmarking case A. Furthermore, case E refers to the state-of-the-art solution in today's manned aircraft while case F makes it possible to assess the combined effect of all three devices.

In addition, and in order to evaluate the profitability of the weighting coefficient as it has been defined in Equation (5.15), optimization case I.A and I.B are run twice - a first time using the standard approach, thus arithmetically weighting each subspace, thus assigning an equal relevance to each subspace regardless of their relative energy consumption, and a second time without including the new weighting coefficient presented, which depends on the relative energy consumption of each subspace. In addition to this, a compatibility factor study is also performed using the same optimization cases.

In all of the three mission profiles presented in the forthcoming sections the wing airfoil chosen was the OGIII - which is an in-house developed airfoil optimized using a specifically tailored software tool for airfoil shape optimization. It performs well in the range of Reynolds numbers and wing loadings that the mission profiles and remaining design specifications require. The airfoil chosen for both the horizontal and vertical stabilizers was the NACA 0009, which is also a common choice for these lifting surfaces.

The case studies' missions have been chosen in such a way that they result in a similar DTOW, thus the optimization variables bounds are the same, except for the propeller size, in accordance with Table 5.6.

Table 5.6: Optimization variables bounds.

Variable	$\bar{c}[m]$	$b[m]$	$d_{prop}[in]$	$p_{prop}[in]$	$c_f/\bar{c}$	$\delta_f[deg]$
Minimum	0.25	2.65	14.0 (I & II) / 20.0 (III)	6.0 (I & II) / 14.0 (III)	0.2	-5.0
Guess	0.30	3.10	16.0 (I & II) / 22.0 (III)	8.0 (I & II) / 16.0 (III)	0.3	0.0
Maximum	0.35	3.55	18.0 (I & II) / 24.0 (III)	10.0 (I & II) / 18.0 (III)	0.4	15.0

While benchmarking the different optimization cases (VSW, VCF and VPP), it should be kept in mind that the implemented physical models only encompass a structural weight penalty for the VSW, which is among the three solutions, the one that is expected to have a greater impact on the structural weight. This means that the actual results for the VCF and VPP should actually be a little worse than the ones presented below.

5.7.1 Mission I

Fig. 5.7 depicts the case study mission I. It consists of a take-off ($V_{wind} = 0$, $h_{to} = 0$), climb (power setting is fixed to 100%, C_l is selected for maximum rate of climb, $h_{min} = 0m$, $h_{max} = 1,000m$), cruise ($V = 30m/s$, $\Delta t = 1,800s$), and a final descent ($V_{min} = 15m/s$ (limited by $0.8 < C_l < 0.9$), $h_{max} = 1,000m$, $h_{min} = 0m$) to the take-off altitude with the goal of landing. This profile corresponds to using the mission stages 1, 3, 4 and 6, in accordance with their definition of Table 5.3.

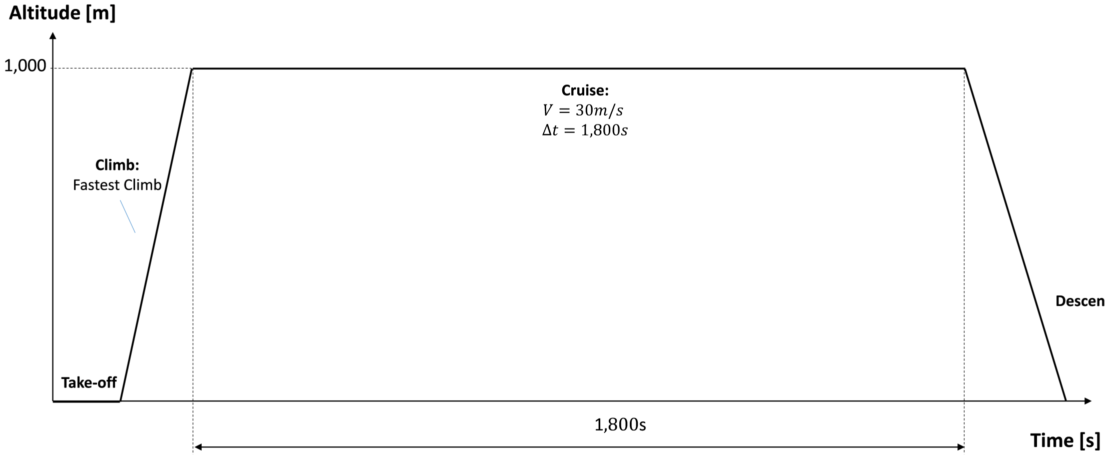


Figure 5.7: Case study mission I profile.

It is worthwhile to mention that the Scorpion $SII - 4025 - 520KV$ motor together with a $(16'' \times 8'')$ propeller have been used as initial guess.

The energy consumption results for each of the different optimization cases of mission I (Fig. 5.7) are shown in Fig. 5.8, whereas the vehicles' weight breakdown is presented in Table 5.7.

The results show that the VSW (case I.B) delivers poorer energetic performance when compared with a FW solution (case I.A). From Fig. 5.8, it is noticeable that the VSW penalizes the energy consumption on the climb and cruise stages, with an energetic penalty of (14%) and (2%), respectively. If one compares the lift-to-drag ratio (L/D) for each of these two flight phases, one concludes that the VSW (L/D) consumes always at least (11%) more than the FW homologue, as the former's geometry is closer to the aerodynamic optimum. Hence, it can be inferred that this increased energy consumption derives from the meaningful structural weight penalty of the VSW (case I.B). In addition, there is a predominant flight phase which accounts for the vast majority of the total energy spent which means that the VSW solution is less interesting.

The variable pitch solution (case I.C) results in a marginal energy saving, since the local optima are close, that is, as shown in Table G.1, thus the advantage of the VPP is less relevant. The variable camber flap (case I.D) shows a lower energy consumption when compared with the baseline configuration (I.A) and shows a relatively better performance when compared with the variable pitch propeller. Since both the variable pitch propeller and the variable camber flap bring energetic savings, their combined use brings an even greater energetic saving (case

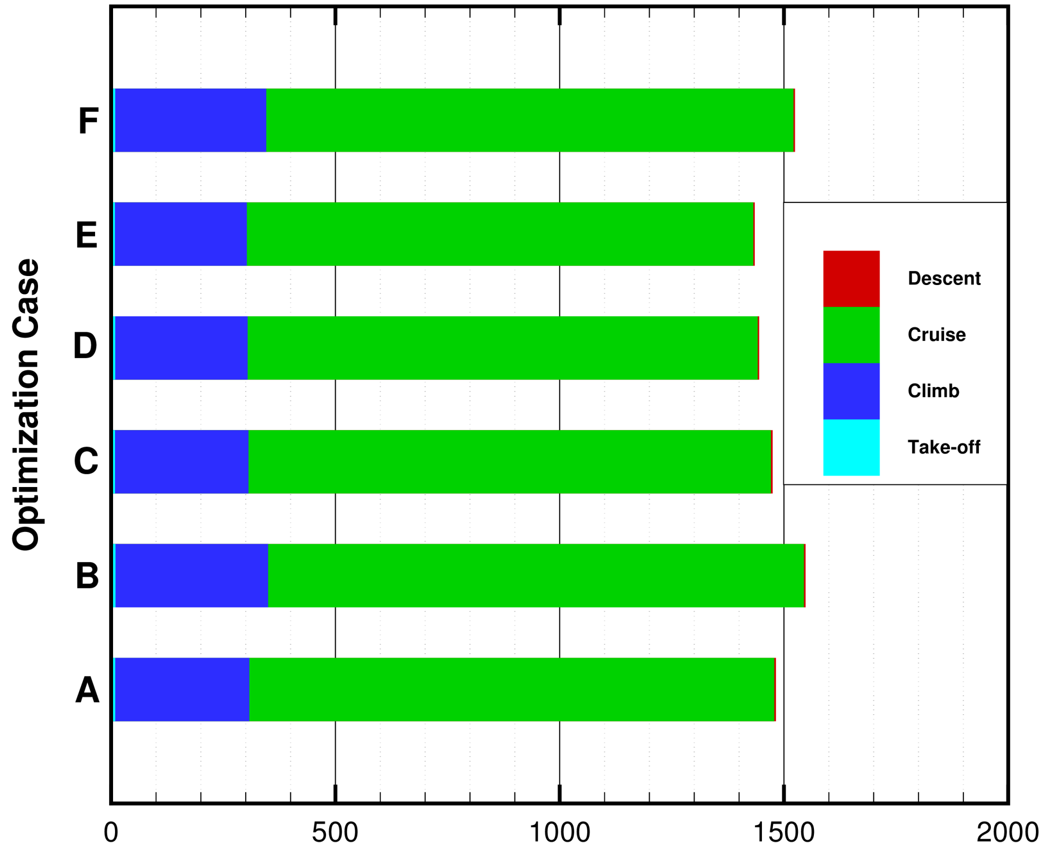


Figure 5.8: Energy Consumption [kJ] in mission I.

I.E). Finally, the combined use of a VSW, variable pitch propeller and flapped airfoil (case I.F) is penalized by the increased structural weight of the VSW mechanism, which can be justified likewise the poor relative performance of optimization case I.B.

Table 5.7: Aircraft weight distribution - mission I.

Case	W_{str}	W_{ene}	W_{sys}	W_{pay}	**MTOW**	**Notes**
I.A	38.63	30.89	11.46	40.00	120.98	Baseline
I.B	54.24	32.28			137.98	VSW
I.C	38.51	30.74			120.71	VPP
I.D	38.42	30.12			120.00	VCF
I.E	38.34	29.92			119.72	VPP & VCF
I.F	54.16	31.79			137.41	VSW, VPP & VCF

The optimized design variables for each optimization case are shown in Figure 5.9 and listed in Table G.1 in Appendix G.

The iteration history for all six optimization cases of Mission I is shown in Figure 5.10. It includes the evolution of the objective function (energy) and the global design variables, which are the ones that are subjected to compatibility between subspaces. These are plotted against

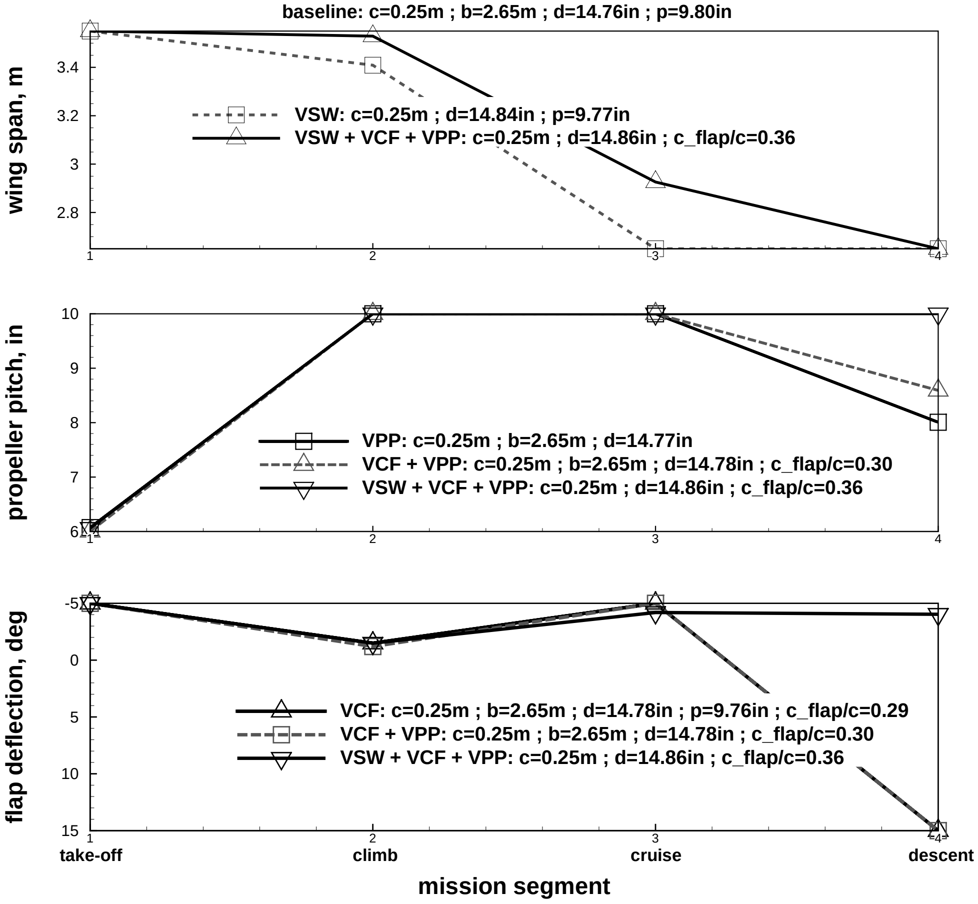
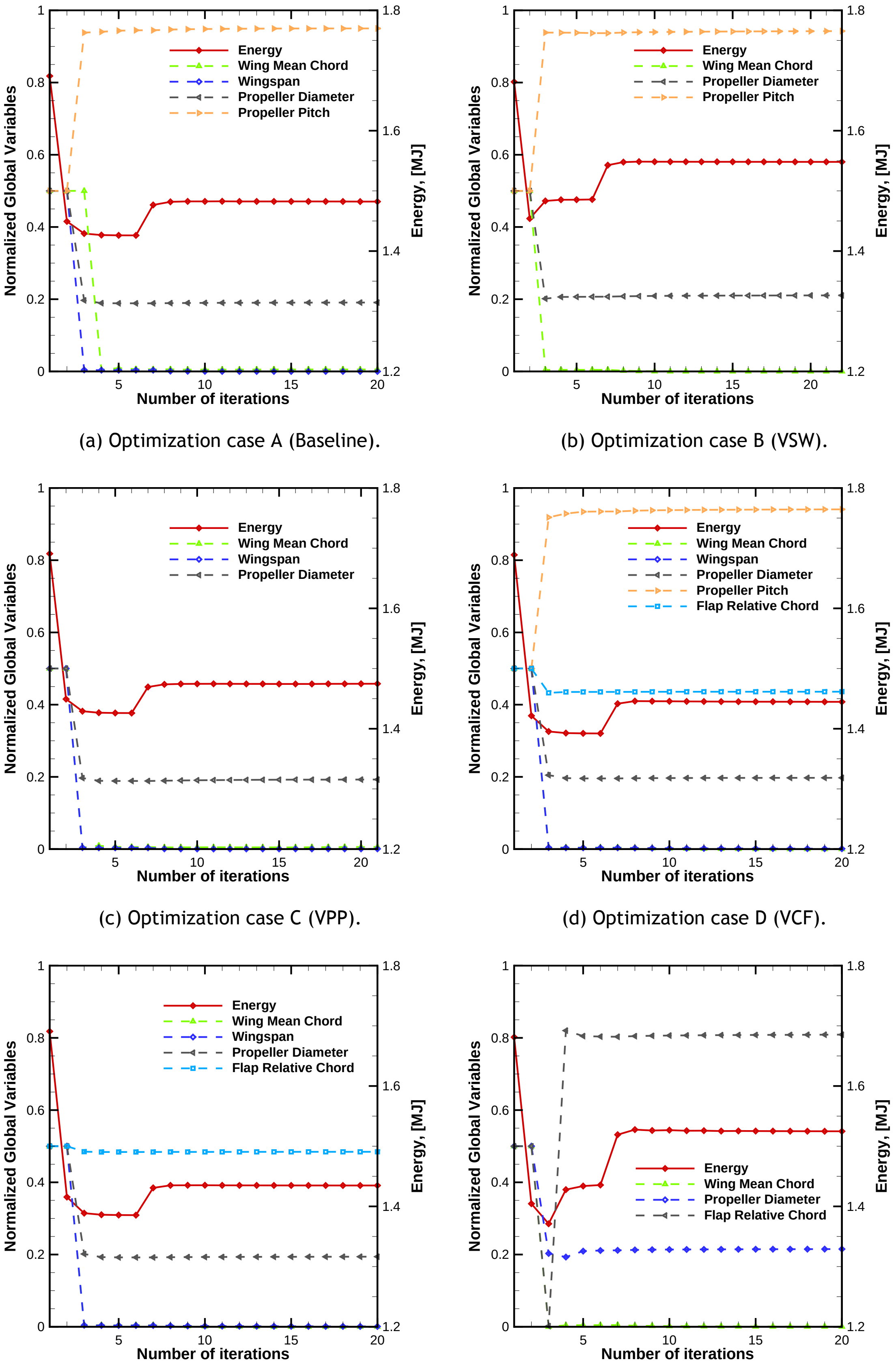


Figure 5.9: Mission I design variables as a function of the optimization cases.

the number of iterations.



(a) Optimization case A (Baseline).

(b) Optimization case B (VSW).

(c) Optimization case C (VPP).

(d) Optimization case D (VCF).

(e) Optimization case E (VPP & VCF).

(f) Optimization case F (VSW, VPP & VCF).

Figure 5.10: Convergence of global/shared design variables - mission I.

5.7.2 Mission II

Fig. 5.11 depicts the case study mission II. It consists of a **Take-off** ($V_{wind} = 0$, $h_{to} = 0$), **Climb 1** (power setting is fixed to 100%, C_l is selected for maximum rate of climb, $h_{min} = 0m$, $h_{max} = 250m$), low altitude high-speed cruise (Cruise 1) ($V = 33m/s$, $R = 20,000m$), **Loiter 1** at ($V = 15m/s$, $\Delta t = 30min$), further climb to high altitude (**Climb 2**) (power setting is fixed to 100%, C_l is selected for maximum rate of climb, $h_{min} = 250m$, $h_{max} = 1,000m$), high altitude low-speed cruise (**Cruise 2**) ($V = 25m/s$, $\Delta t = 800s$), first descent to low altitude (**Descent 1**) ($V = 18m/s$, $h_{max} = 1,000m$, $h_{min} = 250m$), **Loiter 2** ($V = 15m/s$, $\Delta t = 30min$), low altitude high-speed cruise (**Cruise 3**) ($V = 33m/s$, $R = 20,000m$), **Descent 2** to sea level ($V = 20m/s$ (limited by $0.8 < C_l < 0.9$), $h_{max} = 250m$, $h_{min} = 0m$) with the goal of landing. This profile corresponds to using the mission stages 1, 3, 4, 5, 3, 5, 6, 5, 4 and 6, in accordance with their definition, in Table 5.3.

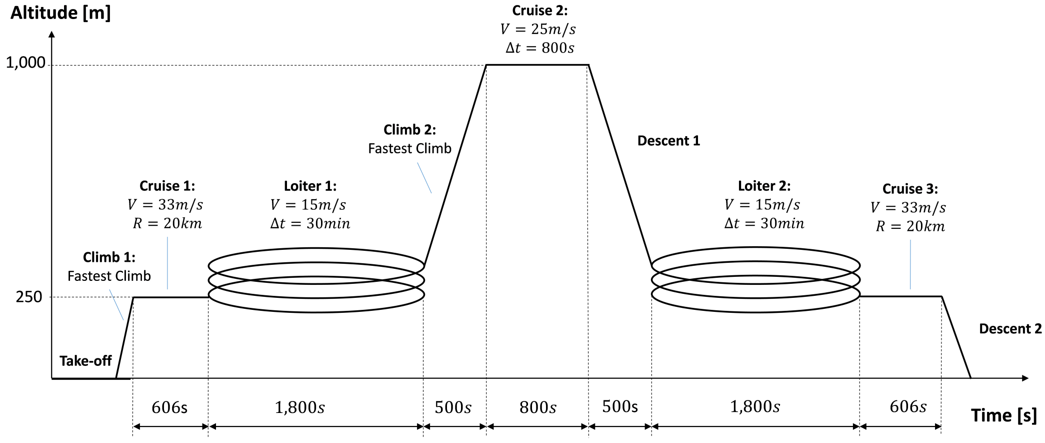

Altitude [m]
1,000
250
Take-off
Climb 1: Fastest Climb
Cruise 1: $V = 33m/s$ $R = 20km$
Loiter 1: $V = 15m/s$ $\Delta t = 30min$
Climb 2: Fastest Climb
Cruise 2: $V = 25m/s$ $\Delta t = 800s$
Descent 1
Loiter 2: $V = 15m/s$ $\Delta t = 30min$
Cruise 3: $V = 33m/s$ $R = 20km$
Descent 2
606s
1,800s
500s
800s
500s
1,800s
606s
Time [s]

Figure 5.11: Case study mission II profile.

Likewise mission I, the Scorpion $SII - 4025 - 520KV$ motor has been chosen, together with a $(16'' \times 8'')$ propeller as initial guess.

The energy consumption results for each of the different optimization cases of mission II (Fig. 5.11) are shown in Fig. 5.12, whereas the vehicles' weight breakdown is presented in Table 5.8.

In mission II, contrarily to what happens in mission I, the VSW (case II.B) delivers higher energetic performance when compared with a FW solution (case II.A). Analyzing each different mission stage separately, it is noticeable that in the climb stages, case II.B delivers poorer energetic performance than case II.A (over 10% more despite the better performance L/D of 5% and 7%, respectively), which is justified by the heavier aircraft structure and the tremendous impact it has on climbing (due to the potential energy increase). Likewise, in Cruise 2 there is a slightly loss of performance of the VSW (about 3%) with respect to the FW, whereas for Cruise 1 and Cruise 3 the energy consumption is nearly the same, with the increased aerodynamic performance balancing the increased weight. Conversely, in both Loiter stages there is a meaningful advantage of the VSW (about 8% less for an L/D which is nearly 13% higher than case I.A). Given the relative relevance of these stages for the overall energy consumption, these make case II.B outweighing case II.A. Finally, in the descent stages, although the aerodynamic

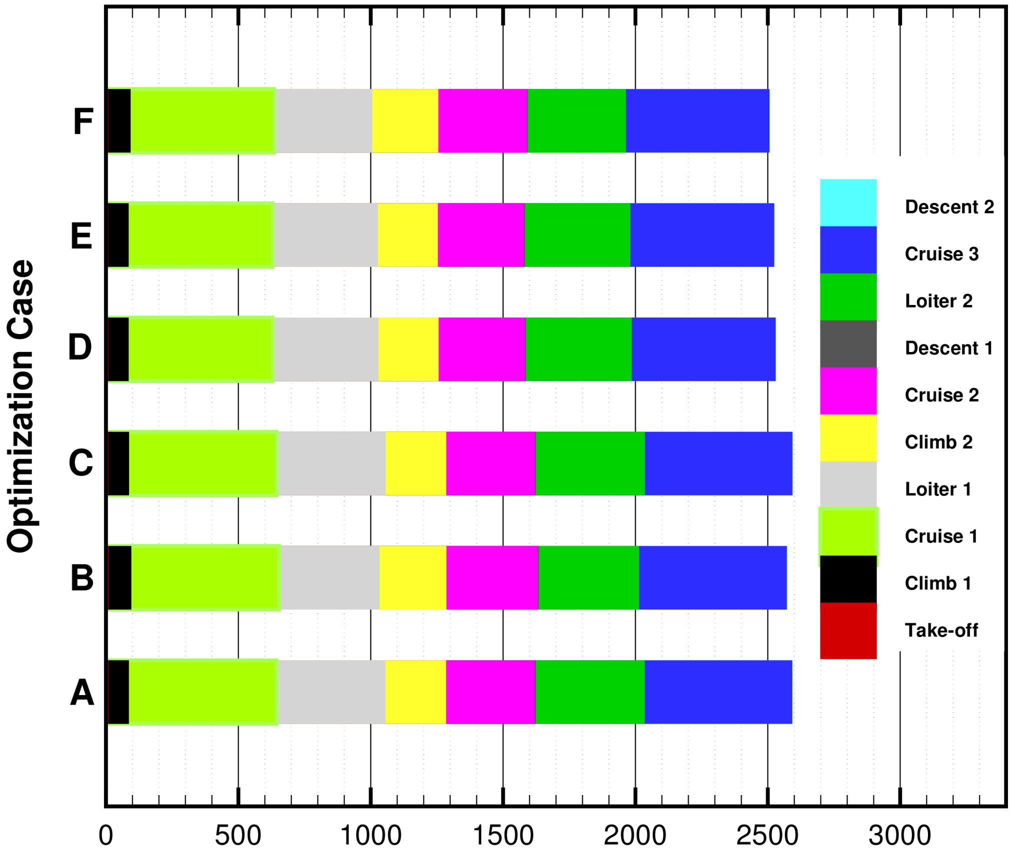


Figure 5.12: Energy Consumption [kJ] in mission II.

performance of the VSW worsens when compared with the FW, the heavier vehicle contributes to an energy saving of about 15% with respect to case II.A. Nevertheless, the descent stages energy consumption is nearly negligible when compared to the remaining stages.

Furthermore, mission II features a larger number of flight stages with similar energy consumption and different velocities, which altogether outweighs the structural weight penalty of the VSW solution. Hence, it can be concluded, with no surprise, that the VSW profitability is highly dependent on the mission profile.

It is noticeable that the variable camber flap solution (case II.D) results in an improved energetic efficiency, which is a result of the different velocity regimes of mission II, each with a different optimum airfoil shape. Following a similar reasoning, it can be easily inferred that the combined effect of the variable pitch propeller with the flap (case II.E) result in greater energy savings.

Amongst all the devised optimization cases, (case II.F) delivers the lowest energy consumption, which means that for this mission profile, all three devices (VSW, VPP and VCF) contribute to the overall energy savings. Once again, the enhanced aerodynamic performance of the VSW outweighs the structural weight penalty related to the VSW mechanism.

The optimized design variables for each optimization case are shown in Figure 5.13 and listed in Table G.2in Appendix G.

Table 5.8: Aircraft weight distribution - mission II.

Case	W_{str}	W_{ene}	W_{sys}	W_{pay}	MTOW	Notes
II.A	42.33	54.13	11.46	20.00	127.92	Baseline
II.B	55.79	53.70			140.95	VSW
II.C	42.36	54.14			127.96	VPP
II.D	42.66	52.82			126.94	VCF
II.E	42.43	52.70			126.59	VPP & VCF
II.F	55.00	52.34			138.80	VSW, VPP & VCF

The iteration history for all six optimization cases of Mission II is shown in Figure 5.14. It includes the evolution of the objective function (energy) and the global design variables, which are the ones that are subjected to compatibility between subspaces. These are plotted against the number of iterations.

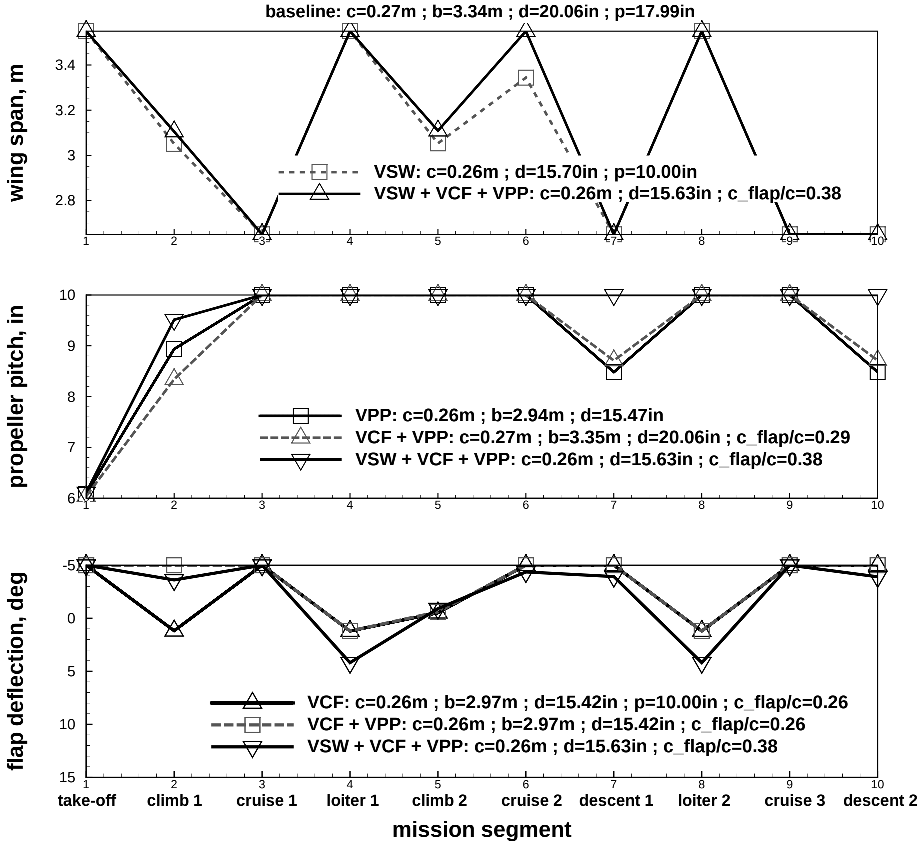
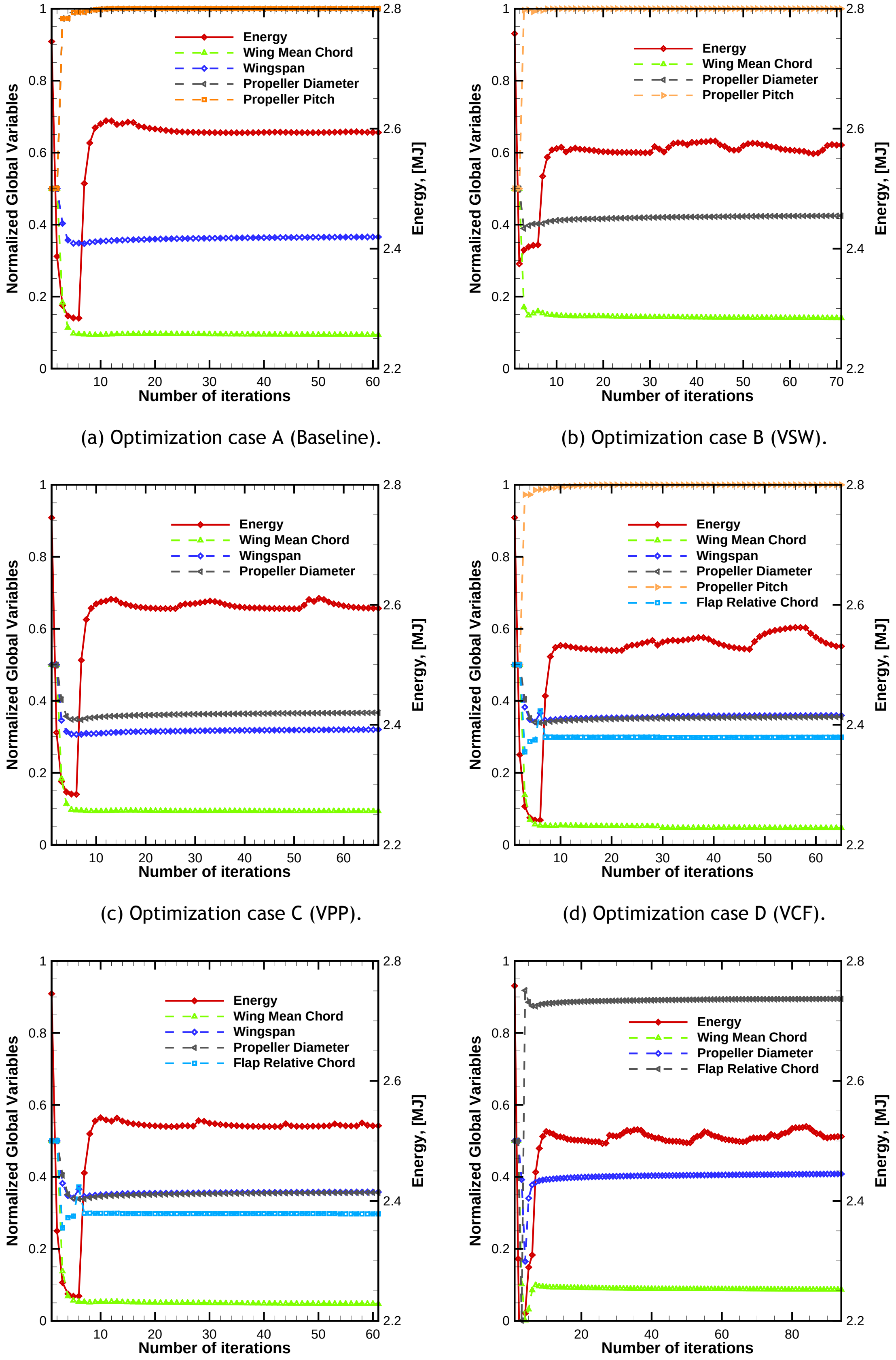


Figure 5.13: Mission II design variables as a function of the optimization cases.



(a) Optimization case A (Baseline).

(b) Optimization case B (VSW).

(c) Optimization case C (VPP).

(d) Optimization case D (VCF).

(e) Optimization case E (VPP & VCF).

(f) Optimization case F (VSW, VPP & VCF).

Figure 5.14: Convergence of global/shared design variables - mission II.

5.7.3 Mission III

Fig. 5.15 depicts the case study mission III. It consists of a **Take-off** ($V_{wind} = 0$, $h_{to} = 0$ and $\Delta x = 80m$), **Climb** (power setting is fixed to 100%, C_l is selected for maximum rate of climb, $h_{min} = 0m$, $h_{max} = 700m$), fast cruise speed with headwind (**Cruise 1**) ($V = 30m/s$, $R = 30,000m$ and $V_{wind} = 3m/s$), **Loiter** ($V = 16m/s$, $\Delta t = 32,400s$ and $V_{wind} = 0$), fast cruise speed with tailwind (**Cruise 2**) ($V = 30m/s$, $R = 30,000m$ and $V_{wind} = -3m/s$) and **Descent** to sea level ($V = 20m/s$, $h_{max} = 700m$, $h_{min} = 0m$) with the goal of landing. This profile corresponds to using the mission stages 1, 3, 4, 5, 4 and 6, in accordance with the definition of Table 5.3.

Since mission III is much more energy consuming than missions I and II, a combustion engine has been used ($3W - 55i$). The propeller ($22'' \times 16''$) has been adopted as initial guess.

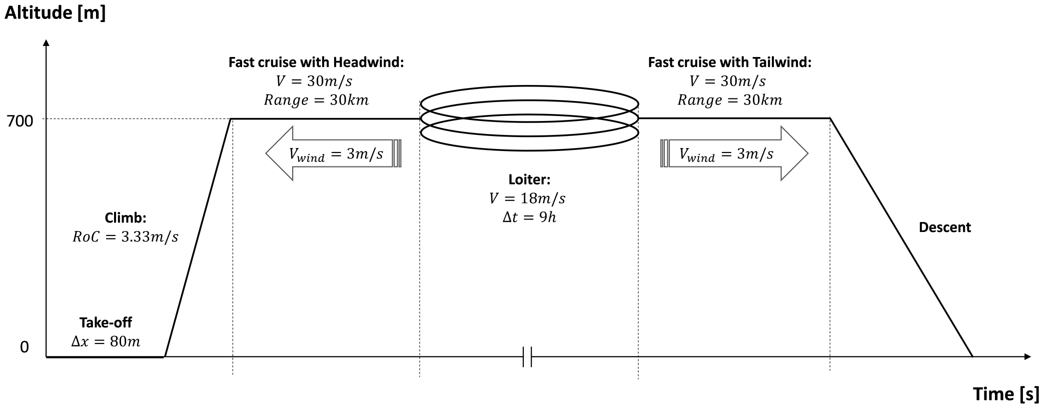


Figure 5.15: Case study mission III profile.

The energy consumption results for each of the different optimization cases of mission III (Fig. 5.15), the only using a combustion engine, are shown in Fig. 5.16, whereas the vehicles' weight breakdown is presented in Table 5.9.

In the third and last mission profile analyzed it turns out that the VSW (case III.B) delivers poorer energetic performance when compared with a FW solution (case III.A). Since this mission profile features a prominent loiter stage of 9 hours which accounts for about 75% of the total energy consumption. The VSW solution spends about 3% more energy than its FW homologue (despite a L/D 3% better), which means that the increased structural weight is more important. In the two cruise stages of case III.B, despite a very meaningful aerodynamic performance increase (L/D is 10% and 12% higher, respectively), the energy consumption is only 3% and 4% lower. Finally, despite their lowest significance, the take-off and climb are penalized in the VSW compared with the FW because of the structural weight increase in 6% and 7%, respectively, despite a (L/D) as much as 9% higher in the climb, again due to the significant weight penalty.

The variable pitch solution (case III.C) results in a marginal energy saving, since the local and global optima are close (optimum pitch is $18.00''$ in all mission stages except for mission stage number 2, which is $16.54''$), as shown in Table G.3. Conversely, in what refers to the variable camber flap solution (case III.D), there is a sensible energy saving. This means that the airfoil chosen is a good compromise between the highest and lowest velocities flight phases. However,

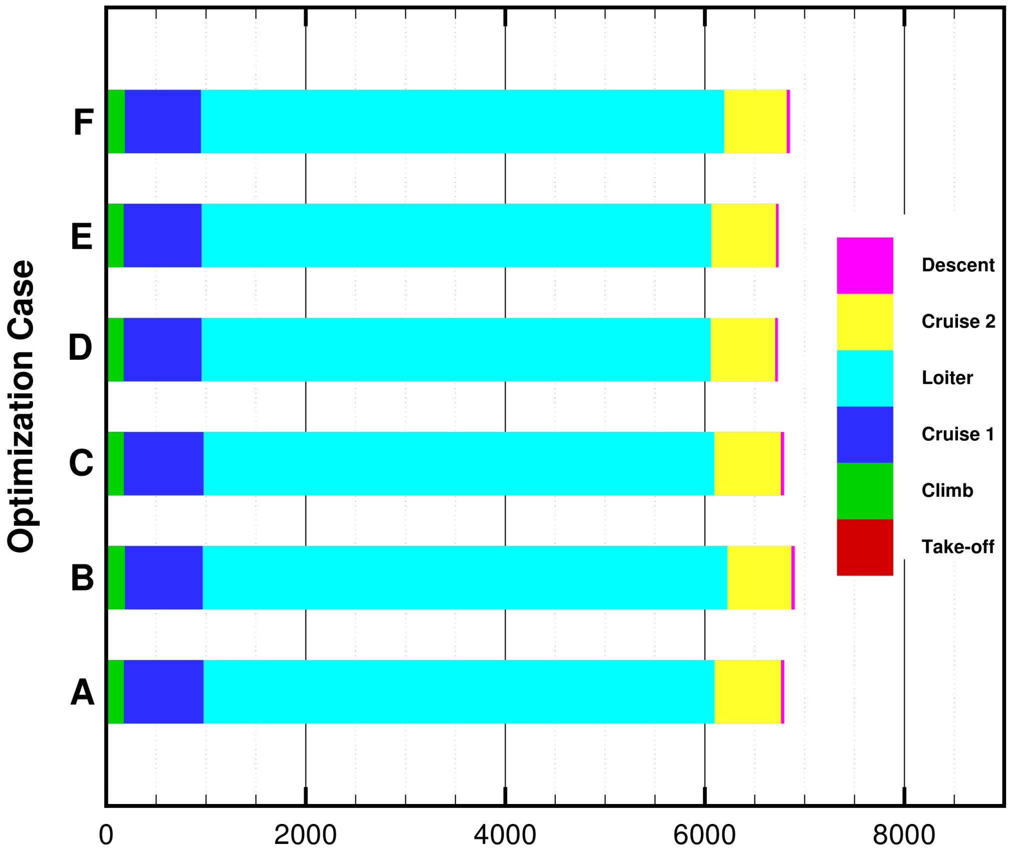


Figure 5.16: Energy Consumption [kJ] in mission III.

the low speed cruise is the predominant flight stage and therefore, an optimized airfoil for this particular stage could potentially present even more promising results.

As for the combined impact of a variable pitch propeller together with a variable camber flap (case III.E), it is noticeable that it contributes to an even greater energy saving when compared with the baseline solution (case III.A).

Despite delivering better energetic efficiency than the VSW (case III.B) alone, (case III.F) is also penalized by the increased structural weight due to the VSW mechanism and therefore it presents a poorer energetic efficiency than the ones with a constant geometry airfoil (cases III.A, III.C and III.D).

Table 5.9: Aircraft weight distribution - mission III.

Case	W_{str}	W_{ene}	W_{sys}	W_{pay}	MTOW	Notes
III.A	48.04	18.43	30.88	40.00	137.35	Baseline
III.B	56.75	18.66			146.29	VSW
III.C	48.04	18.42			137.34	VPP
III.D	48.06	18.31			137.25	VCF
III.E	48.22	18.33			137.43	VPP & VCF
III.F	56.72	18.56			146.16	VSW, VPP & VCF

The optimized design variables for each optimization case are shown in Figure 5.17 and listed in Table G.3in Appendix G.

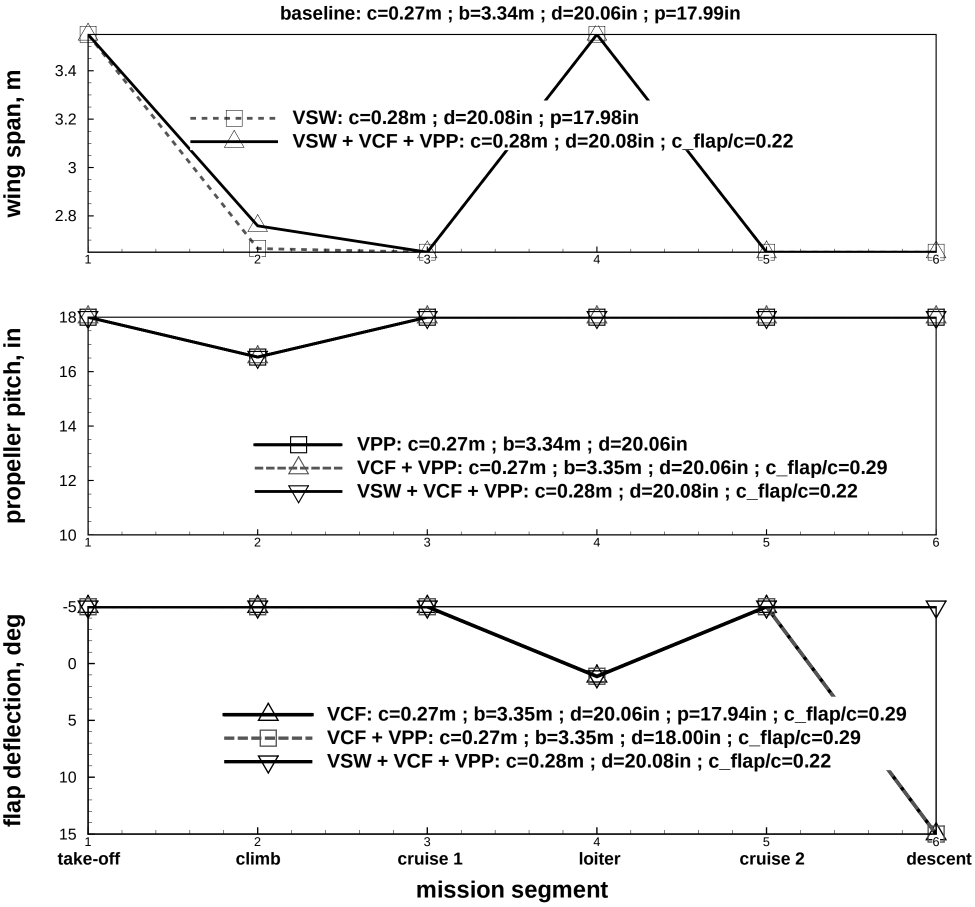
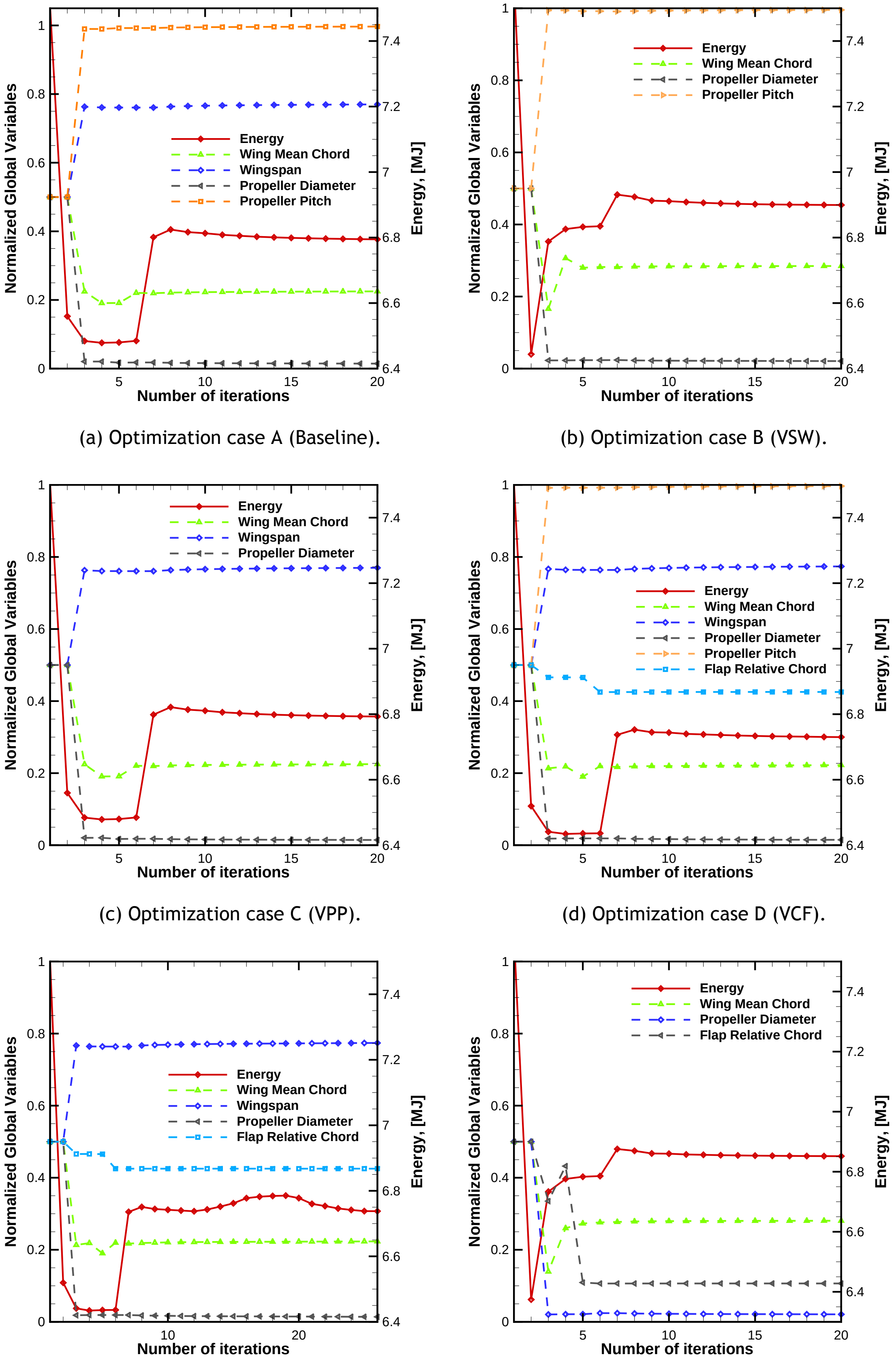


Figure 5.17: Mission III design variables as a function of the optimization cases.

The iteration history for all six optimization cases of Mission III is shown in Figure 5.18. It includes the evolution of the objective function (energy) and the global design variables, which are the ones that are subjected to compatibility between subspaces. These are plotted against the number of iterations.

The total running time of the three mission profiles analyzed for each of the six optimization cases is listed in Table H.1 of the Appendix H.



(a) Optimization case A (Baseline).

(b) Optimization case B (VSW).

(c) Optimization case C (VPP).

(d) Optimization case D (VCF).

(e) Optimization case E (VPP & VCF).

(f) Optimization case F (VSW, VPP & VCF).

Figure 5.18: Convergence of global/shared design variables - mission III.

5.7.4 Effect of Energy Weighting

The original Enhanced Collaborative Optimization formulation establishes an arithmetic average of the subspaces as the way to compute the system level target for each of the design variables. However, given the physics and the optimization layout devised in the current study it has been found worthwhile to assign a different weight to each subspace. Accordingly, the aim is to minimize the energy consumption optimizing the design variables for each subspace. It is therefore relevant to assign a greater relevance to the subspaces with the greatest energy consumption. To quantify this relevance, a benchmark study between the weighted average (Equation (5.15)) and the arithmetic average of the design variables has been performed. Its results have been summarized in Table 5.10, where (c^{norm},b^{norm},d^{norm}_{prop},p^{norm}_{prop}) represent the normalized wing mean chord, wingspan, propeller diameter and propeller pitch.

Table 5.10: Effect of energy weighting results.

Case	Average	Iterations	$E[J]$	$MTOW[N]$	c^{norm}	b^{norm}	d^{norm}_{prop}	p^{norm}_{prop}
I.A	**Weighting**	33	$1,481,024$	120.9	0.003159	1.75×10^{-6}	0.194	0.957
I.A	**Arithmetic**	28	$1,608,116$	126.4	0.259	0.201	0.645	0.608
I.B	**Weighting**	36	$1,546,946$	137.9	5.44×10^{-6}	0.825	0.215	0.951
I.B	**Arithmetic**	31	$1,662,629$	142.1	0.249	0.870	0.644	0.609

In optimization case I.A, where all the optimization variables are global variables, an arithmetic average results in a 8.6% energy consumption increase and a 4.6% DTOW increase with respect to the weighted average solution, despite enabling a 15% decrease on the number of system level iterations.

In optimization case I.B, where all the optimization variables are global variables with the exception of the wingspan (b), an arithmetic average results in a 7.5% energy consumption increase and a 3.1% DTOW increase with respect to the weighted average solution, though enabling a 14% decrease on the number of system level iterations. The lesser loss of performance in this optimization case is probably motivated by the wingspan being a local variable, which adapts to the other sub-optimal parameters enabling a relatively better final result.

In conclusion, these results have shown that an adequate weighting of the optimization variables is tremendously important. The magnitude of its relevance does however depend on the mission profile under analysis. The greater the difference of magnitude of the energy consumption at each subspace and the greater the difference of the design variables local optima, the greater is the relevance of duly weighting the design variables.

5.7.5 Effect of Compatibility Parameter

As already discussed, the compatibility parameter used for the sake of this study is a dynamic quantity instead of a constant value. This approach was followed to avoid a premature compatibility between subspaces that could prejudice the quest for minimization. As such, the user can choose the number of system level iterations up to the point where the compatibility parameter is set to zero (n_{comp}) to make sure that all the optimization variables are at their optimal position at each subspace:

$$\lambda_C^* = 0, n_{iter} \leq n_{comp} \tag{5.17}$$

After that point, the compatibility parameter value is computed according to:

$$\lambda_C^* = k_C \times (n_{iter} - n_{comp}), n_{iter} > n_{comp} \tag{5.18}$$

, where (k_C) is a compatibility factor and (n_{iter}) is the current iteration number.

It is therefore up to the user to set the compatibility factor value (k_C). This approach implies several trial-and-error runs because the number of iteration for convergence is not known in advance. After several runs, it was realized that this factor's choice could be very important. As such, it was decided to conduct a study on its impact.

Fig. 5.19 depicts the compatibility factor impact on mission I, case A total energy and design take-off weight, whereas Fig. 5.20 shows the same effect on the optimization variables.

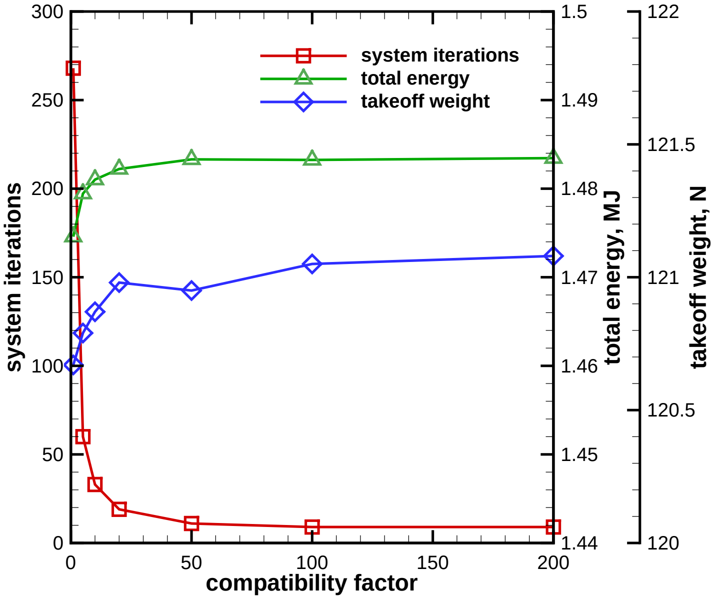


Figure 5.19: Total energy and take-off weight mission (I.A) compatibility factor study.

For case I.A (Figs. 5.19 and 5.20), the variations of energy, weight, wing mean chord and wingspan are always less than 1%. The variations in the propeller diameter and pitch are about 6%. A compatibility factor of 20 seems to be a balanced choice given that there is a reduction of 93% of the system level iterations with respect to a compatibility factor of 1.

Fig. 5.21 depicts the compatibility factor impact on mission I, case B total energy and design take-off weight, whereas Fig. 5.22 shows the same effect on the optimization variables.

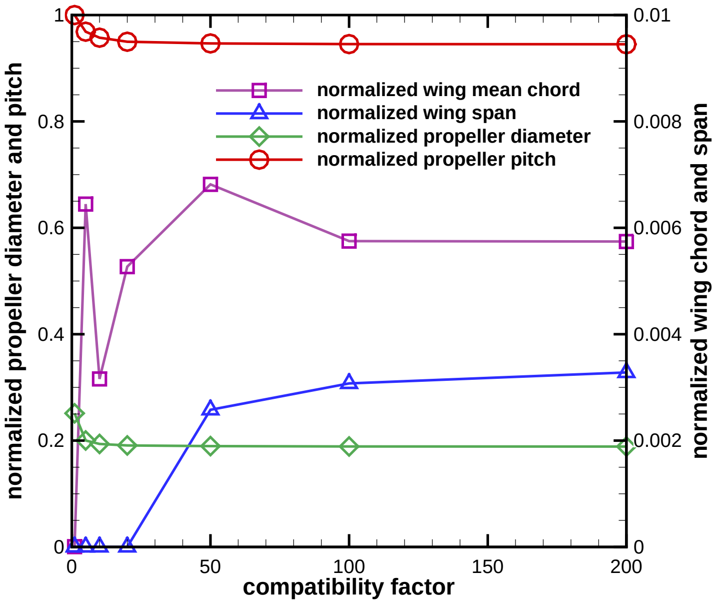


Figure 5.20: Optimization variables mission (I.A) compatibility factor study.

For case I.B (Figs. 5.21 and 5.22), the variations of energy, weight and wing mean chord are always less than 1%. The wingspan variation (when it is a local variable) is 15%. The variations in the propeller diameter and pitch are about 10%. In this case, a maximum compatibility factor of 10 would be preferable. In this case there is an 87% reduction in the number of iterations with respect to a compatibility factor of 1, though it has 70% more iterations than for a compatibility factor of 20.

The results of the two cases analyzed have shown consistency in attesting that the lower the compatibility factor, the lower will the aircraft energy consumption be. Conversely, the computational cost (number of iterations) grows sharply as the compatibility factor drops. It is also apparent that the optimized solution with the wingspan as a local variable (mission I.B) happens to be less influenced by the compatibility factor's choice, because of the fewer number of variables subject to compatibility, as was expected.

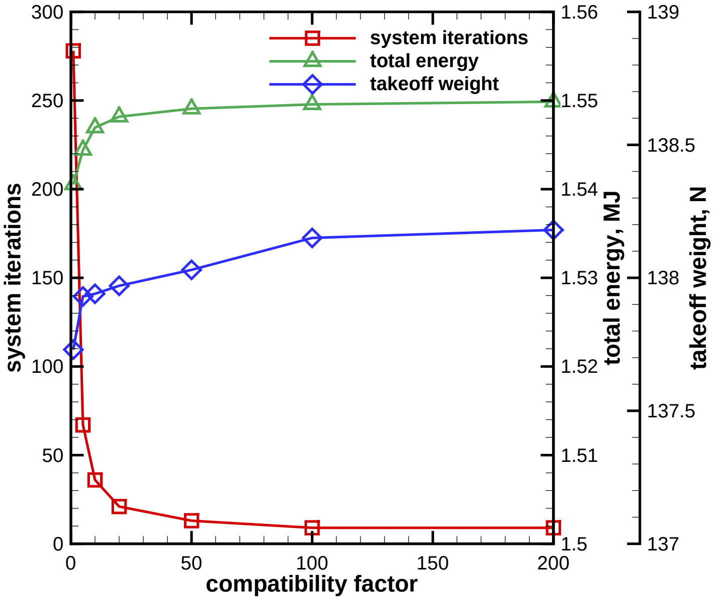


Figure 5.21: Total energy and take-off weight mission (I.B) compatibility factor study.

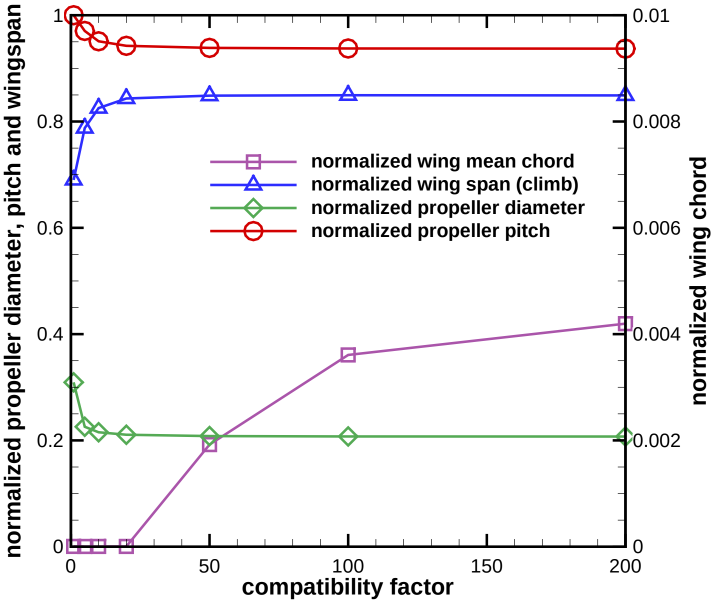


Figure 5.22: Optimization variables mission (I.B) compatibility factor study.

5.8 Concluding Remarks

The mission-based optimization implemented in the MTOP program herein devised is a holistic approach to the preliminary design stage of aircraft design, which shows the tremendous asset distributed design optimization architectures can be to the achievement of enhanced design solutions.

Using low-fidelity models for the aircraft design disciplines together with the Enhanced Collaborative Optimization (ECO) distributed architecture, the MTOP methodology introduced a new approach where the optimization subspaces are not the aforementioned traditional design disciplines, but rather the mission stages. Hence, a mission-based optimization methodology has been implemented.

Two modifications to the original ECO formulation have been presented and discussed. A weighting coefficient (${\lambda^{i}}_{weight}$) multiplies the squared difference between the subspaces best attempt to match the system level targets (x^{*}) and the system level targets themselves (z) which enables the user to assign a different relative weight to each subspace solution within the overall optimization. It has been shown how this straightforward add-in can meaningfully contribute to a duly weighted and consequently optimized solution. A dynamic compatibility parameter (λ_{c}^{*}) for the subspace level optimization, which is zero in the first system level iteration and then grows at a constant rate actively contributes to avoid an early compatibility between subspace optimizations that could otherwise jeopardize the quest for minimization. Experience has shown that some tuning to each particular problem shall be required to define the rate at which this parameter should grow, so that a good compromise between efficiency (solution time) and effectiveness (optimized value) is found.

Provided the local minima problem, inherent to any gradient-based optimization algorithm and the possibility of getting trapped at an unfeasible design point are suitably handled, the MTOP code allows its users to swift reach an optimized design point at the preliminary design stage of UAVs. This contributes to either expediting or avoiding some of the costly handmade standard design approaches which have been widespread over the last quarter of the twentieth century with the books of Roskam (1985) [118] and Raymer (1989) [1]. Accordingly, the preliminary design stage is therefore concomitantly more efficient (faster) and more effective, in the way the user can get closer to the optimum aircraft layout than it perhaps would via multiple spreadsheets with disciplinary analysis.

In addition to this, the MTOP architecture features a collateral asset of being tailored for evaluating the profitability of morphing wing concepts - each optimization subspace represents a different mission stage and therefore morphing-related design variables are local to each subspace. Likewise, global variables are locally optimized - with respect to each subspace - and globally made compatible using penalty parameters on the local objective functions.

The showcased mission-based MDO methodology makes it possible to design an aircraft for a given simple or complex mission in a expeditious and fairly automated way, involving all the mainstream preliminary aircraft sizing disciplines (aerodynamics, propulsion, weight and stability). Additionally, it features the ability to analyze the effect of different morphing/adaptive devices, which largely widens its range of applications and represents a step up towards a systematic and comprehensive assessment of innovative solutions.

It has been shown how the selection of any combination of morphing/adaptive technologies is highly dependent on the mission profile. Moreover, the morphing/adaptive mechanisms weight has a strong impact on the overall performance which is not easily grasped without an optimization methodology, further justifying the development of comprehensive design optimization methodologies like the one presented.

Finally, it is worthwhile to discuss how the selection of the optimization design variables minimum, maximum and initial guess values may impact the feasibility and the quality of the results. On one hand, if any of these values yield an unfeasible design point, the routine stops and it becomes virtually impossible to get an output. On the other hand it may happen that the optimum value of a variable in one or more subspaces lies out of the specified range, which means that the weighted optimum is mistakenly biased by such fact. In order to overcome these shortcomings, a careful selection of the optimization variables ranges, initial guesses along with ECO's dynamic compatibility parameter gain is paramount.

While this methodology can not only aid the designer getting closer to the actual optimum solution but also in decreasing the preliminary design optimization lead time, it should be kept in mind that an inadequate choice of the optimization variables ranges or initial guesses might put the whole process at risk. Accordingly, a worthy use of this methodology shall require a fair knowledge of the ECO architecture as well as of the inherent limitations of gradient-based optimization algorithms.

The three mission profiles under analysis have clearly shown that the profitability of a VSW from an energy efficiency viewpoint is highly dependent on the kind of mission under consideration. It has been shown how the best combination of morphing/adaptive technologies is highly dependent on the mission profile. Moreover, the morphing mechanisms weight, namely on the VSW, has a strong impact on the overall performance which is not easily grasped without an optimization methodology, further justifying the development of comprehensive design optimization methodologies like the one presented.

The full list of outputs provided by MTOP is listed in Tables I.1 and I.2 of Appendix I. A free version of the MTOP code shall soon be released online so that it becomes available for the academic and engineering community.

Chapter 6

Conclusions

6.1 Summary

In the course of this thesis two low-fidelity multidisciplinary UAV design optimization methodologies were presented. Their most distinguishing feature is to rely on a mission-based optimization approach, where the optimum aircraft layout significantly depends on the mission profile to which it is optimized rather than on a specific flight condition, as it is common in most design optimization methodologies.

The first methodology is based on a parametric study approach and relies on evaluating several combinations of some paramount design variables (e.g. wing mean chord and wingspan) to generate multidimensional plots and multiple performance metrics data. By doing so the designer can understand the impact of changing the design variables in a rather straightforward way.

The second methodology features a multidisciplinary design optimization architecture, Enhanced Collaborative Optimization, which provides an automated way of optimizing the aircraft layout using off-the-shelf gradient-based optimization algorithms. An additional advantage of this tool is that it enables the assessment of morphing solutions, namely variable span wing (VSW), variable camber flap (VCF) and variable pitch propeller(VPP).

While the parametric methodology provides results to feed the user's guided optimization, the distributed optimization approach performs the optimization automatically thus reducing the designer's workload. However, this last methodology is more sensitive to the topology. In case the objective function gets trapped in local extrema that can ultimately contribute to a poorer optimization result.

These methodologies have been used to create two different design optimization programs, which have been thoroughly tested and can thereafter be used to guide the designer aircraft sizing task at a preliminary design stage. The disciplinary analyses considered have been the aerodynamics, propulsion, performance, weight and stability (static and dynamic). In addition to that, their graphical user interfaces (GUIs), which largely enhance their user friendliness, have also been developed and tested.

It has been shown how the selection of any combination of morphing/adaptive technologies is highly dependent on the mission profile. Moreover, the morphing mechanisms weight has a strong impact on the overall performance which is not easily grasped without an optimization methodology, further justifying the development of comprehensive design optimization methodologies like the one presented.

The results obtained in this research will form the basis for continuing work towards improving the two multidisciplinary design optimization methodologies, including both programs and respective GUIs. In addition to that, a test bench UAV is being built in order to provide a comprehensive experimental assessment of both the fidelity of the disciplinary analyses adopted and the quality of the optimization methodologies under consideration.

6.2 Contributions to the State-of-the-Art

The core contributions of this thesis are two mission-based multidisciplinary design optimization methodologies for UAVs - a **PAR**amet**R**ic design **O**p**T**imization (PARROT) and a **M**ul**T**ilevel Design **OP**timization (MTOP). The showcased methodologies make it possible to design an aircraft for a given simple or complex mission in an expeditious way, involving all the mainstream preliminary aircraft sizing disciplines (aerodynamics, propulsion, weight and stability), with the latter making it possible to assess morphing solutions against each others and also against conventional configurations. Several publications have been released (Appendix J).

The devised methodologies have been used to create two different codes using similar physical analysis models for the mainstream aircraft design disciplines, namely aerodynamics, propulsion, weight and static and dynamic stability.

Since the two codes are in-house developments, further enhancements will be easily implemented, in order to widen its applicability to other design cases with specific goals and constraints and to integrate other analysis and optimization algorithms, notwithstanding the replacement of the low fidelity models by higher fidelity ones. Two graphical user interfaces (GUIs) have been developed to enhance the user experience. For easiness of access, the two codes have been integrated in XFLR5 using its freeware source code in C++ language.

6.2.1 Parametric Aircraft Design Optimization (PARROT)

The PARROT methodology enables the user to optimize the wing layout (chord, span and shape (lift coefficient)) for two different optimization modes: surveillance mission or maximum payload. It enables the user to rapidly infer about the impact of changing these variables on the most relevant performance indicators, most notably, the energy consumption for the surveillance mission and useful payload lifted for the maximize payload mission.

Besides several benchmarking studies to validate the methodologies and their routines, the ”Olharapo III” UAV has been sized using the MTOP routine, including both its fixed-wing and morphing (variable span and variable camber flap) wing. The aircraft is expected to fly in the forthcoming months.

6.2.2 Multilevel Aircraft Design Optimization (MTOP)

The MTOP methodology relies on a distributed MDO approach combined with the use of morphing solutions making it possible to optimize the UAVs for a pre-defined mission profile. Once the performance targets for each mission stage are defined, the preliminary design stage of the aircraft becomes much faster, since the inherent iterative procedures of aeronautical design are significantly reduced. Furthermore, the overall process is far more efficient and more effective by taking into account how an appropriate weighting of each mission stage impacts the ultimate design goal.

Two modifications to the original Enhanced Collaborative Optimization (ECO) formulation have been presented, measured and discussed. A weighting coefficient will multiply the squared differences between the subspaces best attempt to match the system level targets and the system level targets themselves, enabling the user to assign a different relative weight to each subspace solution within the overall optimization. It has been shown how this add-in contributes to a duly weighted and consequently more robust solution. A dynamic compatibility parameter for the subspace level optimization is set to zero in the first system-level iterations and then it is steadily increased. This approach contributes to avoid an early compatibility between subspace optimizations that could jeopardize the quest for minimization.

In addition to this, the MTOP architecture features a collateral asset of being tailored for evaluating the usefulness of morphing concepts - each optimization subspace represents a different mission stage instead of representing the traditional design disciplines. Therefore, morphing-related design variables are local to each subspace. Likewise, global variables are locally optimized - with respect to each subspace - and globally made compatible using penalty parameters on the local objective functions.

6.3 Future Work

Future developments may include experimental, numerical and/or analytical correlations to estimate the structural and systems weight penalties of a variable camber flap (VCF) and variable pitch propeller (VPP) mechanisms. In addition, it would be interesting to improve the structural weight model estimates by building a prototype, and, if necessary adjust Raymer's experimental correlations. In the same way, adjust the variable span wing (VSW) structural weight penalty model by benchmarking it with data from a real prototype.

As for other design disciplines, it would be interesting to improve the propeller performance functions using test-bench results, as well as assessing the variable camber flap (VCF) effect on the aerodynamic coefficients in a more accurate way, perhaps using CFD.

Furthermore, it would be useful to test a wide range of mission profiles and requirements to determine trends for the kind of missions where the morphing technologies considered (VSW, VCF and VPP) are worthwhile.

Last but not least, it would certainly be interesting to widen the number of design variables being evaluated.

Bibliography

[1] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA, isbn: 978-1-60086-911-2 edition, 1989. 1, 37, 38, 51, 57, 59, 138

[2] Joaquim R. R. A. Martins and Andrew B. Lambe. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal*, 51:2049–2075, 2013. 1, 3, 5, 9, 11, 12, 13, 14, 15, 16, 21

[3] Olivier de Weck and Karen Willcox. Multidisciplinary system design optimization. In *Multidisciplinary System Design Optimization*. Massachusetts Institute of Tecnology, 2011. 1

[4] Autoridada Nacional de Aviação Civil. Condições de operação aplicáveis à utilização do espaço aéreo pelos sistemas de aeronaves civis pilotadas remotamente ("drones"). volume Regulamento n.º 1093/2016 of *Diário da República, 2.ª série — N.º 238 — 14 de dezembro de 2016*, 2016. 2, 28

[5] Advanced Aircraft Analysis. [http://www.darcorp.com/Software/AAA/]. Last accessed: 2015-08-13. 5

[6] CEASIOM website. [http://www.ceasiom.com]. Last accessed: 2015-08-13. 5

[7] XFLR5 website. [http://www.xflr5.com/xflr5.html]. Last accessed: 2015-07-28. 5

[8] Youngren Harold Drela, Mark. XFOIL 6.9 User Primer. *MIT/Aerocraft*, 2001. 5

[9] Emilio M. Botero, Andrew Wendorff, Timothy MacDonald, Anil Variyar, Julius M. Vegh, Trent W. Lukaczyk, Juan J. Alonso, Tarik H. Orra, and Carlos Ilario da Silva. SUAVE: An Open-Source Environment for Conceptual Vehicle Design and Optimization. *54th AIAA Aerospace Sciences Meeting, AIAA SciTech*, 2016. 6

[10] A. Suleman, F. Lau, J. Vale, and F. Afonso. Multidisciplinary Performance Based Optimization of Morphing Aircraft. *22nd AIAA/ASME/AHS Adaptive Structures Conference, AIAA SciTech Forum*, 2014. 6

[11] Z. Lyu and J. Martins. Aerodynamic Shape Optimization of an Adaptive Morphing Trailing-Edge Wing. *Journal of Aircraft*, 52(6):1951–1970, 2015. 6

[12] D. Burdette, G. Kenway, Z. Lyu, and J. Martins. Aerostructural Design Optimization of an Adaptive Morphing Trailing Edge Wing. *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum , AIAA 2015-1129*, pages 1289–1300, 2015. 6

[13] J. Vale, F. Afonso, A. Oliveira, F. Lau, and A. Suleman. On The Usage of Morphing Camber for Performance Improvement and Load Alleviation of a Small Scale Glider. In *6th EASN International Conference on Innovation in European Aeronautics Research, Porto, Portugal*, 2016. 7

[14] Satyajit S. Ghoman, Rakesh K. Kapania, P. C. Chen, Darius Sarhaddi, and D. H. Lee. Multifidelity, multistrategy, and multidisciplinary design optimization environment. *Journal of Aircraft*, 49(5):1255–1270, 2012. 9

[15] Gianfranco La Rocca and Michel J. L. van Tooren. Knowledge-Based Engineering Approach

to Support Aircraft Multidisciplinary Design Optimization. *Journal of Aircraft*, 46(6): 1875–1885, 2009. 9

[16] L. A. Schmit. Structural Design by Systematic Synthesis. *2nd Conference on Electronic Computation ASCE, New York*, page 105–132, 1960. 9

[17] L. A. Schmit. Structural Synthesis—Its Genesis and Development. *AIAA Journal*, 19(10): 1249–1263, 1981.

[18] L. A. Schmit. Structural Synthesis — Precursor and Catalyst. Recent Experiences in Multidisciplinary Analysis and Optimization. *NASA*, 19:1249–1263, 1984. 9

[19] R. T. Haftka. Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements. *NASA Langley Research Center, Hampton, VA*, 1973. 9

[20] R. T. Haftka. Optimization of Flexible Wing Structures Subject to Strength and Induced Drag Constraints. *AIAA Journal*, 14(8):1106–1977, 1977.

[21] R. T. Haftka and C. P. Shore. Approximate Methods for Combined Thermal/Structural Design. *NASA*, 1979. 9

[22] I. M. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski. Multidisciplinary Optimization Methods for Aircraft Preliminary Design. *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1994. 9

[23] N. E. Antoine and I. M. Kroo. Framework for Aircraft Conceptual Design and Environmental Performance Studies. *AIAA Journal*, 43(10):2100–2109, 2005. 9

[24] R. P. Henderson, J. R. R. A. Martins, and R. E. Perez. Aircraft Conceptual Design for Optimal Environmental Performance. *The Aeronautical Journal*, 116(1175):1–22, 2012. 9

[25] S. Wakayama. Blended-Wing-Body Optimization Problem Setup. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, volume ISBN: 10.2514/6.2000-4740 of *AIAA Paper 2000-4740*. AIAA/USAF/NASA/ISSMO, 2000. 9

[26] O. Gur, M. Mason Bhatia, J. W. Schetz, R. Kapania, and T. Nam. Development of framework for truss-braced wing conceptual mdo. *AIAA Paper 2010-2754*, pages 2049–2075, 2010. 9

[27] R. T. Haftka, J. Sobieszczanski-Sobieski, and S. L. Padula. On Options for Interdisciplinary Analysis and Design Optimization. *Structural Optimization*, 4:65–74, 1992. 9

[28] E. J. Cramer, J. E. Dennis Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994. 9, 12, 13

[29] R. J. Balling and J. Sobieszczanski-Sobieski. Optimization of Coupled Systems: A Critical Overview of Approaches. *AIAA Journal*, 34(1):pp. 6–17, 1996. 9, 13

[30] N. Alexandrov and M. Y. Hussaini. Multidisciplinary Design Optimization: State-of-the-Art. *SIAM*, 1997. 9, 13

[31] I. M. Kroo. ”MDO for Large-Scale Design” Multidisciplinary Design Optimization: State-of-the-Art. *edited by N. Alexandrov and M. Y. Hussaini SIAM*, page pp. 22–44, 1997. 9

[32] J. Sobieszczanski-Sobieski and R. T. Haftka. Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments. *Structural Optimization*, 14(1):pp. 1–23, 1997. 9

[33] R. T. Haftka. Simultaneous Analysis and Design. *AIAA Journal*, 23(7):1099–1103, 1985. 12

[34] J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, 2013. 13

[35] Niekamp R. Matthies, H. G. and J. Steindorf. Algorithms for strong coupling procedures. *Comput. Methods Appl. Mech. Eng.*, page 2028–2049, 2006. 13

[36] Ismail Farajpour and Sez Atamturktur. Optimization-Based Strong Coupling Procedure for Partitioned Analysis. *Journal of Computing in Civil Engineering*, 26, 2012. 13

[37] Jaroslaw Sobieszczanski-Sobieski. Sensitivity of Complex, Internally Coupled Systems. *AIAA Journal*, 28(1):153–160, 1990. 14

[38] Alonso J. J. Martins, J. R. R. A. and J. J. Reuther. A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design. *Optimization and Engineering*, 6(1): 33–62, 2005. 14

[39] J. Sobieszczanski-Sobieski. Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems. *NASA Langley Research Center, Hampton, VA*, 1988. 14

[40] J. Shankar, C. J. Ribbens, R. T. Haftka, and L. T. Watson. Computational Study of a Nonhierarchical Decomposition Algorithm. *Computational Optimization and Applications*, 2:273–293, 1993. 15

[41] R. E. Perez, H. H. T. Liu, and K. Behdinan. Evaluation of Multidisciplinary Optimization Approaches for Aircraft Conceptual Design. *Proceedings of the 10th AIAA/ISSMO Muldisiciplinary Analysis and Optimization Conference, Albany, NY,Aug. 2004*, 2004. 15, 18, 23

[42] S. I. Yi, J. K. Shin, and G. J. Park. Comparison of MDO Methods with Mathematical Examples. *Structural and Multidisciplinary Optimization*, 39:391–402, 2008. 15, 18, 21, 23, 104

[43] N. P. Tedford and J. R. R. A. Martins. Benchmarking Multidisciplinary Design Optimization Algorithms. *Optimization and Engineering*, 11:159–183, 2010. 15, 23

[44] A. J. de Wit and F. van Keulen. Numerical Comparison of Multi-level Optimization Techniques. *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Honolulu, HI*, 2007. 15, 16, 18, 21, 23, 104

[45] H. M. Kim. Target Cascading in Optimal System Design. Master's thesis, University of Michiga, 2001. 15

[46] H. M. Kim, N. F. Michelena, P. Y. Papalambros, and T. Jiang. Target Cascading in Optimal System Design. *Journal of Aircraft Design*, 125(3):474–480, 2003. 15

[47] J. T. Allison, B. Roth, M. Kokkolaras, I. M. Kroo, and P. Y. Papalambros. Aircraft family design using decomposition-based methods. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2006-6950. AIAA 2006-6950, 2006. 16, 17

[48] J. T. Allison, D. Walsh, M. Kokkolaras, P. Y. Papalambros, and M. Cartmell. Analytical target cascading in aircraft design. In *44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, doi: 10.2514/6.2006-1325. AIAA 2006-1325, 2006. 16

[49] S. Tosserams, M. Kokkolaras, L. F. P. Etman, and J. E. Rooda. A Nonhierarchical Formulation of Analytical Target Cascading. *Journal of Mechanical Design*, 132(5), 2010. 16

[50] R. D. Braun. Collaborative Optimization: An Architecture for Large-Scale Distributed Design. Master's thesis, Stanford University, 1996. 16

[51] Robert Braun, Peter Gage, Ilan Kroo, and Ian Sobieski. Implementation and performance issues in collaborative optimization. *Proceedings of the 6th AIAAUSAFNASAISSMO Multidisciplinary Analysis and Optimization Symposium, Bellevue, WA, Sept. 1996, AIAA 1996-4017*, 22, 1996. 16

[52] A. V. DeMiguel and W. Murray. An analysis of collaborative optimization methods. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA*, doi: 10.2514/6.2000-4720. AIAA 2000-4720, 2000. 17

[53] N. M. Alexandrov and R. M. Lewis. Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design. *AIAA Journal*, 40(2):301–309, 2002. 17

[54] R. D. Braun, A. A. Moore, and I. M. Kroo. Collaborative approach to launch vehicle design. *Journal of Spacecraft*, 34(4):478–486, 1997. 17

[55] G. Cai, J. Fang, Y. Zheng, X. Tong, J. Chen, and J. Wang. Optimization of System Parameters for Liquid Rocket Engines with Gas-Generator Cycles. *Journal of Propulsion and Power*, 26(1):113–119, 2010. 17

[56] I. M. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski. Multidisciplinary Optimization Methods for Aircraft Preliminary Design. *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page pp. 22–44, 1994. 17

[57] Brian Roth and Ilan Kroo. Enhanced Collaborative Optimization: Application to an Analytic Test Problem and Aircraft Design. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC, Canada*, 2008. 17, 23, 104, 105, 107, 109, 111, 112

[58] Brian Roth and Ilan Kroo. Enhanced Collaborative Optimization: A Decomposition-Based Method for Multidisciplinary Design. *ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008*, 2008. 17, 18, 104, 105, 107, 109

[59] S. Kodiyalam. Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Phase I. *Tech. Rep. CR-1998-208716, NASA*, 1998. 18, 23, 104

[60] Jaroslaw Sobieszczanski-Sobieski, Jeremy Agte S., and Robert R. Jr. Sandusky. Bi-level Integrated System Synthesis (BLISS). *NASA/TM-1998-208715*, 1998. 18

[61] Jaroslaw Sobieszczanski-Sobieski, Troy D. Altus, Matthew Phillips, and Robert Sandusky. Bilevel Integrated System Synthesis for Concurrent and Distributed Processing. *AIAA Journal*, 41(10), 2003. 18, 19

[62] J. Ahn and J. H. Kwon. An Efficient Strategy for Reliability-Based Multidisciplinary Design Optimization Using BLISS. *Structural and Multidisciplinary Optimization*, 31:363–372, 2006. 18

[63] P. A. Legresley and J. J. Alonso. Improving the performance of design decomposition methods with pod. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2004-4465. AIAA 2004-4465, 2004. 18

[64] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:pp. 539–575, 1993. 18

[65] N. F. Brown and J. R. Olds. Evaluation of Multidisciplinary Optimization Techniques Applied to a Reusable Launch Vehicle. *Journal of Spacecraft and Rockets*, 43(6):1289–1300, 2006. 19, 23

[66] V. DeMiguel and W. Murray. A Local Convergence Analysis of Bilevel Decomposition Algorithms. *Optimization and Engineering*, 7(2):99–133, 2006. 20

[67] S. Tosserams, L. F. P. Etman, and J. E. Rooda. An Augmented Lagrangian Decomposition Method for Quasiseparable Problems in MDO. *Structural and Multidisciplinary Optimization*, 34:211–227, 2007. 20

[68] B. Liu, R. T. Haftka, and L. T. Watson. Global-Local Structural Optimization Using Response Surfaces of Local Optimization Margins. *Structural and Multidisciplinary Optimization*, 27 (5):352–359, 2004. 21

[69] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet. *Journal of Aircraft*, 41(3):523–530, 2004. 21

[70] I. R. Chittick and J. R. R. A. Martins. Optimization and Engineering. *SIAM Journal on Optimization*, 10(1):133–152, 2009. 21, 23

[71] C. Marriage. Automatic Implementation of Multidisciplinary Design Optimization Architectures Using MDO. Master's thesis, University of Toronto, 2008. 23

[72] B. D. Roth. Aircraft Family Design Using Enhanced Collaborative Optimization. Master's thesis, Stanford University, 2008. 23, 104

[73] Garret N. Vanderplaats. *Fortran 952003 for Scientists & Engineers*. VRD, isbn-10: 0944956041 edition, 2007. 24

[74] Paul Witherell, Sundar Krishnamurty, and Ian R. Grosse. Ontologies for Supporting Engineering Design Optimization. *J. Comput. Inf. Sci. Eng 7(2)*, pages 141–150, 2006. 25

[75] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. (The Addison-Wesley series in artificial intelligence) First Edition (US) First Printing Edition, isbn-10: 0201055945 edition, 1984. 25

[76] Metaheuristics. [https://cs.stackexchange.com/questions/74898/are-most-metaheuristic-algorithms-different-metaphors-for-the-same-method]. Last accessed: 2017-10-05. 26

[77] Terrence A. Weisshaar. Morphing aircraft technology – new shapes for aircraft design, [www.dtic.mil/cgi-bin/gettrdoc?ad=ada479821], 2006. 27, 30

[78] Barbarino S., O. Bilgen, R. M. Ajaj, M.I. Friswell, and D.J. Inman. A review of morphing aircraft. *Journal of Intelligent Material Systems and Structures*, 22:pp. 823–877, 2011. 27, 28, 29, 30, 31, 113, 117

[79] S. Joshi, Z. Tidwell, W. Crossley, and S. Ramakrishnan. Comparison of Morphing Wing Strategies Based Upon Aircraft Performance Impacts. *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2004. 27

[80] Riga Declaration on Remotely Piloted Aircraft (drones) - Framing the Future of Aviation. European Union, 2015. 28

[81] Press Release – DOT and FAA Propose New Rules for Small Unmanned Aircraft Systems. [https://www.faa.gov/news/press_releases/news_story.cfm?newsId=18295]. Last accessed: 2015-09-20. 28

[82] Rafic Ajaj, Andy Keane, Christopher Beaverstock, Michael Friswell, and Inman Daniel. Morphing aircraft: The need for a new design philosophy. In *Ankara International Aerospace Conference*, doi: 10.1016/j.ast.2015.11.039. AIAC-2013-054, 2013. 28

[83] Hamid Basaeri, Aghil Yousefi-Koma, M.R. Zakerzadeh, and Seyed Mohtasebi. Experimental study of a bio-inspired robotic morphing wing mechanism actuated by shape memory alloy wires. 24, 11 2014. 29, 30

[84] J. Manzo, E. Garcia, A. Wickenheiser, and G.C. Horner. Design of a Shape-Memory Alloy Actuated Macro-scale Morphing Aircraft Mechanism. *Proceedings of SPIE Smart Structures and Materials 2005: Smart Structures and Integrated Systems*, 5764:744, 2005. 31

[85] A.Y.N. Sofla, S.A. Meguid, K.T. Tan, and W.K. Yeo. Shape Morphing of Aircraft Wing: Status and Challenges. *Materials and Design*, 2010. 31

[86] Eric Gamble, Dwain Terell, and Rich De Francesco. Nozzle Selection and Design Criteria. *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2004. 31

[87] David A. Neal, Matthew G. Good, Christopher O. Johnston, Harry H Robertshaw, William H. Mason, and Daniel J. Inman. Design and Wind-tunnel analysis of a fully adaptive aircraft configuration. *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2004. 31

[88] J. S. Flanagan, R. C. Strutzenberg, R.B. Myers, and J. E. Rodrian. Development and flight testing of a morphing aircraft, the nextgen mfx-1. *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007. 31

[89] J. Vale, F. Lau, A. Suleman, and Gamboa. Multidisciplinary Design Optimization of a Morphing Wing for an Experimental UAV. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2006-7131, doi: 10.2514/6.2006-7131. AIAA/ISSMO, 2006. 32

[90] J. Vale, F. Lau, A. Suleman, and P. Gamboa. Optimization of a Morphing Wing Based on Coupled Aerodynamic and Structural Constraints. In *3rd AIAA MultiDisciplinary Design Optimization Specialists Conference*, AIAA 2007-1890, doi: 10.2514/1.39016, page 27. AIAA, 2007. 32

[91] P. Gamboa, P. Aleixo, J. Vale, F. Lau, and A. Suleman. Design and testing of a morphing wing for an experimental uav. In *RTO-MP-AVT-146, Neuilly-sur-Seine, France*, doi: Accession number - ADA478692. RTO-MP-AVT-146, 2007. 32

[92] A. Leite, J. Vale, F. Lau, and A. Suleman. ‘development of morphing strategies for flight demonstrator rpv. In *50th AIAA/ASME/ASCE/AHS/ASCStructures, Structural Dynamics, and Materials Conference*, doi: 10.2514/6.2009-2134. AIAA 2009-2134, 2009. 32

[93] Stephen J. Chapman. *Fortran 952003 for Scientists & Engineers*. McGraw Hill, isbn-10: 0073191574 edition, 2008. 33

[94] Thomas C. Corke. *Design of Aircraft*. Prentice Hall, isbn: isbn-13: 978-0-13-089234-8 edition, 2002. 38

[95] Ray Prouty. *Helicopter Aerodynamics Volume I*. Eagle Eye Solutions, LLC (2007), isbn: 978-0979263811 edition, 2007. 45

[96] Brian R. Gyles. PropSelector Help. *Gyles AeroDesign*, 1999. 46

[97] Mark Drela. QPROP User Guide, Version 1.21. *MIT*, 2007. 46

[98] Martin Hepperle. JavaProp Users Guide. *MIT*, 2015. 46

[99] Pedro V. Gamboa, J. Vale, F. J. P. Lau, and A. Suleman. Optimization of a Morphing Wing Based on Coupled Aerodynamic and Structural Constraints. *AIAA Journall*, 47(9): 2087–2104, 2009. 51

[100] Pedro F. Albuquerque, Pedro V. Gamboa, and Miguel A. Silvestre. Multidisciplinary and Multilevel Design Methodology of Unmanned Aerial Vehicles Using Enhanced Collaborative Optimization. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 9(4):470–479, 2015. 51, 109, 111

[101] Rui Cunha. Structural Analysis of a Variable-span Wing-box. Master's thesis, University of Beira Interior, 2014. 52, 114, 116

[102] Bernard Etkin. *Dynamics of Flight - Stability and Control*. John Wukey & Cons, Inc, ISBN-10: 0471034185, 1996. 56, 58, 59, 60

[103] Egbert Torenbeek. *Synthesis of Subsonic Airplane Design*. Delft University Press, Kluwer Academic Publishers, Dordrecht/Boston/London, 1982, ISBN-10: 9024727243, 1996. 57

[104] The USAF Stability And Control Digital Datcom. *Users Manual. USAF Technical Report AFFDL-TR-79-3032 (AD A086557)*, I, 1979. 58, 59

[105] Pedro V. Gamboa, Miguel A. Silvestre, and Pedro F. Albuquerque. Aircraft Design Methodology Using Span and Mean Wing Chord as Main Design Parameters. *International Conference on Engineering of University of Beira Interior, Covilhã, Portugal*, 2013. 77

[106] Air Cargo Challenge Regulations. *EUROAVIA Stuttgart*, 31, 2015. 86

[107] International Standards Organization. International Standard Atmosphere, ISO 2533:1975. 1975. 89

[108] S. Kodiyalam and C. Yuan. Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Part 2. *Tech. Rep. CR-2000-210313, NASA*, 2000. 104

[109] Xiao Mi, Qiu Haobo, Gao Liang, Shao Xinyu, and Chu Xuezheng. An Enhanced Collaborative Optimization Methodology for Multidisciplinary Design Optimization. *The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, P.R.China*, 2010. 105

[110] P. Spellucci. A SQP method for general nonlinear programs using only equality constrained subproblems. *Math. Prog.*, 82:413–448, 1998. 106

[111] P. Spellucci. A new technique for inconsistent problems in the SQP method. Math. Meth. of Oper. Res. *Math. Meth. of Oper. Res.*, 47:355–400, 1998. 106

[112] Jian L. Zhou, Andr L. Tits, and Craig T. Lawrence. User's Guide for FFSQP Version 3.7: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints1. *Electrical Engineering Department and Institute for Systems Research University of Maryland, College Park, MD 20742*, 1992. 106

[113] FP7 CHANGE Consortium. http://change.tekever.com/homepage. Last accessed: 2016-06-26. 113, 116

[114] J. R. C. Mestrinho, P. V. Gamboa, and P. D. Santos. Design Optimization of a Variable-Span Morphing Wing for a Small UAV. *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials (and co-located) Conferences, Denver, Colorado, USA*, 2011. 114, 116

[115] J.M.I. Felício, P. D. Santos, P. V. Gamboa, and M. A. R. Silvestre. Evaluation of a Variable-Span Morphing Wing for a Small UAV. *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials (and co-located) Conferences, Denver, Colorado, USA*, 2011. 116

[116] J. Katz and A. Plotkin. Low-Speed Aerodynamics. Master's thesis, Cambridge University Press, 2nd ed, 2001. 119

[117] R. Mukherjee and A. Gopalarathnam. An Iterative Decambering Approach for Post-Stall Prediction of Wing Characteristics using known Section Data. *41st AIAA Aerospace Sciences Meeting, Reno, Nevada*, 2003. 119

[118] Jan Roskam. *Aircraft Design - Part I: Preliminary Sizing of Airplanes*. DARCorporation, isbn-10: 188488542x edition, 1985. 138

Appendix A

CHANGE Consortium

Table A.1: CHANGE consortium responsibilities.

Partner	Responsibility
Tekever	Coordinator of the project; Flight validation testing. Manufacture and integration of the morphing wing.
DLR	Provide conceptual design and detailed design for a morphing concept. Analysis of the final model of the UAV.
ARA	Morphing layout assessment software. Design of aerodynamic target wing shapes. Experimental validation. Final performance data analysis CFD and wind tunnel.
UBI	Development of span change strategies; Detailed design of the morphing. system prototype. Integration of the wing with all the morphing systems. Dissemination and technology watch.
Cranfield University	Morphing layout assessment software. Structural and detailed design of the morphing wing. Modeling and simulation of skin material. Dissemination and exploitation.
Swansea University	Morphing optimization. The design of morphing wings using active twist.
INVENT	Conduct material selection for the morphing skin and wing. Manufacture tooling and morphing skin. Conduct material performance evaluation.
Middle East Technical University	Development of camber change strategies. Development of twist change strategies. Detailed design of the morphing system prototype. Integration of the wing with all the morphing systems. Dissemination and technology watch through workshops. Web design and maintenance; Wind tunnel tests.
Technical University of Delft	Development of application scenarios for the morphing aircraft. Development of morphing aircraft design software. Testing and validation of the morphing aircraft design software. Structural and detailed design of loitering morphing aircraft. Wind tunnel tests of the morphing aircraft models.

Appendix B

Continuous Variable Camber Flap Geometry

In order to estimate the flapped airfoil aerodynamic coefficients as a function of the Reynolds number and angle-of-attack, one must know the airfoil's geometry. As such, a swift way of computing the airfoil geometry based on the original airfoil for several combinations of flap chord (c_f) and flap deflection (δ_f) so that the aerodynamic analyzes can take place and the aerodynamic coefficients be estimated at each optimization point using a multilinear interpolation approach. Accordingly, the airfoil lift coefficient, drag coefficient and pitching moment coefficient are thus a function of the Reynolds number (Re), the airfoil angle-of-attack (α), the flap chord (c_f) and the flap deflection (δ_f). Let us start by determining the deflected coordinates of the upper surface of the airfoil of Figure B.1.

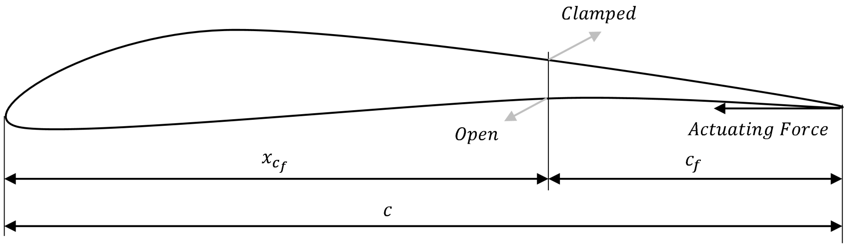


Figure B.1: Variable camber flap airfoil geometry.

Assumptions:

- Assume upper surface of flap to be straight;
- Actuation force produces a linearly varying bending moment;
- Deflection of the flap of the form of Equation (B.1):

$$\delta_f = K(3L - x)x^2 \tag{B.1}$$

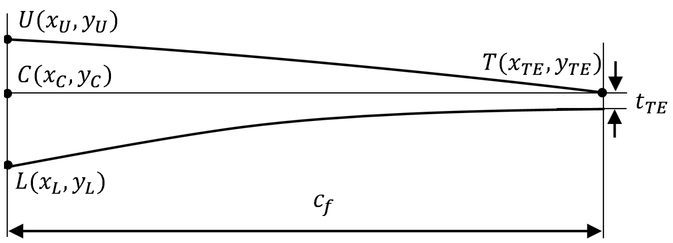


Figure B.2: Airfoil Geometry Definition.

Geometry Definition, Equation (B.2)**:**

$$\begin{cases} x_{c_f} = c - c_f \\ x_U = x_L = x_C = x_{c_f} \\ z_C = 0 \\ x_{TE} = c \\ z_{TE} \geq 0 \\ t_{TE} \geq 0 \\ l_{UT} = constant \end{cases} \tag{B.2}$$

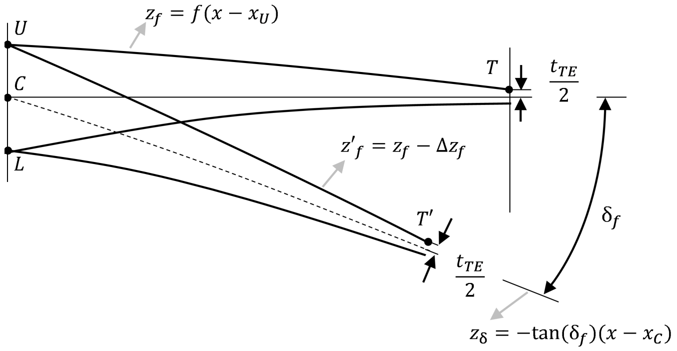


Figure B.3: Flapped and Unflapped Airfoil Geometry Definition.

The flap (z) coordinate is given by Equation (B.3):

$$\Delta z_f = K\left[3(x_T - x_U) - (x - x_U)\right](x - x_U)^2 \tag{B.3}$$

The deflected flap (z) coordinate, (z'_f), is given by Equation (B.4):

$$z'_f(x'_T) = z_\delta(x'_T) \tag{B.4}$$

The deflected flap (x) coordinate, (x'_f), is given by Equation (B.5):

$$x'_f = x_U + \frac{x'_T - x_U}{x_T - x_U}(x - x_U) \tag{B.5}$$

Thus, the deflected flap (z) coordinate is given by Equation (B.6):

$$z'_f = z_f - K\left[3(x_T - x_U) - (x - x_U)](x - x_U)^2\right] \tag{B.6}$$

The goal is thus to determine K so that $l_{UT} = l'_{UT}$

1. Determine the coordinates of $U(x_U, z_U)$;
2. Determine the arc length l_{UT}, Equation (B.7);

$$l_{UT} \approx \sum_{i=1}^{n} \sqrt{(z_{i+1} - z_i)^2 + (x_{i+1} - x_i)^2} \tag{B.7}$$

3. Determine the coordinates of $C(x_C, z_C)$, Equation (B.8);

$$x_C = 0 \qquad z_C = 0 \tag{B.8}$$

4. Define the line of deflection (z_δ), Equation (B.9);

$$z_\delta = -tan(\delta_f).(x - x_C) \tag{B.9}$$

5. Assume x'_T;
6. Assume K;
7. Define the curve of deflected upper surface (z'_f), Equation (B.10);

$$z'_f = z - K[3(x_T - x_U) - (x - x_U)](x - x_U)^2 \tag{B.10}$$

8. Determine K^{new}, Equation (B.11);

$$\begin{cases} z_\delta(x'_T) = -tan(\delta_f)(x'_T - x_C) \\ z'_f(x_T) = z - K[3(x_T - x_U) - (x_T - x_U)](x_T - x_U)^2 \end{cases} \tag{B.11}$$

   if $z_\delta(x'_T) = z'_f(x_T)$ exit;
   else, go to 6;

9. Determine ${x'_T}^{new}$, Equation (B.12);

$$\begin{cases} x'_f = x_U + \frac{x'_T - x_U}{x_T - x_U}(x - x_U) \\ z'_f = z_f - K[3(x_T - x_U) - (x_T - x_U)](x - x_U)^2 \end{cases} \tag{B.12}$$

$$l'_{UT} = \sum_{i=1}^{n} \sqrt{(z'_{i+1} - z'_i)^2 + (x'_{i+1} - x'_i)^2} \tag{B.13}$$

   if $l'_{UT} = l_{UT}$ exit;
   else, go to 5;

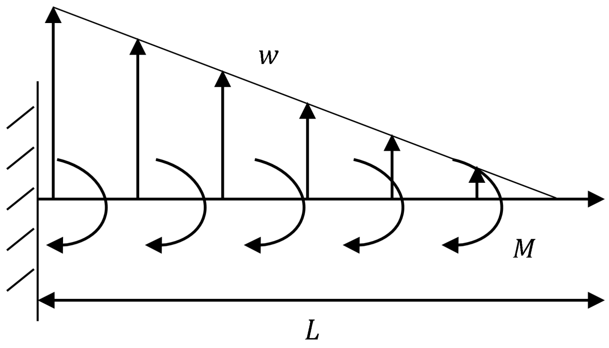


Figure B.4: Cantilever beam under a linearly distributed bending moment and linearly distributed load.

Deflection of a cantilever beam under a linearly distributed bending moment and linearly distributed load:

(a) Linearly distributed bending moment, Equation (B.14):

$$\begin{cases} M = M_0 \quad at \quad x = 0 \\ M = 0 \quad at \quad x = L \\ M = M_0 + \frac{0-M_0}{L}x - 0 = \frac{M_0}{L}(L-x) \end{cases} \tag{B.14}$$

(b) Linearly distributed load, Equation (B.15):

$$\begin{cases} w = w_0 \quad at \quad x = 0 \\ w = 0 \quad at \quad x = L \\ w = w_0 + \frac{0-w_0}{L}x - 0 = \frac{w_0}{L}(L-x) \\ M = -\int_0^x w.xdx = -\int_0^x \frac{w_0}{L}(Lx - x^2)dx \\ M = -\frac{w_0}{L}\left(\frac{L}{2}x^2 - \frac{x^3}{3}\right) \end{cases} \tag{B.15}$$

(c) Curvature (δ''), rotation (δ') and deflection (δ), Equation (B.16):

$$\begin{cases} \delta'' = \frac{M}{EI} = \frac{M_0}{EIL}(L-x) - \frac{w_0}{6EIL}(3Lx^2 - 2x^3) \\ \delta' = \frac{M_0}{EIL}\left(Lx - \frac{x^2}{2}\right) - \frac{w_0}{6EIL}\left(Lx^3 - \frac{x^4}{2}\right) + A \\ \delta = \frac{M_0}{EIL}\left(\frac{Lx^2}{2} - \frac{x^3}{6}\right) - \frac{w_0}{6EIL}\left(\frac{Lx^4}{4} - \frac{x^5}{10}\right) + Ax + B \end{cases} \tag{B.16}$$

(d) Boundary conditions, Equation (B.17):

$$\begin{cases} \delta' = 0; \quad at \quad x = 0 \Rightarrow A = 0 \\ \delta = 0; \quad at \quad x = 0 \Rightarrow B = 0 \end{cases} \tag{B.17}$$

(e) Rotation, Equation (B.18):

$$\delta' = \frac{M_0}{12EIL}\left[\frac{6M_0}{w_0}(2L-x) - x^3(2L-x)\right] \tag{B.18}$$

(f) Deflection, Equation (B.19):

$$\delta = \frac{w_0}{120EIL}\left[\frac{20M_0}{w_0}x^2(3L-x) - x^4(5L-2x)\right] \tag{B.19}$$

As for the lower airfoil surface (x') coordinate, a proportional repositioning of the points between (x_L) and (TE) has been made based on the original airfoil distribution. In what concerns to the (z) coordinate, assuming an uniform length of ($L = x(n) - x_L$) and the deflection ($\delta(x)$) with ($x = x(i) - x_L$) comes Equation (B.20).

$$z'(i) = z(i) - \delta(x(i)) - [z(n) - \delta(L) - z_{T_{EL}}].\frac{x'(i) - x_L}{x_{T_{EL}} - x_L} \tag{B.20}$$

, where ($x_{T_{EL}}$) and ($z_{T_{EL}}$) are the trailing edge deflected coordinates.

Appendix C

Multilinear Interpolation

If the given function is not linear, then the linearly interpolated value will be an approximation. If the array has more than two dimensions, the value sought will be at a point within the interior of the corresponding polytope, with the number of mathematical operations to be performed depending on the number of problem dimensions. Not only does the quality of the interpolated solution depend on the number of scattered points for which the function is known, but also on the function behavior. The more linear behavior it exhibits, the better will be the fitting between the actual function and the interpolated values.

Before generalizing the multilinear interpolation approach, let us look at the simplest case of a linear interpolation. Let the superscript (i) denote the variable dimension and subscript (j) denote the point's reference number, (x_j^i).

Figure C.1 depicts a typical scheme with two points for which the function is known $(f(x_0^1))$ and $(f(x_1^1))$ and the point in between these two for which the linear interpolation is required $(f(x_2^1))$.

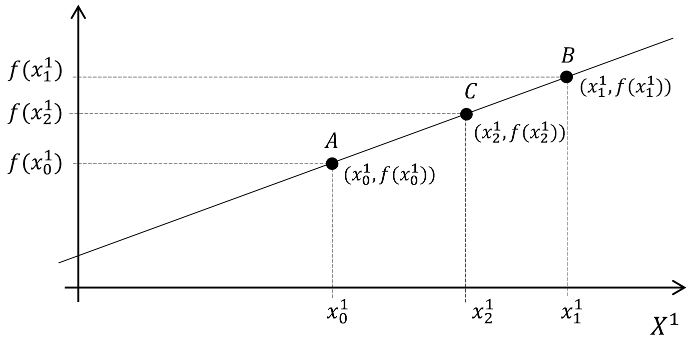


Figure C.1: Linear interpolation.

Equation (C.1) shows the linear interpolation used for computing $f(x_2^1))$.

$$f(x_2^1) = f(x_1^1) + \Big[f(x_1^1) - f(x_0^1)\Big] \frac{(x_2^1 - x_0^1)}{(x_1^1 - x_0^1)} \tag{C.1}$$

If the goal was to obtain a bi-linear interpolation the problem to solve would be something alike the scheme shown in Figure C.2. The function value is known at the four points A, B, C and D, $(f(x_0^1, x_1^2))$, $(f(x_1^1, x_1^2))$, $(f(x_0^1, x_0^2))$ and $(f(x_1^1, x_0^2))$, respectively and the bi-linear interpolation goal is to obtain an approximate value for the same function at point E, $(f(x_2^1, x_2^2))$.

Contrarily to the approach adopted for the linear interpolation, the bi-linear inter-

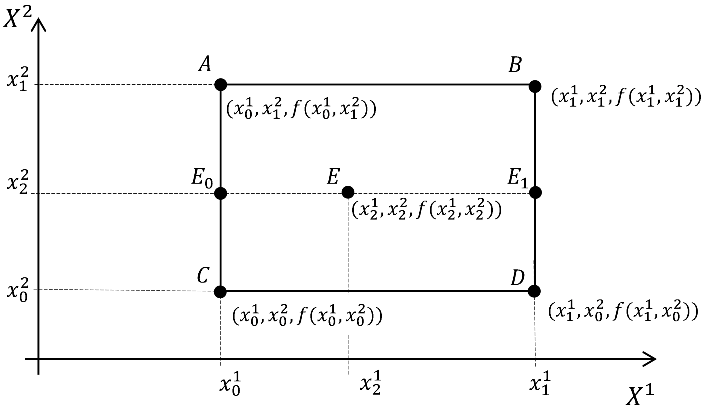

X^2
x_1^2
x_2^2
x_0^2
A
B
$(x_0^1, x_1^2, f(x_0^1, x_1^2))$
$(x_1^1, x_1^2, f(x_1^1, x_1^2))$
E_0
E
E_1
$(x_2^1, x_2^2, f(x_2^1, x_2^2))$
C
D
$(x_0^1, x_0^2, f(x_0^1, x_0^2))$
$(x_1^1, x_0^2, f(x_1^1, x_0^2))$
x_0^1
x_2^1
x_1^1
X^1

Figure C.2: Bilinear interpolation.

polation is divided into two steps, each of which representing a linear interpolation. The first step consists of computing the function value at points E_0 and E_1, each being obtained via a linear interpolation, Equations (C.2) and (C.3).

$$f(x_0^1, x_2^2) = f(x_0^1, x_0^2) + \left[f(x_0^1, x_1^2) - f(x_0^1, x_0^2)\right] \frac{(x_2^2 - x_0^2)}{(x_1^2 - x_0^2)} \tag{C.2}$$

$$f(x_1^1, x_2^2) = f(x_1^1, x_0^2) + \left[f(x_1^1, x_1^2) - f(x_1^1, x_0^2)\right] \frac{(x_2^2 - x_0^2)}{(x_1^2 - x_0^2)} \tag{C.3}$$

The second step is to use these two new points to estimate the function value at point E, in the same way as it is done in the linear interpolation, Equation (C.4).

$$f(x_2^1, x_2^2) = f(x_0^1, x_2^2) + \left[f(x_1^1, x_2^2) - f(x_0^1, x_2^2)\right] \frac{(x_2^1 - x_0^1)}{(x_1^1 - x_0^1)} \tag{C.4}$$

As the number of variables increases the graphical representation becomes harder to understand and therefore it is important to generalize what has been stated for linear and bi-linear interpolation to multivariable interpolation with an arbitrary number of variables. The objective is to estimate $f(x_2^1, x_2^2, x_2^3, ..., x_2^n)$. The first step is to determine the coordinates of the known points which are closer to the point $(x_2^1, x_2^2, x_2^3, ..., x_2^n)$. These points which define the vertices of a polytope surrounding the point at which the function aims to be interpolated.

$$\begin{array}{c} (x_0^1, x_0^2, ..., x_0^n) \\ (x_0^1, x_0^2, ..., x_1^n) \\ \vdots \\ (x_1^1, x_1^2, ..., x_0^n) \\ (x_1^1, x_1^2, ..., x_1^n) \end{array} \tag{C.5}$$

Following a similar approach to the one presented for the linear and bi-linear interpolation, and beginning by interpolating variable (x_2^1):

$$\begin{aligned}
f(x_2^1, x_0^2, ..., x_0^n) &= f(x_0^1, x_0^2, ..., x_0^n) + \left[f(x_1^1, x_0^2, ..., x_0^n) - f(x_0^1, x_0^2, ..., x_0^n)\right] \tfrac{(x_2^1 - x_0^1)}{(x_1^1 - x_0^1)} \\
f(x_2^1, x_0^2, ..., x_1^n) &= f(x_0^1, x_0^2, ..., x_1^n) + \left[f(x_1^1, x_0^2, ..., x_1^n) - f(x_0^1, x_0^2, ..., x_1^n)\right] \tfrac{(x_2^1 - x_0^1)}{(x_1^1 - x_0^1)} \\
&\vdots \\
f(x_2^1, x_1^2, ..., x_0^n) &= f(x_0^1, x_1^2, ..., x_0^n) + \left[f(x_1^1, x_1^2, ..., x_0^n) - f(x_0^1, x_1^2, ..., x_0^n)\right] \tfrac{(x_2^1 - x_0^1)}{(x_1^1 - x_0^1)} \\
f(x_2^1, x_1^2, ..., x_1^n) &= f(x_0^1, x_1^2, ..., x_1^n) + \left[f(x_1^1, x_1^2, ..., x_1^n) - f(x_0^1, x_1^2, ..., x_1^n)\right] \tfrac{(x_2^1 - x_0^1)}{(x_1^1 - x_0^1)}
\end{aligned} \tag{C.6}$$

Interpolation of variable (x_2^2):

$$\begin{aligned}
f(x_2^1, x_2^2, ..., x_0^n) &= f(x_2^1, x_0^2, ..., x_0^n) + \Big[f(x_2^1, x_1^2, ..., x_0^n) - f(x_2^1, x_0^2, ..., x_0^n)\Big] \frac{(x_2^2 - x_0^2)}{(x_1^2 - x_0^2)} \\
&\vdots \\
f(x_2^1, x_2^2, ..., x_1^n) &= f(x_2^1, x_0^2, ..., x_1^n) + \Big[f(x_2^1, x_1^2, ..., x_1^n) - f(x_2^1, x_0^2, ..., x_1^n)\Big] \frac{(x_2^2 - x_0^2)}{(x_1^2 - x_0^2)}
\end{aligned} \tag{C.7}$$

Interpolation of variable (x_2^n):

$$\begin{aligned}
f(x_2^1, x_2^2, \dots, x_2^n) = &f(x_2^1, x_2^2, \dots, x_1^n) + \\
&+ \Big[f(x_2^1, x_2^2, \dots, x_1^n) - f(x_2^1, x_2^2, \dots, x_0^n)\Big] \frac{(x_2^n - x_0^n)}{(x_1^n - x_0^n)}
\end{aligned} \tag{C.8}$$

The number of points delimiting this multidimensional polytope is equal to 2^n, with n being the number of problem dimensions and the number of linear interpolations to be performed is equal to $\left(\sum\limits_{i=1}^{n} 2^i\right)$.

Appendix D

PARROT GUI Users' Manual

PARROT

Parametric Aircraft Design Optimization (PARROT) User's Manual

Aerospace Sciences Department
Faculty of Engineering
University of Beira Interior
PORTUGAL

Authors:
Pedro F. Albuquerque
Filipe R. Fraqueiro
Professor Pedro V. Gamboa

Covilhã, September, 2016

ii

Contents

iv

List of Figures

vi

List of Tables

Chapter 1

Introduction

1.1 Foreword

This document is the Parametric AiRcRaft design OpTimization (PARROT) graphical user interface [1] user's manual. Its purpose is to provide an outlook on the methods used in the calculations, and to provide assistance for the less intuitive aspects of this software's handling.

The PARROT software has been developed at the Department of Aerospace Sciences of the University of Beira Interior [2] with the ultimate goal of fostering a more efficient, effective and comprehensive preliminary design optimization of unmanned aerial vehicles. The appropriate sizing of an aircraft is essential to produce a high performance design. As it is known, size and mass also have a close correlation with costs. The design methodology developed is based on an extensive parametric study developed in-house in a spreadsheet [3] which has been converted into a Fortran code for the sake of efficiency, easiness of handling and modularity. This code optimizes the wing size for one of two different optimization modes: minimum energy or maximum payload. Whereas in the former the goal might be to maximize the flight range or endurance, the latter's objective is to maximize the useful payload lifted.

1.2 Methodology

One of the most particular features of the PARROT software, is that it is a mission-based design optimization, which means that the aircraft is optimized taking into account its complete mission profile, rather than for a particular mission stage. This methodology's primary design parameters are the wing span (b) and the wing mean chord ($\bar{c}$). As part of the input data set, the user will define the number of different wing layouts to be studied as well as the domain of variation of the wing mean chord and wingspan. Other design parameters can be the wing airfoil cruise lift coefficient (C_l), the center of gravity (CG) position, the lifting surfaces' airfoils, the motor/engine and the propeller size, among many others. The PARROT's users have to choose between two different optimization modes: minimum energy or maximum payload, depending on the mission specific objectives. Furthermore, the user has to define the mission profile itself and the respective performance requirements. The code will then generate several different wing geometries (different wing mean chord and wingspan combinations), which can be assessed against each other using parametric plots representation. Therefore, the designer (user) can make more informed decisions at the preliminary design phase, which will significantly contribute to getting closer to the optimum solution in a fewer number of iterations. Constraints include specified performance criteria, like maximum take-off distance, climb rate, bank angle, cruise velocity and alike. Internally, the routine comprises several disciplinary models [2], including low fidelity models for the aerodynamics (via XFOIL), for the propulsion – with the possibility of choosing either a combustion engine or an electrical motor -, for the weight – with a particular focus on delivering a fair structural weight estimate - and for the stability – where the horizontal and vertical empennages are sized. Each of these disciplinary models will require a significant number of inputs which are explained in Chapter 5 of the current manual. Each different wing layout will then follow the user defined mission pattern and as the required energy is computed the design take-off weight (DTOW) will be iterated accordingly. Depending on the optimization mode – either maximizing the take-off weight or minimizing the energy consumption – the optimization will behave differently. The following two subsections will briefly highlight their most prominent differences.

1.2.1 Maximum Payload

The first optimization mode that can be selected is the one that aims to maximize the payload lifted. Firstly, the user defines the energy and the systems weights (W_{ene}, W_{sys}) together with the intended mission profile and specific energy. Then, a first guess for the DTOW will get the optimization routine started. A scattered number of wingspan (b) and mean wing chord ($\bar{c}$) combinations will be generated, in accordance with the user settings. The PARROT code will then start the sequential analysis of each mission stage according to the mission profile performance targets.

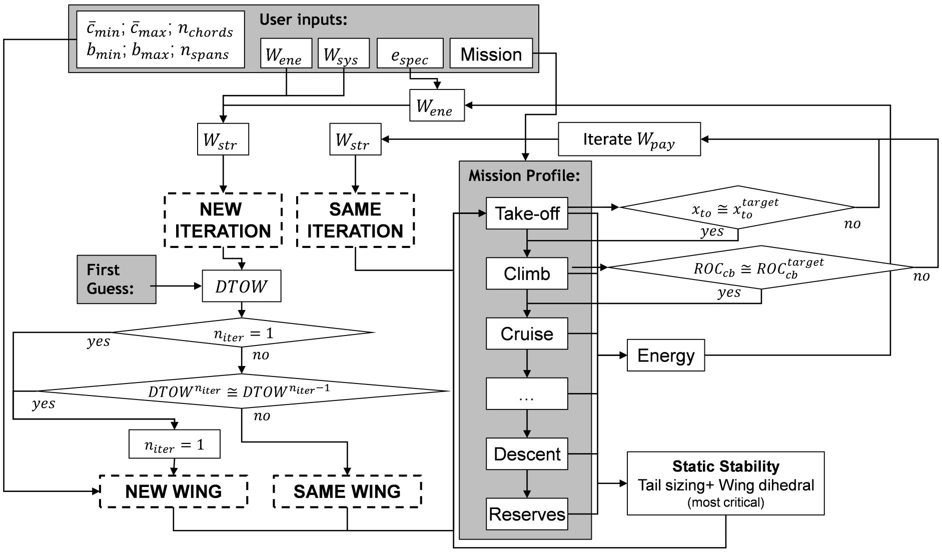


Figure 1.1: Scheme featuring the system level iterative procedure for the PARROT code in the maximize payload mode.

Since this optimization mode aims at lifting the maximum possible payload, the iterative optimization procedure is significantly different from the minimize energy optimization mode as far as the DTOW convergence is concerned. It is assumed that all mission profiles will start with a take-off stage followed by a climb stage, these being precisely the two most relevant stages in limiting the MTOW and consequently the maximum payload weight (W_{pay}), since the take-off stage limits the available take-off distance, whereas the climb stage possibly includes some limitation regarding the rate-of-climb (RoC). As for the take-off, all the available distance will be used. If that happens not to be the case, the payload weight will be corrected accordingly, increased if the required take-off distance is lower than the one available and decreased otherwise. For the first climb stage, the user defined rate-of-climb shall be met - which can in some cases be mandatory as far as clearance of obstacles is concerned. However, if the available propulsive power is not enough to meet it, the payload weight is reduced accordingly. In the end, the most critical of these two requirements (take-off distance and rate-of-climb) will dictate the maximum payload.

1.2.2 Minimum Energy

The second optimization mode aims at minimizing the vehicle's energy consumption and may be interesting if the goal is to optimize a surveillance or cruise type of mission. In this mode, the user defines the payload and the systems weights (W_{pay}, W_{sys}), together with the intended mission profile and the batteries or fuel specific energy. Then, a first guess for the DTOW will get the optimization routine started.

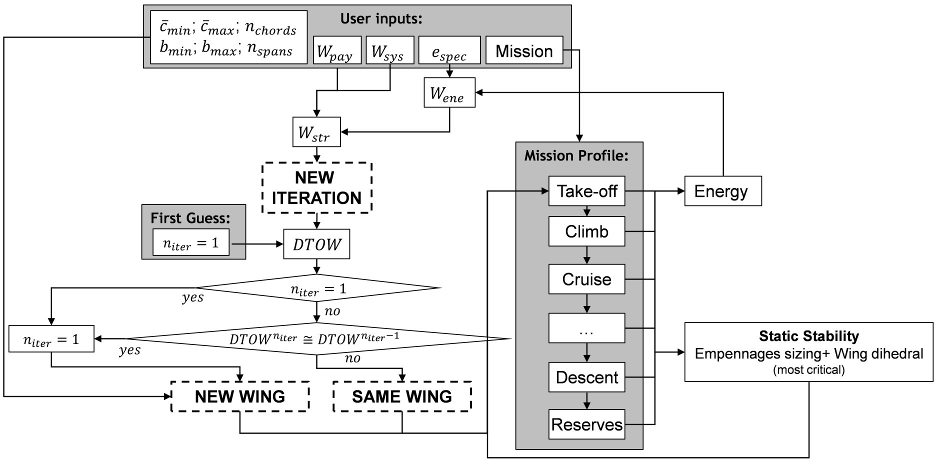


Figure 1.2: Scheme featuring the system level iterative procedure for the PARROT code in the minimize energy mode.

At each iteration, the user defined mission profile is simulated and the total energy (E) required for performing such mission is known, which means that the energy weight (W_{ene}) can be computed from the specific energy. At this point, it is possible to re-run the structural weight model and consequently updating the DTOW accordingly so that the following iteration can take place. Once the DTOW converges or the maximum number of iterations is reached, the iterative procedure comes to an end and the following wing mean chord and wingspan combination is analyzed. Once all the wings have been computed, the designer can evaluate all outputs to determine the impact of varying the wing mean chord ($\bar{c}$) and/or the wingspan (b) on the most relevant design functions.

1.3 Structure

The programming language used was Fortran 90 using Microsoft Visual Studio 2010. Within the program's working directory there is a folder called «input_data» where all the input data files that the user should feed or update prior to running the PARROT can be found. Likewise, all the output files are created in a folder called «output_data», also within the working directory.

Chapter 2

PARROT Graphical User Interface

2.1 GUI Implementation

The development of PARROT's graphical user interface (GUI) was made using the open source XFLR5 GUI, which is programmed in C++ language. The main reasons for using the XFLR5 framework were the fact that it is an open source code, easy to handle and already having expedite methods for the aerodynamic analysis of airfoils (using XFOIL).

2.2 GUI Handling

After opening the standard XFLR5 File Menu, the options shown in (Figure 2.1) are displayed. Within the red rectangle, it is possible to see a new sub-menu called *Aircraft Optimization*, which shall be selected. Alternatively, the user can choose the keyboard shortcut (Ctrl+P).

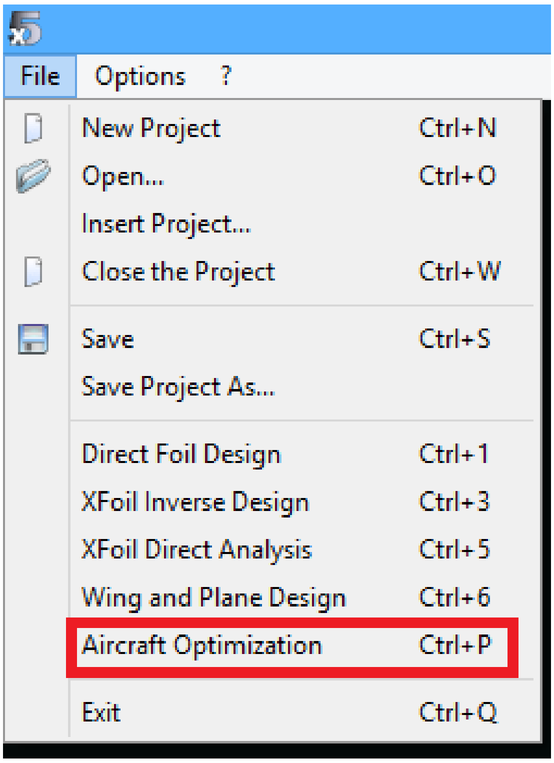

File
Options
?
New Project Ctrl+N
Open... Ctrl+O
Insert Project...
Close the Project Ctrl+W
Save Ctrl+S
Save Project As...
Direct Foil Design Ctrl+1
XFoil Inverse Design Ctrl+3
XFoil Direct Analysis Ctrl+5
Wing and Plane Design Ctrl+6
Aircraft Optimization Ctrl+P
Exit Ctrl+Q

Figure 2.1: XFLR5 Sub-Menus.

Once the *Aircraft Optimization* option has been chosen, it is possible to visualize two new options: *Analysis* and *Flap*. If the *Analysis* separator is chosen, two aircraft optimization codes are available: PARROT and MTOP (Figure 2.16). To get to the PARROT code main menu, the user shall choose the PARROT option. Alternatively, the user can choose the keyboard shortcut (Ctrl+R).

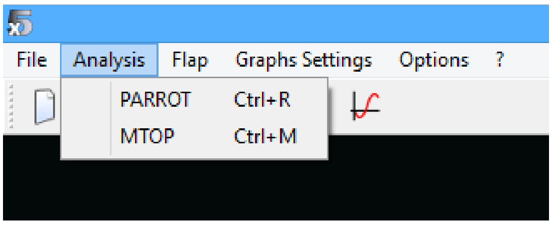
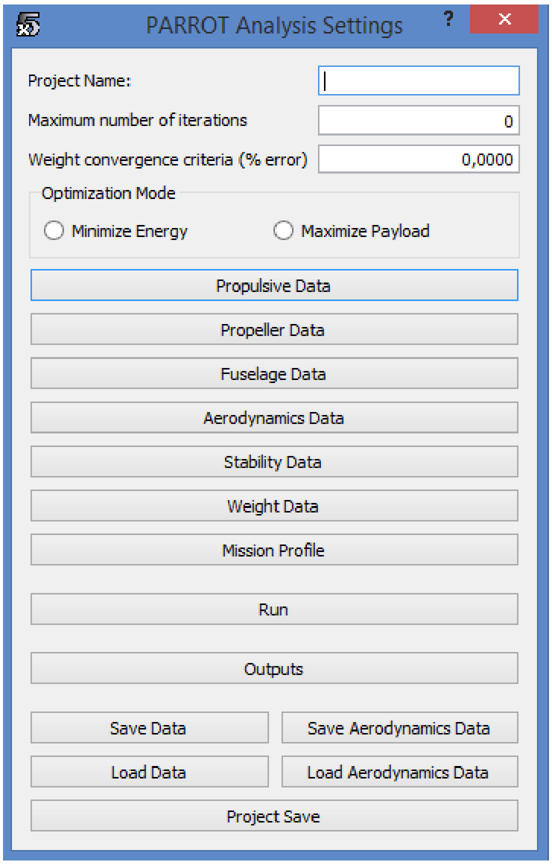


Figure 2.2: Analysis Types.

At this point, the user can have an overview of PARROT's main menu (Figure 2.3).



Project Name:

Maximum number of iterations 0

Weight convergence criteria (% error) 0,0000

Optimization Mode

○ Minimize Energy ○ Maximize Payload

Figure 2.3: PARROT Main Menu.

At the top of this main menu window (Figure 2.3), the user can load the general inputs listed in Table 2.1.

Table 2.1: General Inputs.

Label	Description
Project Name	name of the project
Maximum number of iterations	maximum number of acceptable iterations of the vehicle's design take-off weight
Weight convergence criteria (% error)	weight convergence criteria as a percentage of the design take-off weight
Mission Type	mission optimization mode, minimize energy or a maximize payload

It is also possible to visualize the various input categories required for the PARROT analysis to be run:

- Propulsive Data;
- Propeller Data;
- Fuselage Data;
- Aerodynamics Data;
- Stability Data;
- Weight Data;
- Mission Profile.

Furthermore, it is possible to visualize the *Analyze* button, which can trigger the PARROT optimization as soon as all the input data is loaded. Below this option, the *Outputs* button can be found. Its selection makes it possible to visualize the output data. Lastly, it is possible to visualize the different options which either enable data storage in ".txt" files or input data load using ".txt" files:

- Save Data;
- Save Aerodynamics Data;
- Load Data;
- Load Aerodynamics Data;
- Save Project.

The forthcoming sections describe the different disciplinary input data windows as well as the mission profile window.

2.2.1 Inputs

2.2.1.1 Propulsion

After clicking on *Propulsive Data* button, a new window will open. Firstly, the user has to decide between two types of propulsion: Electric Motor (Figure 2.4) or Combustion Engine (Figure 2.5). There are some input parameters that are independent of the propulsion type. These are described in Tables 2.2 and 2.3.

Table 2.2: Propulsion Limitations.

LIMIT ENGINE RPM /MOTOR CURRENT	
YES	NO

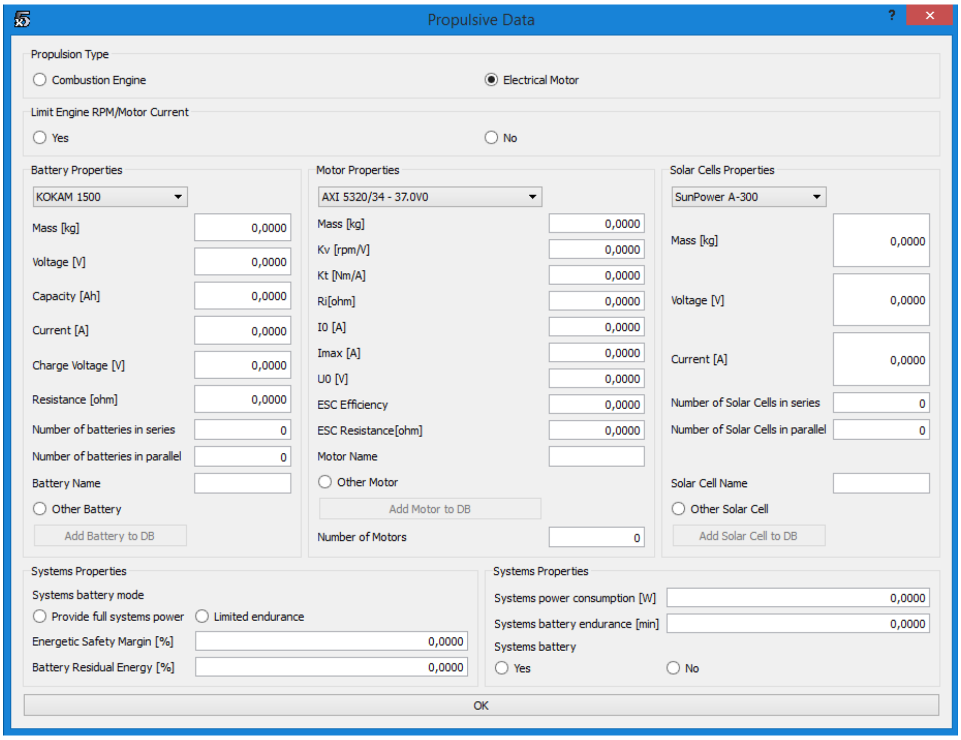
Table 2.3: Systems properties inputs description.

Label	Description
Systems battery mode	provide full systems power / limited endurance
Energetic Safety Margin [%]	-
Battery Residual Energy [%]	-
Systems battery	Yes / No
Systems power consumption [W]	-
Systems battery endurance [min]	-

If the user chooses the Electric Motor, the requested input data includes the Battery, Motor and Solar Cells properties (Figure 2.4). These inputs are explained in Table 2.4.

Conversely, if the aircraft has a combustion engine the user's task becomes limited to loading the engine data. The respective input window is featured in Figure 2.5, whereas its variables are described in 2.5.

In this last window it is possible to select the engine/motor, the battery and the solar cell directly from a database. Alternatively, if the user intends to use a different engine/motor, battery and/or solar cell, the respective toggle button *Other Engine/Motor/Battery/Solar Cell* of Figures 2.4 and 2.5 shall be used and the corresponding data shall be directly loaded on the GUI window. If the user intends to add his/her own values to the database, after loading the data, he/she should click on the button Other Engine/Motor/Battery/Solar Cell to DB (Figures 2.4 and 2.5).



Propulsive Data

Propulsion Type

○ Combustion Engine ● Electrical Motor

Limit Engine RPM/Motor Current

○ Yes ○ No

Battery Properties

KOKAM 1500

Mass [kg]	0,0000
Voltage [V]	0,0000
Capacity [Ah]	0,0000
Current [A]	0,0000
Charge Voltage [V]	0,0000
Resistance [ohm]	0,0000
Number of batteries in series	0
Number of batteries in parallel	0
Battery Name	

○ Other Battery

Motor Properties

AXI 5320/34 - 37.0V0

Mass [kg]	0,0000
Kv [rpm/V]	0,0000
Kt [Nm/A]	0,0000
Ri[ohm]	0,0000
I0 [A]	0,0000
Imax [A]	0,0000
U0 [V]	0,0000
ESC Efficiency	0,0000
ESC Resistance[ohm]	0,0000
Motor Name	
Number of Motors	0

○ Other Motor

Solar Cells Properties

SunPower A-300

Mass [kg]	0,0000
Voltage [V]	0,0000
Current [A]	0,0000
Number of Solar Cells in series	0
Number of Solar Cells in parallel	0
Solar Cell Name	

○ Other Solar Cell

Systems Properties

Systems battery mode

○ Provide full systems power ○ Limited endurance

Energetic Safety Margin [%]	0,0000
Battery Residual Energy [%]	0,0000

Systems Properties

Systems power consumption [W]	0,0000
Systems battery endurance [min]	0,0000

Systems battery

○ Yes ○ No

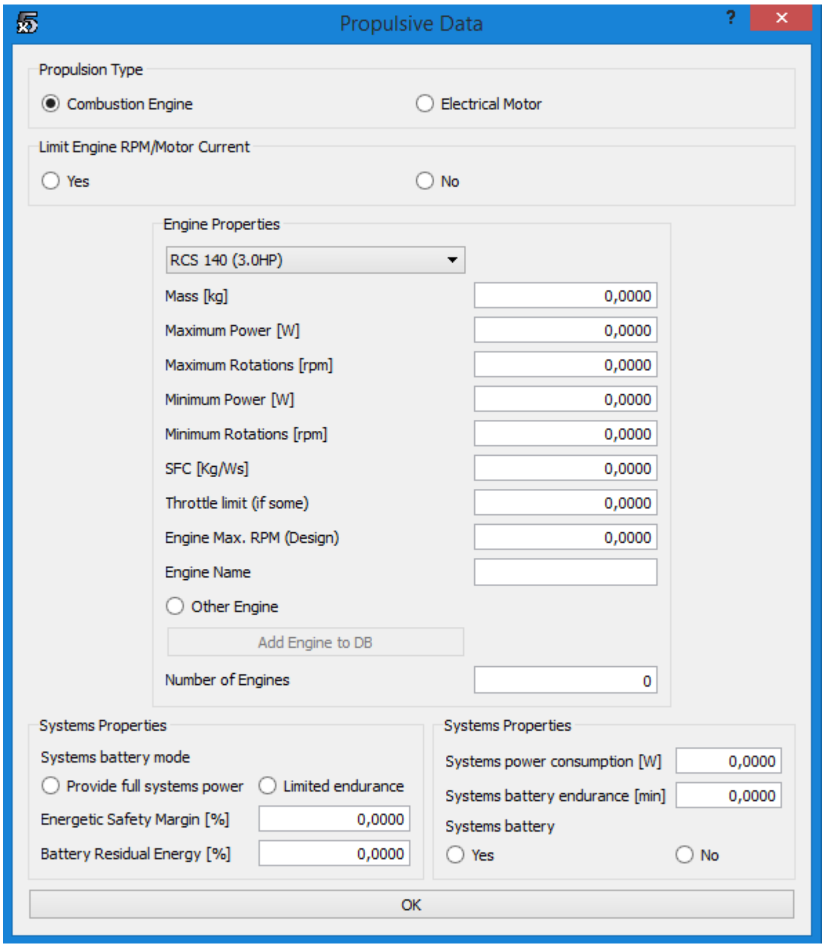
Figure 2.4: Electric motor inputs.

Table 2.4: Electric motor: battery, motor and solar cell inputs description.

Label	Description
Battery	
Mass [kg]	mass
Voltage [V]	voltage
Capacity [Ah]	capacity
Current [A]	electric current
Charge Voltage [V]	charge voltage
Number of batteries in series	-
Number of batteries in parallel	-
Battery Name	-
Motor	
Mass [kg]	mass
Kv [rpm/V]	speed constant
Kt [Nm/A]	torque constant
Ri [ohm]	internal resistance
I0 [A]	no load electric current
Imax [A]	maximum electric current
U0 [V]	no load voltage
ESC efficiency	electronic speed controller efficiency
ESC resistance [ohm]	electronic speed controller resistance
Motor Name	-
Number of Motors	-
Solar Cells	
Mass [kg]	-
Voltage [V]	-
Current [A]	electric nominal current
Number of solar cells in series	-
Number of solar cells in parallel	-
Solar Cell Name	-

Table 2.5: Engine inputs description.

Label	Description
Mass [kg]	engine mass
Maximum Power [W]	-
Maximum Rotations [rpm]	-
Minimum Power [W]	-
Minimum Rotations [rpm]	-
Specific Fuel Consumption (SFC) [kg/Ws]	-
Throttle limit (if some)	-
Engine maximum rpm (Design)	-
Engine Name	-
Number of Engines	-



Propulsive Data

Propulsion Type

● Combustion Engine ○ Electrical Motor

Limit Engine RPM/Motor Current

○ Yes ○ No

Engine Properties

RCS 140 (3.0HP)

Mass [kg]	0,0000
Maximum Power [W]	0,0000
Maximum Rotations [rpm]	0,0000
Minimum Power [W]	0,0000
Minimum Rotations [rpm]	0,0000
SFC [Kg/Ws]	0,0000
Throttle limit (if some)	0,0000
Engine Max. RPM (Design)	0,0000
Engine Name	

○ Other Engine

Add Engine to DB

Number of Engines: 0

Systems Properties

Systems battery mode

○ Provide full systems power ○ Limited endurance

Energetic Safety Margin [%]	0,0000
Battery Residual Energy [%]	0,0000

Systems Properties

Systems power consumption [W]	0,0000
Systems battery endurance [min]	0,0000

Systems battery

○ Yes ○ No

OK

Figure 2.5: Combustion Engine Inputs.

2.2.1.2 Propeller

Besides the propulsion data, which includes all the relevant information regarding the engine(s) or motor(s) in use, propeller performance data is needed. On a fixed pitch propeller, this will typically be dependent on three variables, the propeller diameter, the propeller pitch and the propeller advance ratio. These performance analysis typically uses two different curves, the power coefficient versus advance ratio and the propeller efficiency versus the advance ratio. Accordingly, the PARROT code enables the user to choose between three different ways of loading and handling such data:

- **Mode 1 - Linear Interpolation:** the user loads the ”input_propeller.txt” and the different data sets are consecutively linearly interpolated accordingly;
- **Mode 2 - Least-square method:** the user chooses the polynomial approximation degree of the power coefficient and propeller efficiency curves and loads the ”input_propeller.txt”. The data sets will thereafter be used for generating their respective polynomial approximation using the least-square method;
- **Mode 3 - Polynomial approximation:** the user chooses the polynomial approximation degree of the power coefficient and propeller efficiency curves and loads the polynomial coefficients directly. The user can choose to load an arbitrary number of curves for each of these two variables, each for a different RPM, provided he/she loads an equal number of curves for both variables.

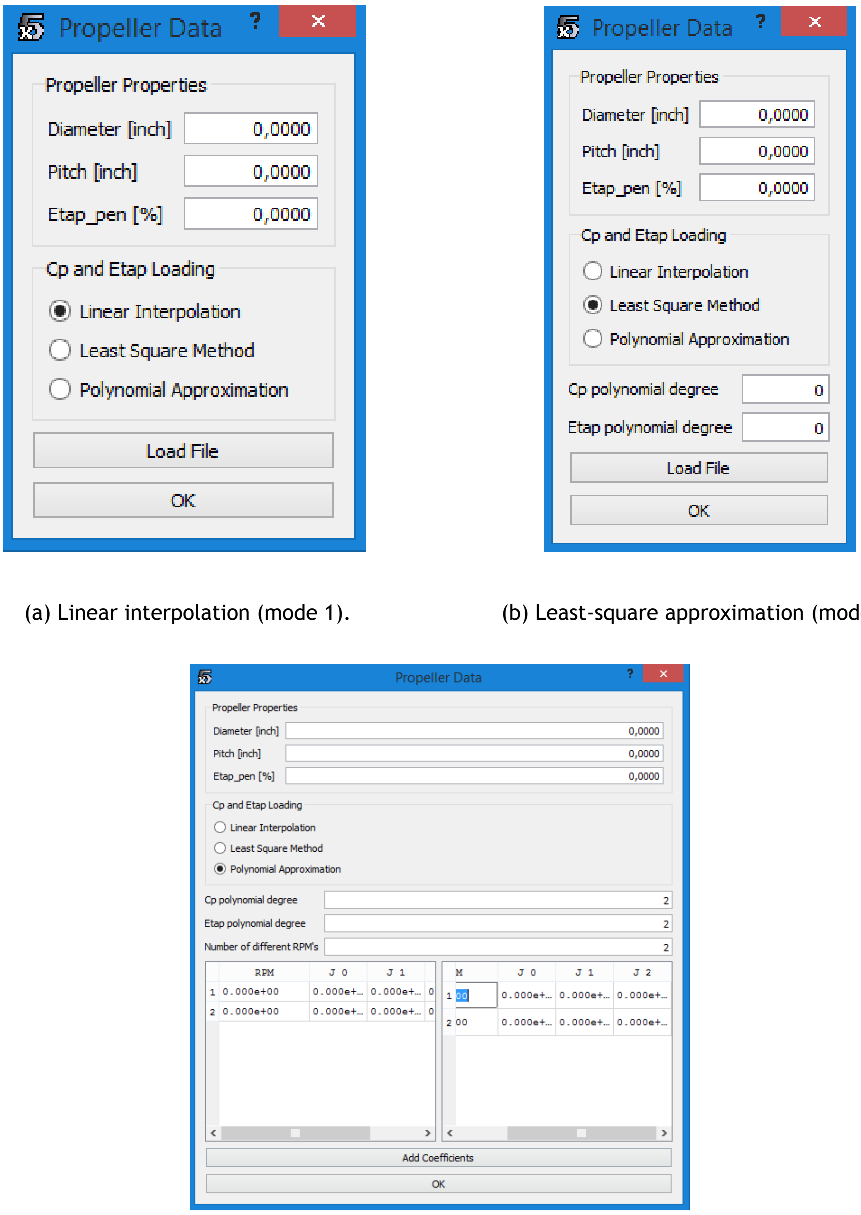
A graphical representation of the Propeller Data window is shown in Figure 2.6a (mode 1) and Figure 2.6b (mode 2) and Figure 2.6c (mode 3).

As already explained, if mode 1 or mode 2 is selected an ”input_propeller.txt” file must be loaded. This file must have the format shown in Figure 2.7, where the number of advance ratio (J) combinations, propeller efficiency (η_p) and propeller power coefficient (C_p) must be equal to the number of points. Furthermore, is the user has (n) data sets, each corresponding to a different RPM, the number of RPM must equal (n) and the respective data must follow the first one, repeating the file format after the horizontal dashed line, inclusively, (n) times.

The remaining data is the same regardless of the data handling mode in use and is described in Table 2.6.

Table 2.6: Propeller inputs description.

Label	Description
Diameter [inch]	propeller diameter
Pitch [inch]	propeller pitch
$\eta_{p_{pen}}$ [%]	efficiency penalty parameter for propeller propulsive efficiency
C_p and η_p Loading	data loading and handling mode



Propeller Data

Propeller Properties

Diameter [inch] 0,0000

Pitch [inch] 0,0000

Etap_pen [%] 0,0000

Cp and Etap Loading

● Linear Interpolation

○ Least Square Method

○ Polynomial Approximation

(a) Linear interpolation (mode 1).

Propeller Data

Propeller Properties

Diameter [inch] 0,0000

Pitch [inch] 0,0000

Etap_pen [%] 0,0000

Cp and Etap Loading

○ Linear Interpolation

● Least Square Method

○ Polynomial Approximation

Cp polynomial degree 0

Etap polynomial degree 0

(b) Least-square approximation (mode 2).

Propeller Data

Propeller Properties

Diameter [inch] 0,0000

Pitch [inch] 0,0000

Etap_pen [%] 0,0000

Cp and Etap Loading

○ Linear Interpolation

○ Least Square Method

● Polynomial Approximation

Cp polynomial degree 2

Etap polynomial degree 2

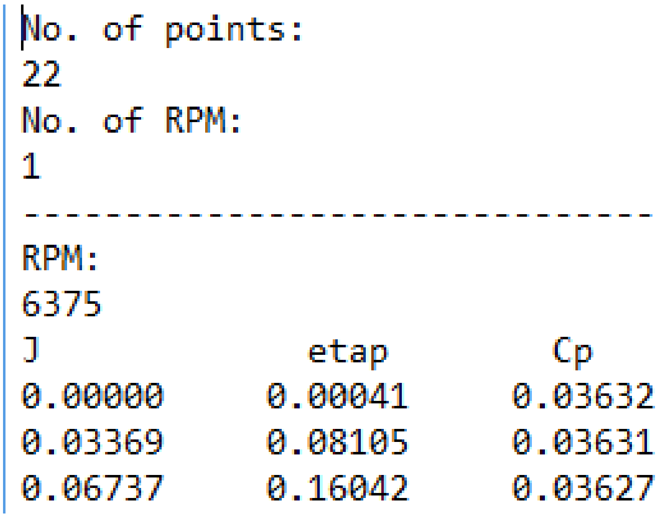
Number of different RPM's 2

	RPM	J 0	J 1	
1	0.000e+00	0.000e+...	0.000e+...	0
2	0.000e+00	0.000e+...	0.000e+...	0

	M	J 0	J 1	J 2
1	00	0.000e+...	0.000e+...	0.000e+...
2	00	0.000e+...	0.000e+...	0.000e+...

(c) Polynomial approximation (mode 3).

Figure 2.6: Propeller input modes.

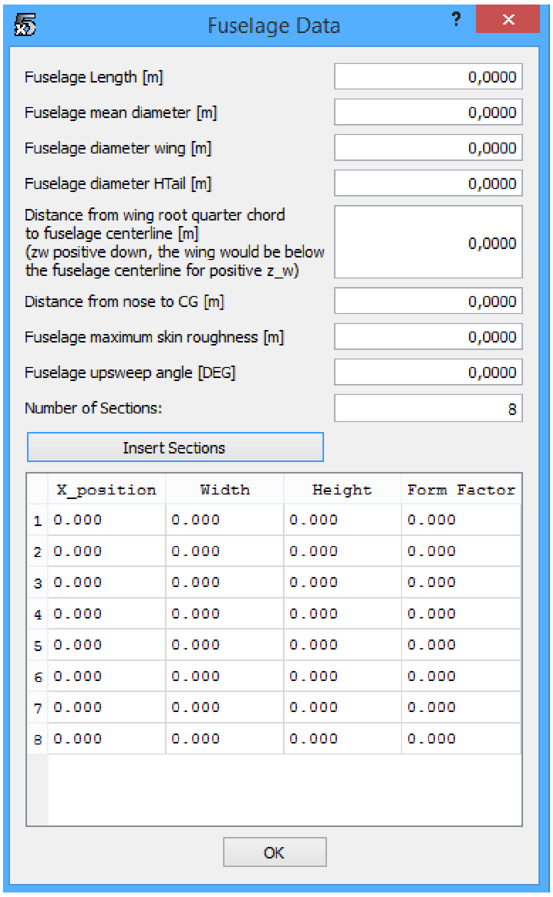


```
No. of points:
22
No. of RPM:
1
--------------------------------
RPM:
6375
J              etap         Cp
0.00000      0.00041      0.03632
0.03369      0.08105      0.03631
0.06737      0.16042      0.03627
```

Figure 2.7: Mode 1 and 2 propeller performance loading format.

2.2.1.3 Fuselage

The Fuselage Data window is presented in Figure 2.8.



Fuselage Length [m]	0,0000
Fuselage mean diameter [m]	0,0000
Fuselage diameter wing [m]	0,0000
Fuselage diameter HTail [m]	0,0000
Distance from wing root quarter chord to fuselage centerline [m] (zw positive down, the wing would be below the fuselage centerline for positive z_w)	0,0000
Distance from nose to CG [m]	0,0000
Fuselage maximum skin roughness [m]	0,0000
Fuselage upsweep angle [DEG]	0,0000
Number of Sections:	8

	X_position	Width	Height	Form Factor
1	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000

Figure 2.8: Fuselage inputs.

The input parameters are described in Table 2.7. After writing the number of fuselage sections on the respective tab, it is required to click on *Insert Sections* so that a table featuring the: lengthwise x-position (starting at the nose and increasing rearwards), width, height and form factor for each section is displayed.

Table 2.7: Fuselage inputs description.

Label	Description
Fuselage length [m]	-
Fuselage mean diameter [m]	-
Fuselage diameter wing [m]	fuselage diameter at the wing's root
Fuselage diameter ht [m]	fuselage diameter at the horizontal stabilizer's root
Vertical distance from wing root quarter chord to fuselage centerline [m]	assumed positive downwards, (the wing would be below the fuselage centerline for positive z_w)
Distance from nose to CG [m]	-
Fuselage maximum skin roughness [m]	-
Fuselage upsweep angle [deg]	-
Number of sections	number of fuselage lengthwise sections
X-position	lengthwise section position measured from the fuselage nose backwards
Width	fuselage width at each section
Height	fuselage height at each section
Form Factor	section form factor - equals unity for a rectangular cross section or a lower value for rounded corners

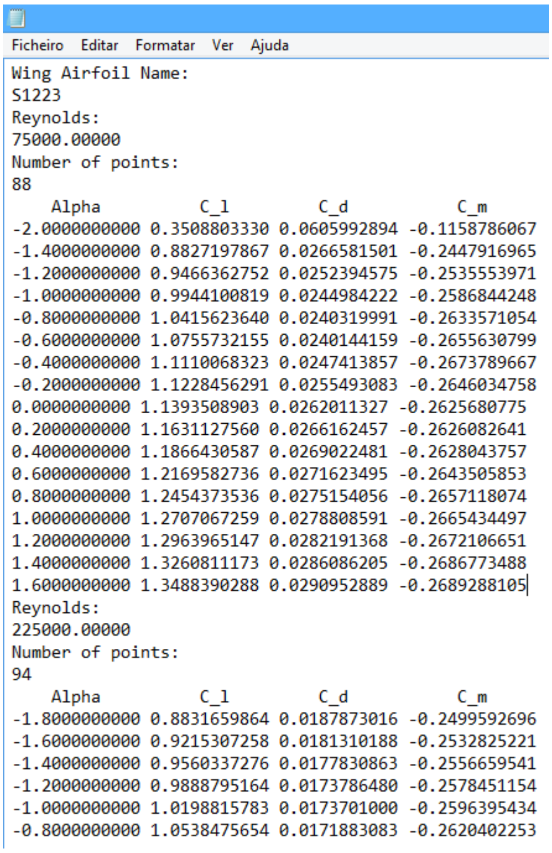
2.2.1.4 Aerodynamics

The PARROT code requires the lifting surfaces (wing, horizontal stabilizer and vertical stabilizer) aerodynamic coefficients to be known for a set of different Reynolds number and angles-of-attack so that these can be accurately estimated at any flow condition. These aerodynamic coefficients are the airfoils':

- Lift coefficient (C_l);
- Drag coefficient (C_d);
- Pitching moment coefficient (C_m).

There are two ways of feeding the PARROT code with these coefficients:

- **Option 1:** The user clicks on *Load Aerodynamic Data* and the GUI will consecutively ask the user to select a ".txt" with the airfoil's aerodynamic data for the wing, the horizontal stabilizer and the vertical stabilizer, respectively. Each ".txt" file must have the format shown in Figure 2.9.

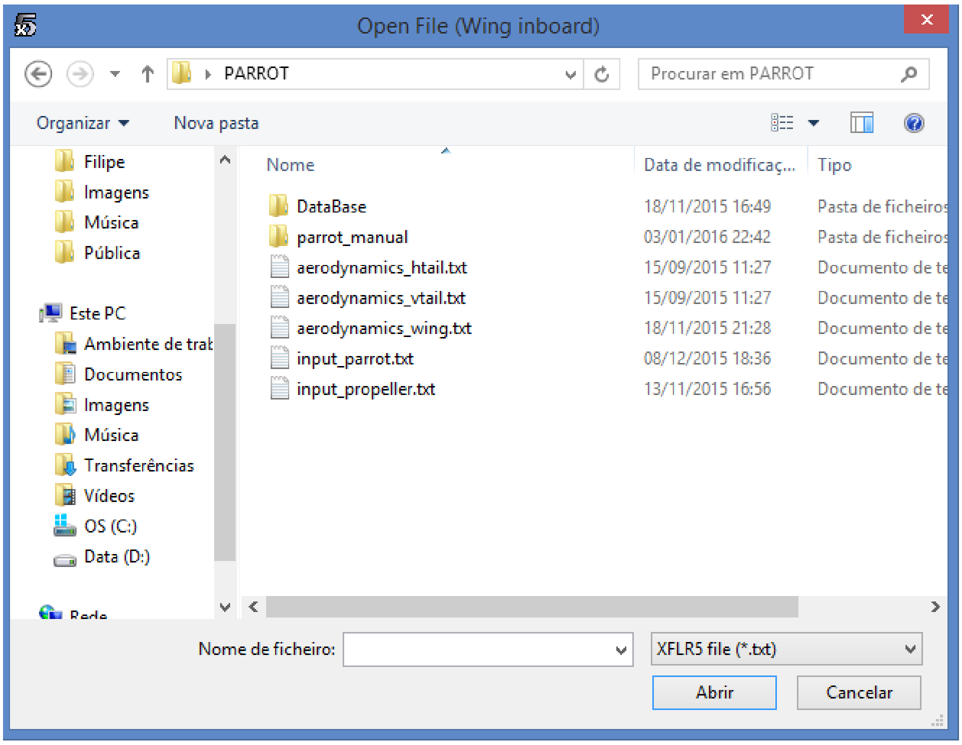


```
Wing Airfoil Name:
S1223
Reynolds:
75000.00000
Number of points:
88
    Alpha          C_l         C_d          C_m
-2.0000000000 0.3508803330 0.0605992894 -0.1158786067
-1.4000000000 0.8827197867 0.0266581501 -0.2447916965
-1.2000000000 0.9466362752 0.0252394575 -0.2535553971
-1.0000000000 0.9944100819 0.0244984222 -0.2586844248
-0.8000000000 1.0415623640 0.0240319991 -0.2633571054
-0.6000000000 1.0755732155 0.0240144159 -0.2655630799
-0.4000000000 1.1110068323 0.0247413857 -0.2673789667
-0.2000000000 1.1228456291 0.0255493083 -0.2646034758
0.0000000000 1.1393508903 0.0262011327 -0.2625680775
0.2000000000 1.1631127560 0.0266162457 -0.2626082641
0.4000000000 1.1866430587 0.0269022481 -0.2628043757
0.6000000000 1.2169582736 0.0271623495 -0.2643505853
0.8000000000 1.2454373536 0.0275154056 -0.2657118074
1.0000000000 1.2707067259 0.0278808591 -0.2665434497
1.2000000000 1.2963965147 0.0282191368 -0.2672106651
1.4000000000 1.3260811173 0.0286086205 -0.2686773488
1.6000000000 1.3488390288 0.0290952889 -0.2689288105
Reynolds:
225000.00000
Number of points:
94
    Alpha          C_l         C_d          C_m
-1.8000000000 0.8831659864 0.0187873016 -0.2499592696
-1.6000000000 0.9215307258 0.0181310188 -0.2532825221
-1.4000000000 0.9560337276 0.0177830863 -0.2556659541
-1.2000000000 0.9888795164 0.0173786480 -0.2578451154
-1.0000000000 1.0198815783 0.0173701000 -0.2596395434
-0.8000000000 1.0538475654 0.0171883083 -0.2620402253
```

Figure 2.9: Airfoils aerodynamic analysis data loading format example.

The name of the aerodynamic surface data being loaded is visible at the top of Figure 2.10.

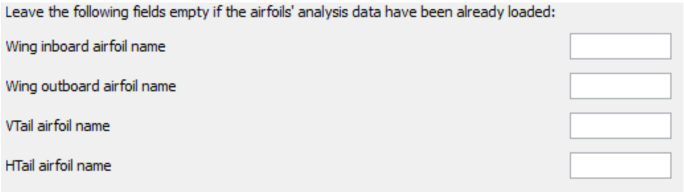
- **Option 2:** The user can load the wing, horizontal stabilizer and vertical stabilizer airfoils'



Nome	Data de modificaç...	Tipo
DataBase	18/11/2015 16:49	Pasta de ficheiros
parrot_manual	03/01/2016 22:42	Pasta de ficheiros
aerodynamics_htail.txt	15/09/2015 11:27	Documento de te
aerodynamics_vtail.txt	15/09/2015 11:27	Documento de te
aerodynamics_wing.txt	18/11/2015 21:28	Documento de te
input_parrot.txt	08/12/2015 18:36	Documento de te
input_propeller.txt	13/11/2015 16:56	Documento de te

Figure 2.10: Aerodynamic data loading data.

coordinates in the XFLR5 and perform a *Batch Analysis*, before opening the PARROT code. Once these aerodynamic analyzes have been run, when the PARROT input data is being loaded, the user must write the airfoils' names in the corresponding fields (Figure 2.11).



Leave the following fields empty if the airfoils' analysis data have been already loaded:

Wing inboard airfoil name

Wing outboard airfoil name

VTail airfoil name

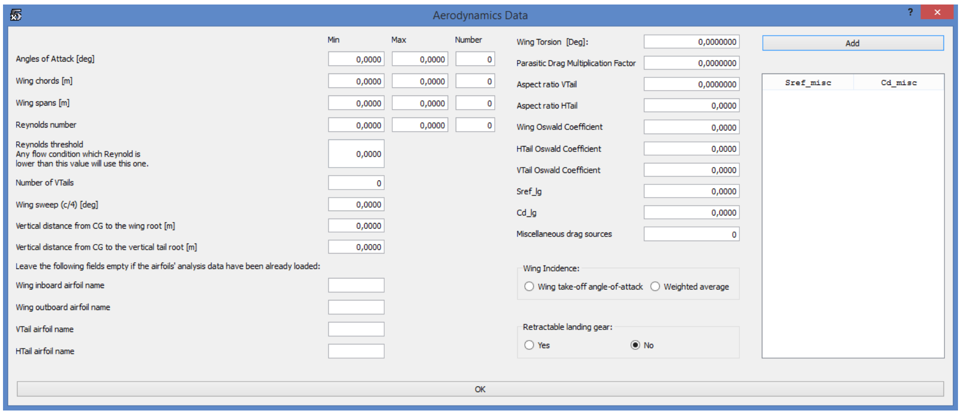
HTail airfoil name

Figure 2.11: Fields to write the name of the airfoils previously analyzed in XFLR5.

The aerodynamic input window is shown in Figure 2.12. It should be observed that the number of minimum and maximum angle-of-attack and the number of minimum and maximum Reynolds number must match the ones defined in the airfoils aerodynamic coefficients file (**Option 1**) or the ones defined for the *Batch Analysis* (**Option 2**).

After loading miscellaneous drag sources click on the *Add* button to be able to load the respective reference area and miscellaneous drag coefficients.

The aerodynamic inputs are described in Table 2.8.



	Min	Max	Number
Angles of Attack [deg]	0,0000	0,0000	0
Wing chords [m]	0,0000	0,0000	0
Wing spans [m]	0,0000	0,0000	0
Reynolds number	0,0000	0,0000	0

Reynolds threshold
Any flow condition which Reynold is
lower than this value will use this one. 0,0000

Number of VTails 0

Wing sweep (c/4) [deg] 0,0000

Vertical distance from CG to the wing root [m] 0,0000

Vertical distance from CG to the vertical tail root [m] 0,0000

Leave the following fields empty if the airfoils' analysis data have been already loaded:

Wing inboard airfoil name

Wing outboard airfoil name

VTail airfoil name

HTail airfoil name

Wing Torsion [Deg]: 0,0000000

Parasitic Drag Multiplication Factor 0,0000000

Aspect ratio VTail 0,0000000

Aspect ratio HTail 0,0000

Wing Oswald Coefficient 0,0000

HTail Oswald Coefficient 0,0000

VTail Oswald Coefficient 0,0000

Sref_lg 0,0000

Cd_lg 0,0000

Miscellaneous drag sources 0

Wing Incidence:
○ Wing take-off angle-of-attack ○ Weighted average

Retractable landing gear:
○ Yes ● No

Sref_misc	Cd_misc

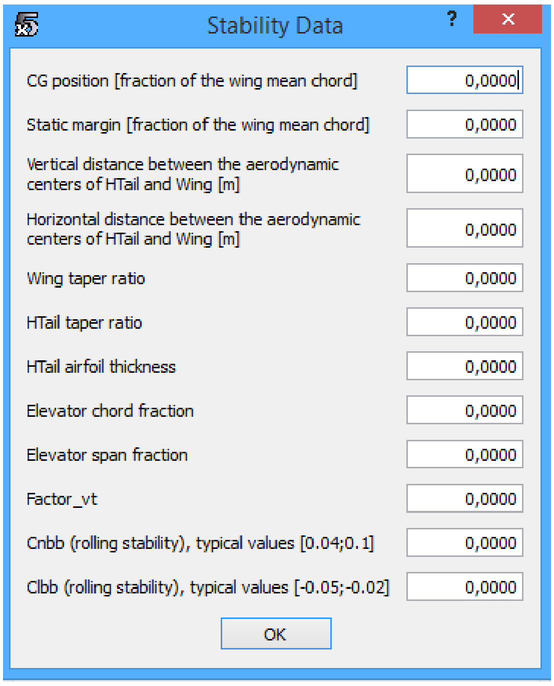
Figure 2.12: Aerodynamic Inputs.

Table 2.8: Aerodynamic inputs description.

Label	Description
Angles of attack	-
Wing chords [m]	-
Wing spans [m]	-
Reynolds number	-
Reynolds threshold	-
Number of v-tails	number of vertical tails
Wing sweep (c/4) [deg]	wing sweep at the wing's quarter chord line
Vertical distance from CG to the wing root [m]	-
Vertical distance from CG to the vertical tail root [m]	-
Parasite Drag Multiplication Factor	factor that multiplies the parasite drag
Aspect Ratio VTail	vertical stabilizer aspect ratio
Aspect Ratio HTail	horizontal stabilizer aspect ratio
Wing Oswald Coefficient	-
HTail Oswald Coefficient	horizontal stabilizer Oswald coefficient
VTail Oswald Coefficient	vertical stabilizer Oswald coefficient
Sref_lg	landing gear reference area (for miscellaneous drag calculation)
Cd_lg	landing gear drag coefficient (for miscellaneous drag calculation)
Miscellaneous drag sources	number of miscellaneous drag sources
Sref_misc	miscellaneous drag reference area (other than the landing gears)
Cd_misc	miscellaneous drag coefficient (other than the landing gears)

2.2.1.5 Stability

PARROT also requires some stability data, which can be loaded on the *Stability Data window*, shown in Figure 2.13.



CG position [fraction of the wing mean chord] 0,0000

Static margin [fraction of the wing mean chord] 0,0000

Vertical distance between the aerodynamic centers of HTail and Wing [m] 0,0000

Horizontal distance between the aerodynamic centers of HTail and Wing [m] 0,0000

Wing taper ratio 0,0000

HTail taper ratio 0,0000

HTail airfoil thickness 0,0000

Elevator chord fraction 0,0000

Elevator span fraction 0,0000

Factor_vt 0,0000

Cnbb (rolling stability), typical values [0.04;0.1] 0,0000

Clbb (rolling stability), typical values [-0.05;-0.02] 0,0000

Figure 2.13: Stability Data input window.

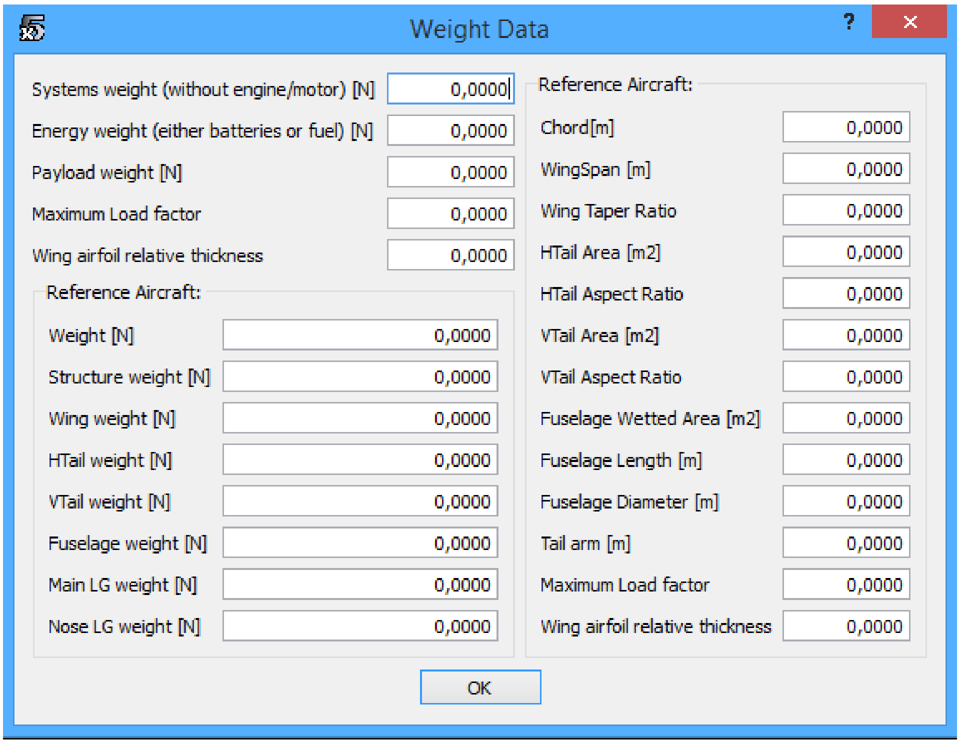
This window's inputs are explained in Table 2.9.

Table 2.9: Stability inputs description.

Label	Description
CG position [% of the wing mean chord]	-
Static margin [% of the wing mean chord]	-
Vertical distance between aerodynamic centres of HTail and Wing [m]	-
Horizontal distance between aerodynamic centres of HTail and Wing [m]	-
Wing taper ratio	-
HTail taper ratio	-
HTail airfoil thickness	-
Elevator chord fraction	-
Elevator span fraction	-
Factor$_{vt}$	-
$C_{n\beta}$ (rolling stability), typical values [0.04;0.1]	-
$C_{l\beta}$ (yawing stability), typical values [-0.05;-0.02]	-

2.2.1.6 Weight

When *Weight Data* is selected, the window shown in (Figure 2.14) is displayed. The first five inputs correspond to the actual aircraft settings. If the minimize energy mission mode is chosen, the energy weight should be left empty, whereas if the maximum payload mission mode is chosen, the payload weight should be left blank. The remaining inputs are inside two boxes called ”Reference Aircraft”. These inputs correspond to a previously built aircraft with similar materials, manufacturing techniques and mission goals, which will be used to estimate the current aircraft structural weight.



Weight Data

Systems weight (without engine/motor) [N] 0,0000
Energy weight (either batteries or fuel) [N] 0,0000
Payload weight [N] 0,0000
Maximum Load factor 0,0000
Wing airfoil relative thickness 0,0000

Reference Aircraft:
Weight [N] 0,0000
Structure weight [N] 0,0000
Wing weight [N] 0,0000
HTail weight [N] 0,0000
VTail weight [N] 0,0000
Fuselage weight [N] 0,0000
Main LG weight [N] 0,0000
Nose LG weight [N] 0,0000

Reference Aircraft:
Chord[m] 0,0000
WingSpan [m] 0,0000
Wing Taper Ratio 0,0000
HTail Area [m2] 0,0000
HTail Aspect Ratio 0,0000
VTail Area [m2] 0,0000
VTail Aspect Ratio 0,0000
Fuselage Wetted Area [m2] 0,0000
Fuselage Length [m] 0,0000
Fuselage Diameter [m] 0,0000
Tail arm [m] 0,0000
Maximum Load factor 0,0000
Wing airfoil relative thickness 0,0000

Figure 2.14: Weight Data input window.

This window’s inputs are explained in Tables 2.10 and 2.11.

Table 2.10: Actual aircraft inputs description.

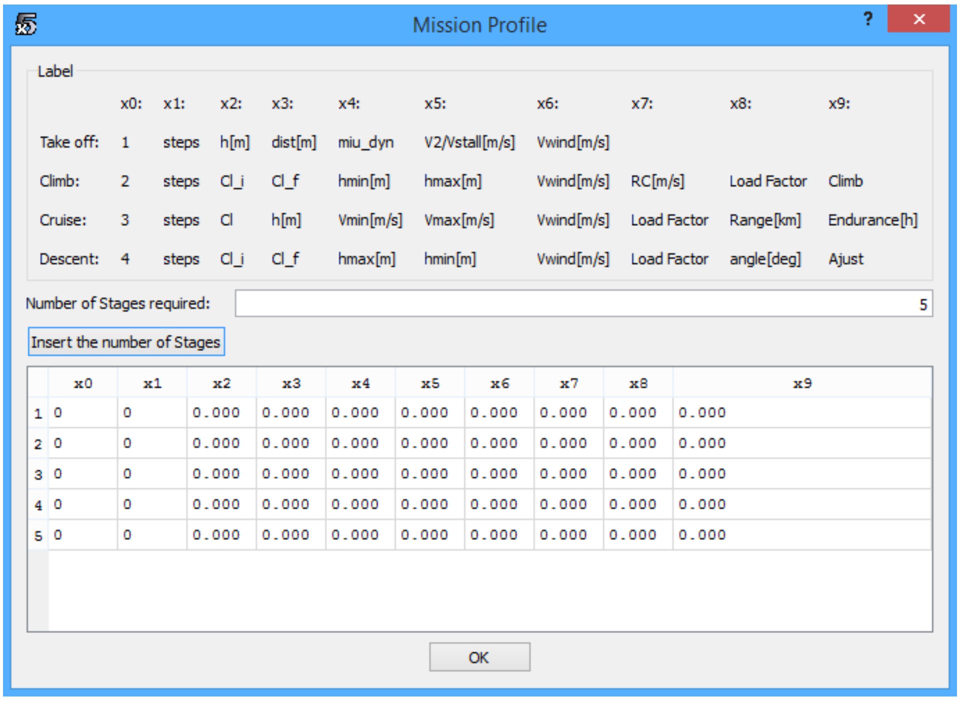
Label	Description
Systems weight (without accounting for engine or motor weight) [N]	-
Energy weight (either batteries or fuel) [N]	-
Payload weight [N]	-
Maximum load factor	maximum intended design load factor
Wing airfoil Relative thickness (t/c)	wing airfoil relative thickness (thickness/chord)

Table 2.11: Reference aircraft inputs description.

Label	Description
Weight [N]	design take-off weight
Structure weight [N]	aircraft structural weight
Wing weight [N]	-
HTail weight [N]	horizontal stabilizer weight
VTail weight [N]	vertical stabilizer weight
Fuselage weight [N]	-
Main LG weight [N]	main landing gear weight
Nose LG weight [N]	nose landing gear weight
Chord [m]	mean chord
WingSpan [m]	-
Wing Taper Ratio	-
HTail Area [m2]	horizontal stabilizer area
HTail Aspect Ratio	horizontal stabilizer aspect ratio
VTail Area [m2]	vertical stabilizer area
VTail Aspect Ratio	vertical stabilizer aspect ratio
Fuselage Wet Area [m2]	fuselage outer skin area
Fuselage Length [m]	-
Fuselage Diameter [m]	-
Tail arm [m]	-
Maximum load factor	maximum design load factor
Wing airfoil relative thickness (t/c)	wing airfoil relative thickness (thickness/chord)

2.2.1.7 Mission

After filling the aforementioned disciplinary windows and before the PARROT is able to run, the user has to load the mission profile requirements. For the PARROT code, each mission is divided in four possible mission stage categories: take-off, climb, cruise and descent. While each mission profile can only feature a single take-off, there can be an arbitrary number of climb, cruise and descent stages. The landing stage is disregarded because it is not considered relevant for the sake of minimizing energy or maximizing payload. First, it is necessary to input the number of stages required and click on the button *Insert the number of Stages*. As a consequence, a table like the one shown in Figure 2.15 is displayed. Above this table, the user can find the code for assigning the right variables in the right cells. If there happens not to be a match between a particular parameter (e.g. x7, x8 and x9 in the take-off stage), the corresponding cell should be left blank.



	x0:	x1:	x2:	x3:	x4:	x5:	x6:	x7:	x8:	x9:
Take off:	1	steps	h[m]	dist[m]	miu_dyn	V2/Vstall[m/s]	Vwind[m/s]			
Climb:	2	steps	Cl_i	Cl_f	hmin[m]	hmax[m]	Vwind[m/s]	RC[m/s]	Load Factor	Climb
Cruise:	3	steps	Cl	h[m]	Vmin[m/s]	Vmax[m/s]	Vwind[m/s]	Load Factor	Range[km]	Endurance[h]
Descent:	4	steps	Cl_i	Cl_f	hmax[m]	hmin[m]	Vwind[m/s]	Load Factor	angle[deg]	Ajust

Number of Stages required: 5

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
1	0	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 2.15: Mission profile input window.

Tables 2.12 through 2.15 feature the different mission stages inputs description.

Table 2.12: Take-off mission stage.

Label	Variable	Description
X0	1	stage reference number
X1	to_steps	number of discretization steps
X2	h [m]	take-off altitude
X3	to_dist [m]	take-off distance
X4	miu_dyn	dynamic rolling friction coefficient
X5	V2/Vstall [m/s]	ratio between the lift-off velocity and the stall velocity
X6	Vwind [m/s]	wind velocity
X7	-	-
X8	-	-
X9	-	-

Table 2.13: Climb mission stage.

Label	Variable	Description
X0	2	stage reference number
X1	cb_steps	number of discretization steps
X2	Cl_cb_i	initial wing airfoil lift coefficient
X3	Cl_cb_f	final wing airfoil lift coefficient
X4	hmin[m]	minimum or initial altitude
X5	hmax[m]	maximum or final altitude
X6	Vwind [m/s]	wing velocity
X7	RC [m/s]	rate-of-climb
X8	Load Factor	-
X9	Climb	1-thrust setting=1; 2-fixed rate-of-climb

Table 2.14: Cruise mission stage.

Label	Variable	Description
X0	3	stage reference number
X1	cz_steps	number of discretization steps
X2	Cl_cz	wing airfoil lift coefficient
X3	h [m]	altitude
X4	Vmin [m/s]	minimum velocity
X5	Vmax [m/s]	maximum velocity
X6	Vwind[m/s]	wing velocity
X7	Load Factor	-
X8	Range [km]	-
X9	Endurance [h]	-

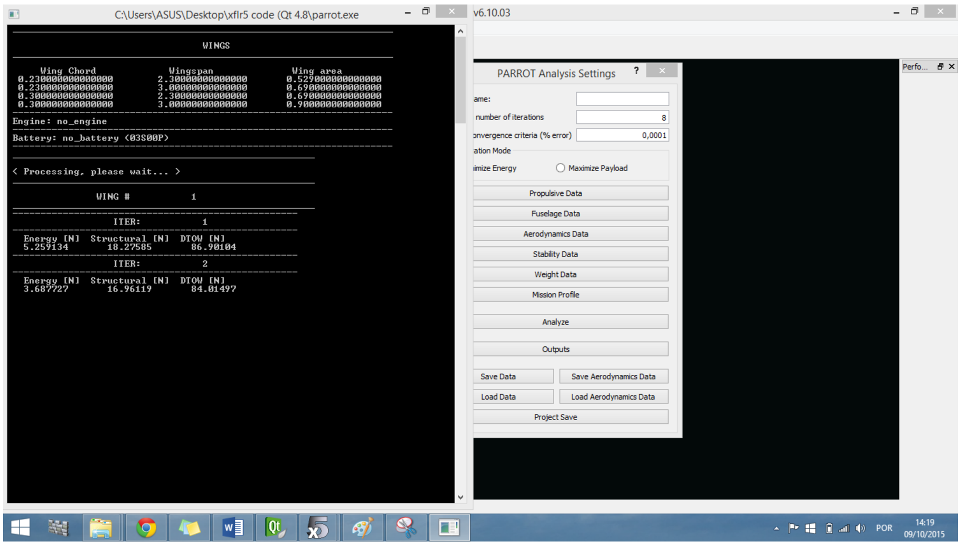
Table 2.15: Descent mission stage.

Label	Variable	Description
X0	4	stage reference number
X1	dt$_{s}teps$	number of discretization steps
X2	Cl_dt_i	initial wing airfoil lift coefficient
X3	Cl_dt_f	final wing airfoil lift coefficient
X4	hmax[m]	maximum or initial altitude
X5	hmin[m]	minimum or final altitude
X6	Vwind[m/s]	wind velocity
X7	Load Factor	-
X8	dt_angle[deg]	descent angle
X9	Ajust	1- adjust Cl_airfoil; 2- adjust angle of descent [deg]

2.2.2 Processing

Once all the inputs have been dully filled, the next step is to run the PARROT code. To do that the user should click on button *Analyze* (Figure 2.3). As the analysis starts, a black window will appear before PARROT Main Menu, which will display the different wing geometries'analysis and weight convergence process in real time (Figure 2.16).

Depending on the number of wings and analysis settings, the running of the PARROT code can take from a few minutes up to several hours. However, the user should be aware that if the process gets stopped, i.e., if the PARROT output window is not updated for a long period of time, it has probably got to an unfeasible solution. If that is the case, the user should stop the process and evaluate the potential causes of such happening.



C:\Users\ASUS\Desktop\xflr5 code (Qt 4.8\parrot.exe

```
                                WINGS

    Wing Chord                 Wingspan                 Wing area
 0.230000000000000         2.30000000000000         0.529000000000000
 0.230000000000000         3.00000000000000         0.690000000000000
 0.300000000000000         2.30000000000000         0.690000000000000
 0.300000000000000         3.00000000000000         0.900000000000000
------------------------------------------------------------------------
Engine: no_engine
------------------------------------------------------------------------
Battery: no_battery (03S00P)
------------------------------------------------------------------------

< Processing, please wait... >

              WING #          1

-----------------------------------------------------
                   ITER:            1
-----------------------------------------------------
  Energy [N]  Structural [N]  DTOW [N]
  5.259134       18.27585       86.90104
-----------------------------------------------------
                   ITER:            2
-----------------------------------------------------
  Energy [N]  Structural [N]  DTOW [N]
  3.687727       16.96119       84.01497
```

v6.10.03

PARROT Analysis Settings

ame:

number of iterations 8

onvergence criteria (% error) 0,0001

ation Mode

imize Energy ○ Maximize Payload

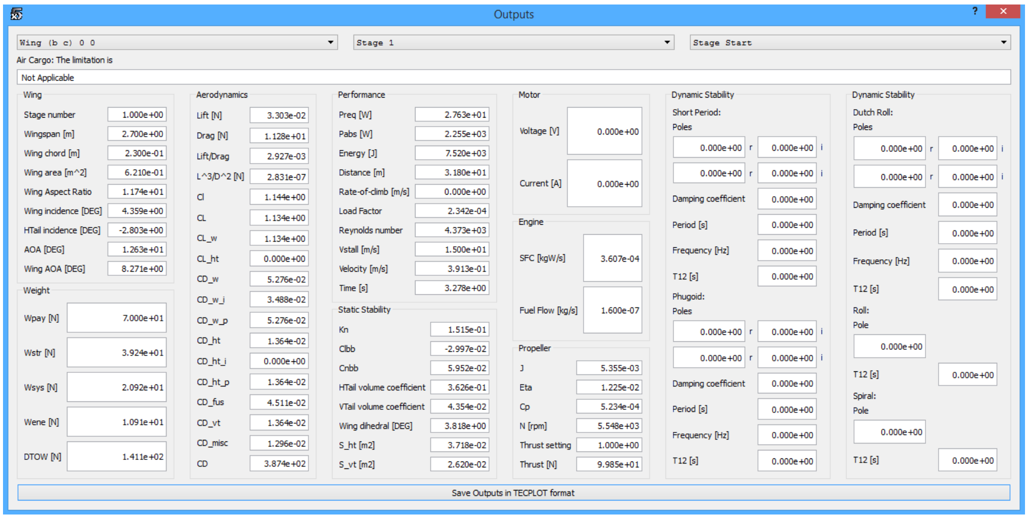
Figure 2.16: PARROT analysis running.

2.2.3 Outputs

After the analysis process is complete, the outputs are ready to be displayed. The user must click on the *Outputs* button to open the respective window (Figure 2.17).

This window has three combo boxes:

- the first one is to choose displaying the outputs of each different wing geometry (wing mean chord and wingspan combination);
- the second allows the user to choose between the different mission stages;
- the third one is to choose between displaying the output parameters at the beginning or at the end of the stage.



Wing (b c) 0 0
Stage 1
Stage Start

Air Cargo: The limitation is
Not Applicable

Wing
Stage number 1.000e+00
Wingspan [m] 2.700e+00
Wing chord [m] 2.300e-01
Wing area [m^2] 6.210e-01
Wing Aspect Ratio 1.174e+01
Wing incidence [DEG] 4.359e+00
HTail incidence [DEG] -2.803e+00
AOA [DEG] 1.263e+01
Wing AOA [DEG] 8.271e+00

Weight
Wpay [N] 7.000e+01
Wstr [N] 3.924e+01
Wsys [N] 2.092e+01
Wene [N] 1.091e+01
DTOW [N] 1.411e+02

Aerodynamics
Lift [N] 3.303e-02
Drag [N] 1.128e+01
Lift/Drag 2.927e-03
L^3/D^2 [N] 2.831e-07
Cl 1.144e+00
CL 1.134e+00
CL_w 1.134e+00
CL_ht 0.000e+00
CD_w 5.276e-02
CD_w_i 3.488e-02
CD_w_p 5.276e-02
CD_ht 1.364e-02
CD_ht_i 0.000e+00
CD_ht_p 1.364e-02
CD_fus 4.511e-02
CD_vt 1.364e-02
CD_misc 1.296e-02
CD 3.874e+02

Performance
Preq [W] 2.763e+01
Pabs [W] 2.255e+03
Energy [J] 7.520e+03
Distance [m] 3.180e+01
Rate-of-climb [m/s] 0.000e+00
Load Factor 2.342e-04
Reynolds number 4.373e+03
Vstall [m/s] 1.500e+01
Velocity [m/s] 3.913e-01
Time [s] 3.278e+00

Static Stability
Kn 1.515e-01
Clbb -2.997e-02
Cnbb 5.952e-02
HTail volume coefficient 3.626e-01
VTail volume coefficient 4.354e-02
Wing dihedral [DEG] 3.818e+00
S_ht [m2] 3.718e-02
S_vt [m2] 2.620e-02

Motor
Voltage [V] 0.000e+00
Current [A] 0.000e+00

Engine
SFC [kgW/s] 3.607e-04
Fuel Flow [kg/s] 1.600e-07

Propeller
J 5.355e-03
Eta 1.225e-02
Cp 5.234e-04
N [rpm] 5.548e+03
Thrust setting 1.000e+00
Thrust [N] 9.985e+01

Dynamic Stability
Short Period:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00
Phugoid:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00

Dynamic Stability
Dutch Roll:
Poles
0.000e+00 r 0.000e+00 i
0.000e+00 r 0.000e+00 i
Damping coefficient 0.000e+00
Period [s] 0.000e+00
Frequency [Hz] 0.000e+00
T12 [s] 0.000e+00
Roll:
Pole
0.000e+00
T12 [s] 0.000e+00
Spiral:
Pole
0.000e+00
T12 [s] 0.000e+00

Figure 2.17: Outputs window.

At the bottom of the window there is the button *Save Outputs in TECPLOT format* which generates a TECPLOT format file with all the listed output data. This is one possible way of generating parametric plots, which can make the analysis between the different wing geometries significantly more straightforward.

2.2.4 Saving and Loading Options

In the PARROT main menu (Figure 2.3) there are four buttons that have not yet been mentioned. These are:

1. Load Data - loads the input data ".txt" file;
2. Save Data - saves the previously loaded data in the user defined directory;
3. Save Aerodynamic Data - saves the airfoils'aerodynamic data into ".txt" files in the user defined directory;
4. Save Project - opens a window to choose a directory where the user wants to save all the input and output data in a folder with the respective project name.

2.3 Post-Processing

The handling of the numeric output is up to the user preferences. Since the output data is automatically written in various files, there are countless alternatives. The authors advise the users to consider using the TECPLOT or an analogous software package, once it enables a swift way of creating multidimensional plots.

Appendix E

MTOP GUI Users' Manual

MTOP

MULTILEVEL DESIGN OPTIMIZATION
(MTOP)
User's Manual

Aerospace Sciences Department
Faculty of Engineering
University of Beira Interior
PORTUGAL

Authors:
Pedro F. Albuquerque
Filipe R. Fraqueiro
Pedro V. Gamboa

Covilhã, September, 2016

ii

Contents

iv

List of Figures

List of Tables

viii

Chapter 1

Introduction

1.1 Foreword

This document is the MulTilevel aircraft design OPtimization (MTOP) graphical user interface user's manual. Its purpose is to provide an outlook on the methods used in the calculations, and to provide assistance for the less intuitive aspects of this software's handling.

The MTOP software has been developed at the Department of Aerospace Sciences of the University of Beira Interior [1] with the ultimate goal of fostering a more efficient, effective and comprehensive preliminary design optimization of unmanned aerial vehicles. As such, it is a multidisciplinary design optimization tool, which includes the mainstream disciplines of aeronautical design, with the exception of structures, though it features a structural weight estimate methodology. Its optimization methodology relies on a distributed approach, with different design disciplines/subspaces being concurrently optimized and guiding the system level optimization, which after convergence returns the design variables' global optimum.

One of the most distinguishing features of this tool is that it optimizes the design variables for the whole mission, instead of using the traditional optimization approach which is commonly focused on a single mission stage.

Using off-the-shelf gradient-based optimization algorithms, the MTOP code, developed in Fortran language, is an expedite way of optimizing the user defined design variables so as to minimize the vehicle's energy consumption complying with the predefined mission profile performance requirements. Furthermore, it enables a primary assessment of the profitability of morphing solutions - namely variable span wing and variable airfoil camber via a continuous flap - for the mission under consideration.

1.2 Methodology

Being a preliminary design optimization code - where the trade-off between accuracy and computational cost typically means that saving time and resources outweighs an acceptable accuracy penalty - low fidelity models have been used for modeling the different design disciplines.

The first question which may arise when a multilevel optimization architecture methodology is implemented lies in defining the disciplines (or subspaces) under consideration. A discipline can represent nearly any part, sub-part, partition or sub-domain of the overall system. In the case of aircraft design, it is common to consider the traditional design disciplines: aerodynamics, structures, performance, stability, controls, etc, as the optimization disciplines of the multilevel subspaces. However that is not necessarily the case.

Since one of the core goals of the current research project is the assessment of morphing technologies, resulting that the wing configuration will change according to each mission stage, the mission stages (e.g. take-off, climb, cruise, etc) are the design optimization disciplines instead of the aforementioned traditional aircraft design disciplines. To avoid any possible confusion between the concepts of discipline and subspace, which are commonly interchangeably used within the context of multilevel optimization, the definition of subspace adopted in this manual refers to the subspace optimizations (e.g. take-off, climb, cruise, etc), whereas discipline refers to the traditional disciplinary models (e.g. aerodynamics, performance, propulsion, etc.).

Amongst the multiple multilevel design optimization architectures presented in the literature over the last decades [2] the Enhanced Collaborative Optimization [3, 4] has been chosen, because it is one of the newest and most promising, yet poorly implemented alternatives. Since a thorough study of the design optimization algorithms is out of the scope of the research work which lead to the development of the current code, two of-the-shelf gradient-based optimization algorithms have been used [5–7].

An additional mention-worth aspect of the MTOP code, is that it enables the user to either locally or globally optimize each of the design variables, with respect to each subspace/mission stage, which is particularly useful in the case of morphing solutions. The optimization variables are the wingspan (b), the wing mean chord ($\bar{c}$), the propeller diameter (d), the propeller pitch (p), the flap relative chord (c_{flap}) and the flap deflection (d_{flap}). While the wing chord, the propeller diameter and the flap relative chord are global variables with respect to the each mission stage, the wingspan, the propeller pitch and the flap deflection can assume different values at each mission stage, if one considers a variable span wing, a variable pitch propeller and a variable deflection flap, resulting in a wing airfoil camber difference, respectively.

1.3 Structure

The programming language used was Fortran 90 using Microsoft Visual Studio 2010. Within the program's working directory there is a folder called «input_data» where all the input data files that the user should feed or update prior to running the MTOP can be found. Likewise, all the output files are created in a folder called «output_data», also within the working directory.

Chapter 2

MTOP Graphical User Interface

2.1 GUI Implementation

The development of MTOP's graphical user interface (GUI) was made using the open source XFLR5 GUI, which is programmed in C++ language. The main reasons for using the XFLR5 framework were the fact that it is an open source code, easy to handle and already having expedite methods for the aerodynamic analysis of airfoils (using XFOIL).

2.2 GUI Handling

After opening the standard XFLR5 File Menu, the options shown in (Figure 2.1) are displayed. Within the red rectangle, it is possible to see a new sub-menu called *Aircraft Optimization*, which shall be selected. Alternatively, the user can choose the keyboard shortcut (Ctrl+P).

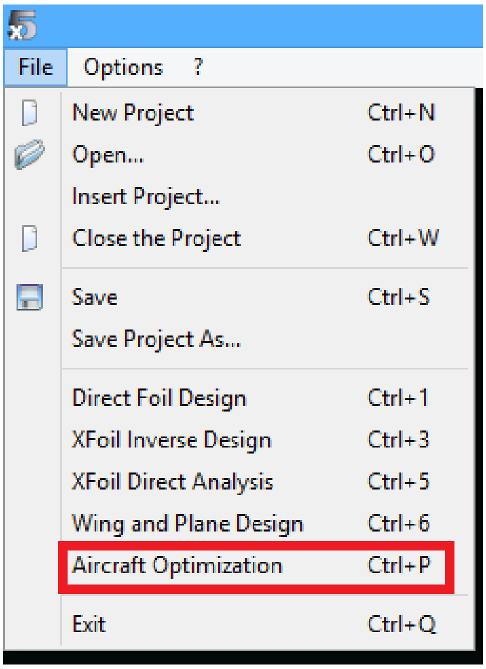

File
Options
?
New Project Ctrl+N
Open... Ctrl+O
Insert Project...
Close the Project Ctrl+W
Save Ctrl+S
Save Project As...
Direct Foil Design Ctrl+1
XFoil Inverse Design Ctrl+3
XFoil Direct Analysis Ctrl+5
Wing and Plane Design Ctrl+6
Aircraft Optimization Ctrl+P
Exit Ctrl+Q

Figure 2.1: XFLR5 Sub-Menus.

Once the Aircraft Optimization option has been chosen, it is possible to visualize two new options: Analysis and Flap. If the Analysis separator is chosen, two aircraft optimization codes are available: PARROT and MTOP (Figure 2.24). To get to the MTOP code main menu, the user shall choose the MTOP option. Alternatively, the user can choose the keyboard shortcut (Ctrl+M).

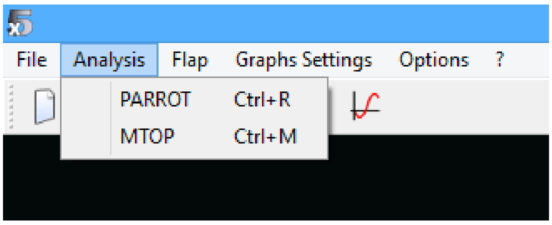
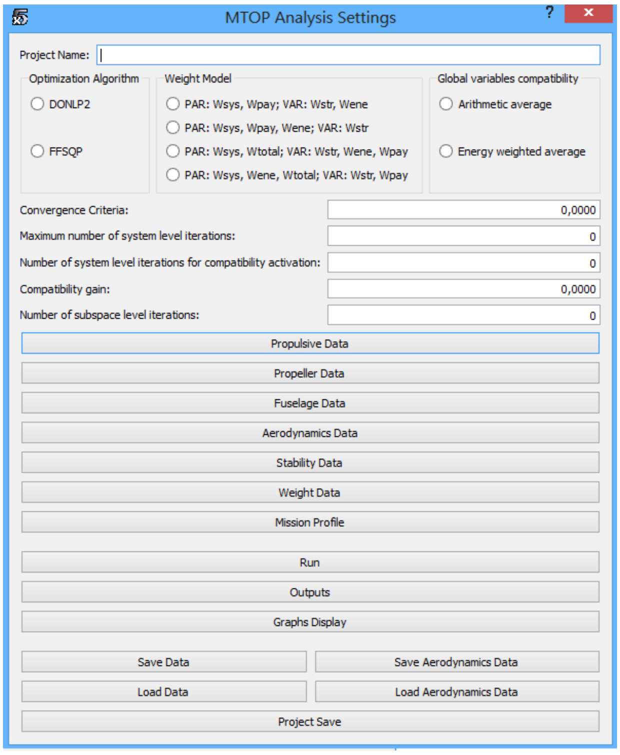


Figure 2.2: Analysis Types.

At this point, the user can have an overview of MTOP's main menu (Figure 2.3).



Project Name:

Convergence Criteria: 0,0000

Maximum number of system level iterations: 0

Number of system level iterations for compatibility activation: 0

Compatibility gain: 0,0000

Number of subspace level iterations: 0

Figure 2.3: MTOP Main Menu.

At the top of this main menu window (Figure 2.3), the user can load the general inputs listed in Table 2.1, presented below.

Table 2.1: General Inputs.

Label	Description
Project Name	name of the project
Maximum number of iterations	maximum number of acceptable iterations of the vehicle's design take-off weight
Weight convergence criteria (% error)	weight convergence criteria as a percentage of the design take-off weight
Mission Type	mission optimization mode, minimize energy or a maximize payload

It is also possible to visualize the various input categories required for the MTOP analysis to be run:

- Propulsive Data;
- Propeller Data;
- Fuselage Data;
- Aerodynamics Data;
- Stability Data;
- Weight Data;
- Mission Profile.

Furthermore, it is possible to visualize the *Analyze* button, which can trigger the MTOP optimization as soon as all the input data is loaded. Below this option, the *Outputs* button can be found. Its selection makes it possible to visualize the output data. Lastly, it is possible to visualize the different options which either enable data storage in ".txt" files or input data load using ".txt" files:

- Save Data;
- Save Aerodynamics Data;
- Load Data;
- Load Aerodynamics Data;
- Save Project.

The forthcoming sections describe the different disciplinary input data windows as well as the mission profile window.

2.2.1 Inputs

2.2.1.1 Propulsion

After clicking on *Propulsive Data* button, a new window will open. Firstly, the user has to decide between two types of propulsion: Electric Motor (Figure 2.4) or Combustion Engine (Figure 2.5). There are some input parameters that are independent of the propulsion type. These are described in Tables 2.2 and 2.3.

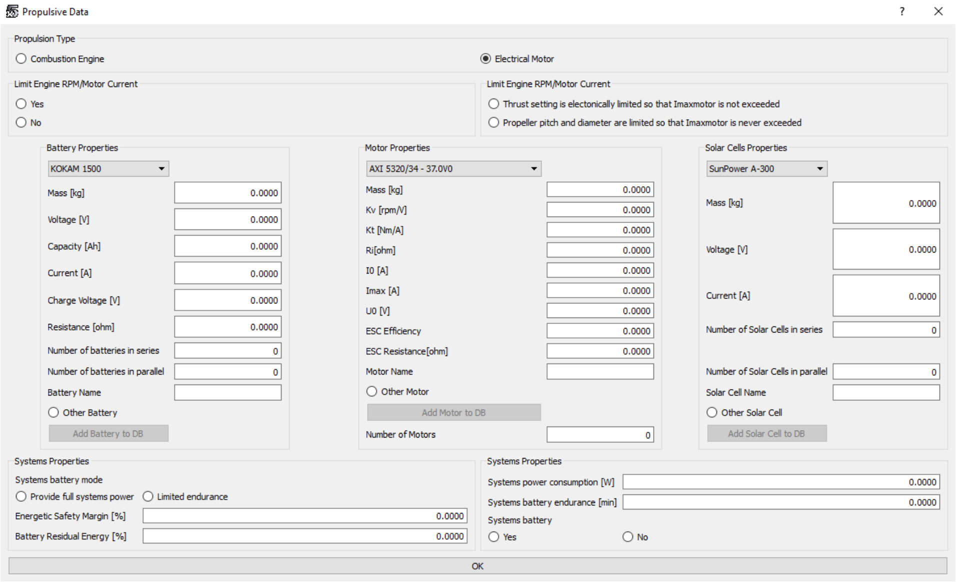
Table 2.2: Propulsion Limitations.

LIMIT ENGINE RPM /MOTOR CURRENT	
YES	NO

Table 2.3: Systems properties inputs description.

Label	Description
Systems battery mode	provide full systems power / limited endurance
Energetic Safety Margin [%]	-
Battery Residual Energy [%]	-
Systems battery	Yes / No
Systems power consumption [W]	-
Systems battery endurance [min]	-

If the user chooses the Electric Motor, the requested input data includes the Battery, Motor and Solar Cells properties (Figure 2.4). These properties are explained in Table 2.4.



Propulsive Data

Propulsion Type

○ Combustion Engine ● Electrical Motor

Limit Engine RPM/Motor Current

○ Yes

○ No

Limit Engine RPM/Motor Current

○ Thrust setting is electonically limited so that Imaxmotor is not exceeded

○ Propeller pitch and diameter are limited so that Imaxmotor is never exceeded

Battery Properties

KOKAM 1500

Mass [kg]	0.0000
Voltage [V]	0.0000
Capacity [Ah]	0.0000
Current [A]	0.0000
Charge Voltage [V]	0.0000
Resistance [ohm]	0.0000
Number of batteries in series	0
Number of batteries in parallel	0
Battery Name	

○ Other Battery

Motor Properties

AXI 5320/34 - 37.0V0

Mass [kg]	0.0000
Kv [rpm/V]	0.0000
Kt [Nm/A]	0.0000
Ri[ohm]	0.0000
I0 [A]	0.0000
Imax [A]	0.0000
U0 [V]	0.0000
ESC Efficiency	0.0000
ESC Resistance[ohm]	0.0000
Motor Name	

○ Other Motor

Number of Motors 0

Solar Cells Properties

SunPower A-300

Mass [kg]	0.0000
Voltage [V]	0.0000
Current [A]	0.0000
Number of Solar Cells in series	0
Number of Solar Cells in parallel	0
Solar Cell Name	

○ Other Solar Cell

Systems Properties

Systems battery mode

○ Provide full systems power ○ Limited endurance

Energetic Safety Margin [%]	0.0000
Battery Residual Energy [%]	0.0000

Systems Properties

Systems power consumption [W]	0.0000
Systems battery endurance [min]	0.0000

Systems battery

○ Yes ○ No

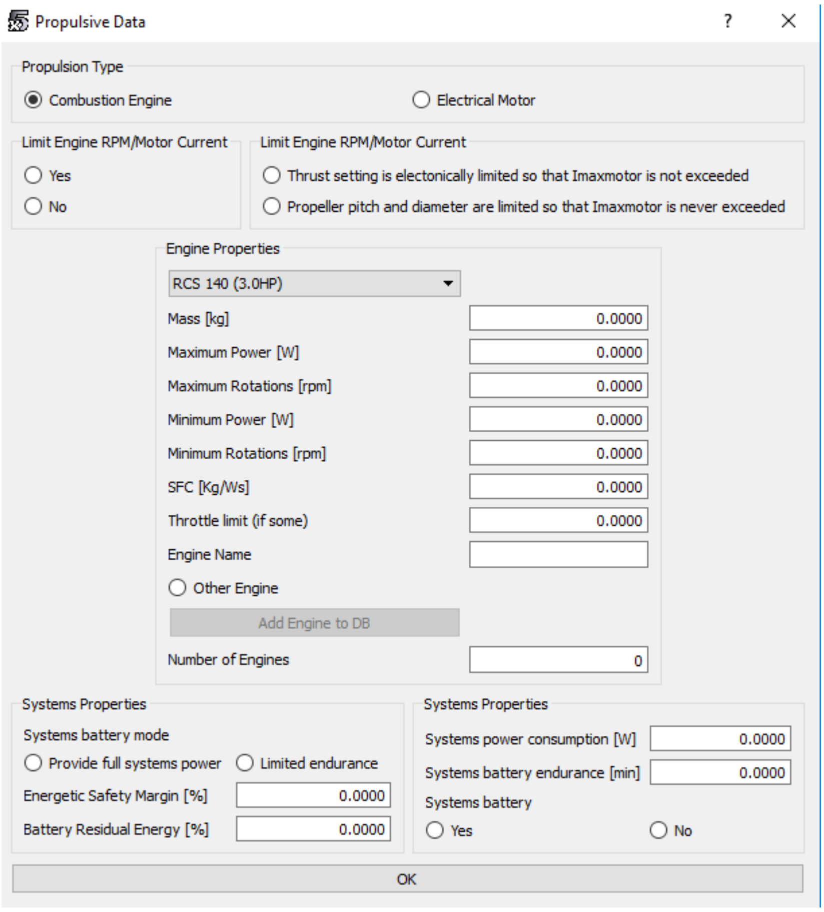
Figure 2.4: Electric Motor Inputs.

Table 2.4: Electric Motor: Battery, Motor and Solar Cell inputs description.

Label	Description
Battery	
Mass [kg]	mass
Voltage [V]	voltage
Capacity [Ah]	capacity
Current [A]	electric current
Charge Voltage [V]	charge voltage
Number of batteries in series	-
Number of batteries in parallel	-
Battery Name	-
Motor	
Mass [kg]	mass
Kv [rpm/V]	speed constant
Kt [Nm/A]	torque constant
Ri [ohm]	internal resistance
I0 [A]	no load electric current
Imax [A]	maximum electric current
U0 [V]	no load voltage
ESC efficiency	electronic speed controller efficiency
ESC resistance [ohm]	electronic speed controller resistance
Motor Name	-
Number of Motors	-
Solar Cells	
Mass [kg]	-
Voltage [V]	-
Current [A]	electric nominal current
Number of solar cells in series	-
Number of solar cells in parallel	-
Solar Cell Name	-

Conversely, if the aircraft has a combustion engine the user's task becomes limited to loading the engine data. The respective input window is featured in Figure 2.5, whereas the respective variables are described in Table 2.5.

In this last window it is possible to select the engine/motor, the battery and the solar cell directly from a database. Alternatively, if the user intends to use a different engine/motor, battery and/or solar cell, the respective toggle button *Other Engine/Motor/Battery/Solar Cell* of Figures 2.4 and 2.5 shall be used and the corresponding data shall be directly loaded on the GUI window. If the user intends to add his/her own values to the database, after loading the data, he/she should click on the button Other Engine/Motor/Battery/Solar Cell to DB (Figures 2.4 and 2.5).



Propulsive Data

Propulsion Type

● Combustion Engine ○ Electrical Motor

Limit Engine RPM/Motor Current

○ Yes

○ No

Limit Engine RPM/Motor Current

○ Thrust setting is electonically limited so that Imaxmotor is not exceeded

○ Propeller pitch and diameter are limited so that Imaxmotor is never exceeded

Engine Properties

RCS 140 (3.0HP) ▾

Mass [kg]	0.0000
Maximum Power [W]	0.0000
Maximum Rotations [rpm]	0.0000
Minimum Power [W]	0.0000
Minimum Rotations [rpm]	0.0000
SFC [Kg/Ws]	0.0000
Throttle limit (if some)	0.0000
Engine Name	

○ Other Engine

Add Engine to DB

Number of Engines 0

Systems Properties

Systems battery mode

○ Provide full systems power ○ Limited endurance

Energetic Safety Margin [%]	0.0000
Battery Residual Energy [%]	0.0000

Systems Properties

Systems power consumption [W]	0.0000
Systems battery endurance [min]	0.0000

Systems battery

○ Yes ○ No

Figure 2.5: Combustion Engine Inputs.

Table 2.5: Engine inputs description.

Label	Description
Mass [kg]	engine mass
Maximum Power [W]	-
Maximum Rotations [rpm]	-
Minimum Power [W]	-
Minimum Rotations [rpm]	-
Specific Fuel Consumption (SFC) [kg/Ws]	-
Throttle limit (if some)	-
Engine Name	-
Number of Engines	-

2.2.1.2 Propeller

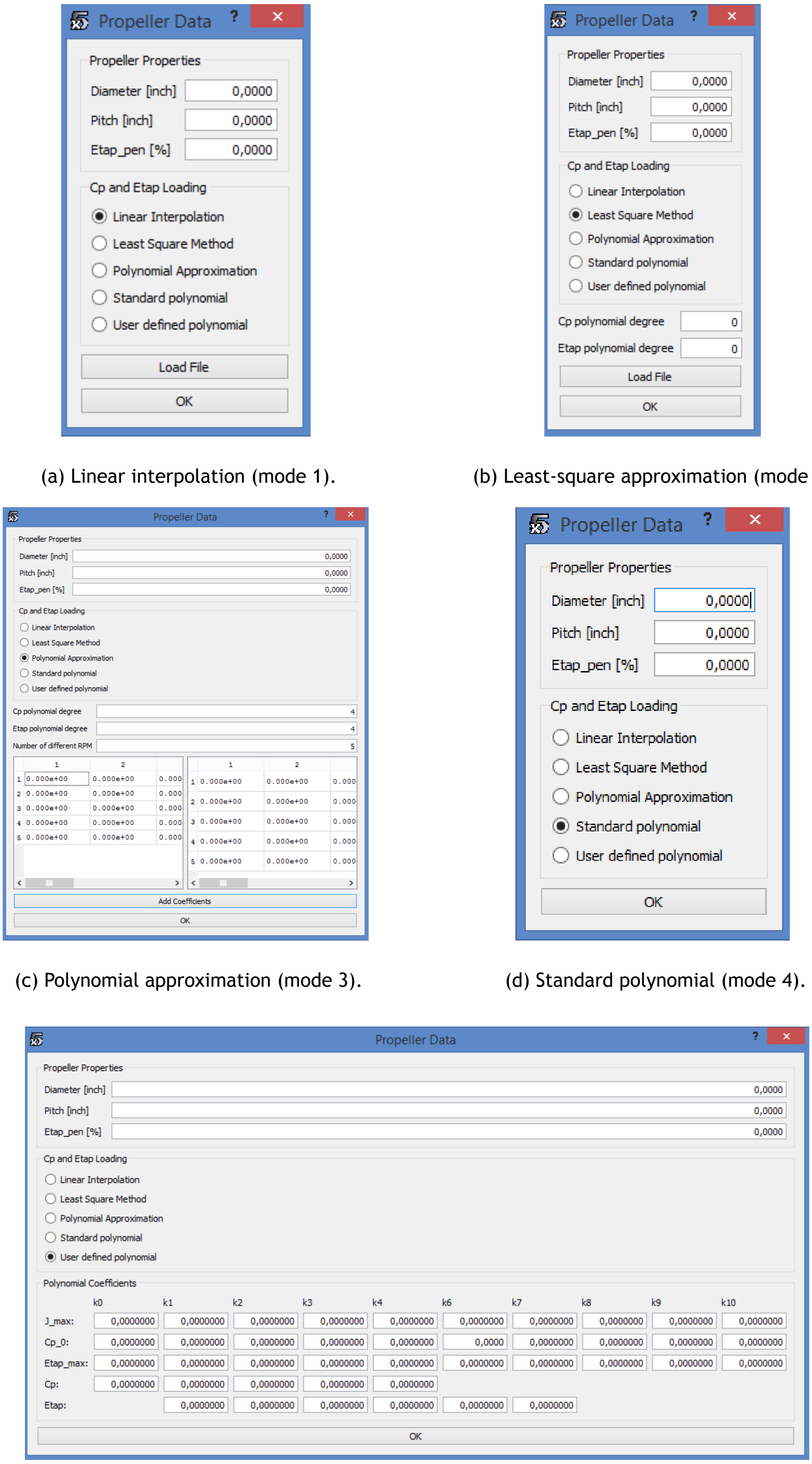
Besides the propulsion data, which includes all the relevant information regarding the engine(s) or motor(s) in use, propeller performance data is needed. On a fixed pitch propeller, this will typically be dependent on three variables, the propeller diameter, the propeller pitch and the propeller advance ratio. These performance analysis typically makes use of two different curves, the power coefficient versus advance ratio and the propeller efficiency versus the advance ratio. Accordingly, the MTOP code enables the user to choose between five different ways of loading and handling such data.

For single propeller, when the propeller diameter and pitch are constant parameters:

- **Mode 1 - Linear Interpolation (Figure 2.6a):** the user loads the ”input_propeller.txt” and the different data sets are consecutively linearly interpolated accordingly;
- **Mode 2 - Least-square method (Figure 2.6b):** the user chooses the polynomial approximation degree of the power coefficient and propeller efficiency curves and loads the ”input_propeller.txt”. The data sets will thereafter be used for generating their respective polynomial approximation using the least-square method;
- **Mode 3 - Polynomial approximation (Figure 2.6c):** the user chooses the polynomial approximation degree of the power coefficient and propeller efficiency curves and loads the polynomial coefficients directly. The user can choose to load an arbitrary number of curves for each of these two variables, each for a different rotational velocity, provided he/she loads an equal number of curves for both variables.

For generic propeller, when the propeller diameter and pitch are optimization variables:

- **Mode 4 - Standard Polynomial (Figure 2.6d):** the user chooses to use a built-in standard polynomials which provide the propeller performance as a function of the propeller diameter and propeller pitch which uses numerical simulation data;
- **Mode 5 - User Defined Polynomial (Figure 2.6e):** the user chooses to use his/her own polynomials which provide the propeller performance as a function of the propeller diameter and propeller pitch which can be fed from enhanced numerical simulation or experimental data.



Propeller Data

Propeller Properties

Diameter [inch] 0,0000
Pitch [inch] 0,0000
Etap_pen [%] 0,0000

Cp and Etap Loading

● Linear Interpolation
○ Least Square Method
○ Polynomial Approximation
○ Standard polynomial
○ User defined polynomial

(a) Linear interpolation (mode 1).

Propeller Data

Propeller Properties

Diameter [inch] 0,0000
Pitch [inch] 0,0000
Etap_pen [%] 0,0000

Cp and Etap Loading

○ Linear Interpolation
● Least Square Method
○ Polynomial Approximation
○ Standard polynomial
○ User defined polynomial

Cp polynomial degree 0
Etap polynomial degree 0

(b) Least-square approximation (mode 2).

Propeller Data

Propeller Properties

Diameter [inch] 0,0000
Pitch [inch] 0,0000
Etap_pen [%] 0,0000

Cp and Etap Loading

○ Linear Interpolation
○ Least Square Method
● Polynomial Approximation
○ Standard polynomial
○ User defined polynomial

Cp polynomial degree 4
Etap polynomial degree 4
Number of different RPM 5

	1	2	
1	0.000e+00	0.000e+00	0.000
2	0.000e+00	0.000e+00	0.000
3	0.000e+00	0.000e+00	0.000
4	0.000e+00	0.000e+00	0.000
5	0.000e+00	0.000e+00	0.000

	1	2	
1	0.000e+00	0.000e+00	0.000
2	0.000e+00	0.000e+00	0.000
3	0.000e+00	0.000e+00	0.000
4	0.000e+00	0.000e+00	0.000
5	0.000e+00	0.000e+00	0.000

(c) Polynomial approximation (mode 3).

Propeller Data

Propeller Properties

Diameter [inch] 0,0000
Pitch [inch] 0,0000
Etap_pen [%] 0,0000

Cp and Etap Loading

○ Linear Interpolation
○ Least Square Method
○ Polynomial Approximation
● Standard polynomial
○ User defined polynomial

(d) Standard polynomial (mode 4).

Propeller Data

Propeller Properties

Diameter [inch] 0,0000
Pitch [inch] 0,0000
Etap_pen [%] 0,0000

Cp and Etap Loading

○ Linear Interpolation
○ Least Square Method
○ Polynomial Approximation
○ Standard polynomial
● User defined polynomial

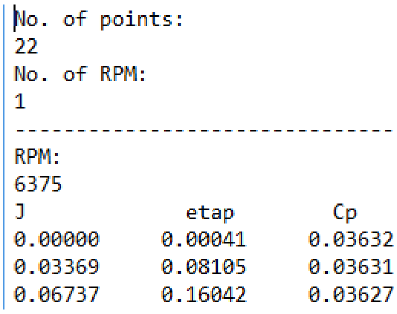
Polynomial Coefficients

	k0	k1	k2	k3	k4	k6	k7	k8	k9	k10
J_max:	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000
Cp_0:	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000	0,0000000	0,0000000	0,0000000	0,0000000
Etap_max:	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000
Cp:	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000					
Etap:		0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000			

(e) User defined polynomial (mode 5).

Figure 2.6: Propeller input modes.

If mode 1 or mode 2 are selected an “input_propeller.txt” file must be loaded. This file must have the format shown in Figure 2.7, where the number of combinations of the advance ratio (J), the propeller efficiency (η_p) and the propeller power coefficient (C_p) must be equal to the number of points. Furthermore, if the user has (n) data sets, each corresponding to a different RPM, the number of RPM must equal (n) and the respective data must follow the first one, repeating the file format after the horizontal dashed line, inclusively, (n) times.



```
No. of points:
22
No. of RPM:
1
--------------------------------
RPM:
6375
J             etap        Cp
0.00000     0.00041     0.03632
0.03369     0.08105     0.03631
0.06737     0.16042     0.03627
```

Figure 2.7: Mode 1 and 2 propeller performance loading format.

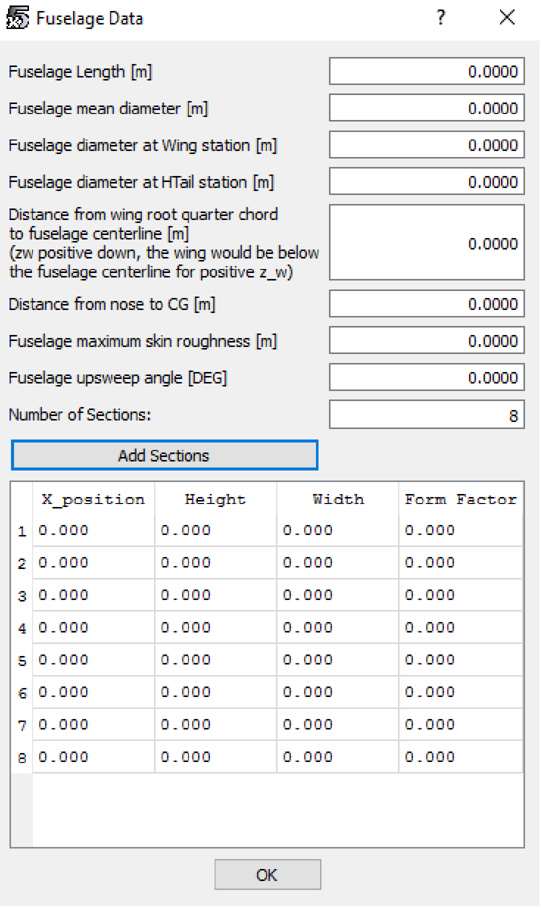
The remaining data is the same regardless of the data handling mode in use and is described in Table 2.6.

Table 2.6: Propeller inputs description.

Label	Description
Diameter [inch]	propeller diameter
Pitch [inch]	propeller pitch
Etap_pen [%]	efficiency penalty parameter for propeller propulsive efficiency
Cp and Eta$_pLoading$	data loading and handling mode

2.2.1.3 Fuselage

The Fuselage Data window is presented in Figure 2.8.



Fuselage Data

Fuselage Length [m]	0.0000
Fuselage mean diameter [m]	0.0000
Fuselage diameter at Wing station [m]	0.0000
Fuselage diameter at HTail station [m]	0.0000
Distance from wing root quarter chord to fuselage centerline [m] (zw positive down, the wing would be below the fuselage centerline for positive z_w)	0.0000
Distance from nose to CG [m]	0.0000
Fuselage maximum skin roughness [m]	0.0000
Fuselage upsweep angle [DEG]	0.0000
Number of Sections:	8

	X_position	Height	Width	Form Factor
1	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000

Figure 2.8: Fuselage inputs.

The input parameters are described in Table 2.7. After writing the number of fuselage sections on the respective tab, it is required to click on *Insert Sections* so that a table featuring the lengthwise x-position (starting at the nose and increasing rearwards), width, height and form factor for each section is displayed.

Table 2.7: Fuselage inputs description.

Label	Description
Fuselage length [m]	-
Fuselage mean diameter [m]	-
Fuselage diameter at Wing station [m]	fuselage diameter at the wing's root
Fuselage diameter at HTail station [m]	fuselage diameter at the horizontal stabilizer's root
Vertical distance from wing root quarter chord to fuselage centerline [m]	assumed positive downwards, (the wing would be below the fuselage centerline for positive z_w)
Distance from nose to CG [m]	-
Fuselage maximum skin roughness [m]	-
Fuselage upsweep angle [deg]	-
Number of sections	number of fuselage lengthwise sections
X-position	lengthwise section position measured from the fuselage nose backwards
Width	fuselage width at each section
Height	fuselage height at each section
Form Factor	section form factor - equals unity for a rectangular cross section or a lower value for rounded corners

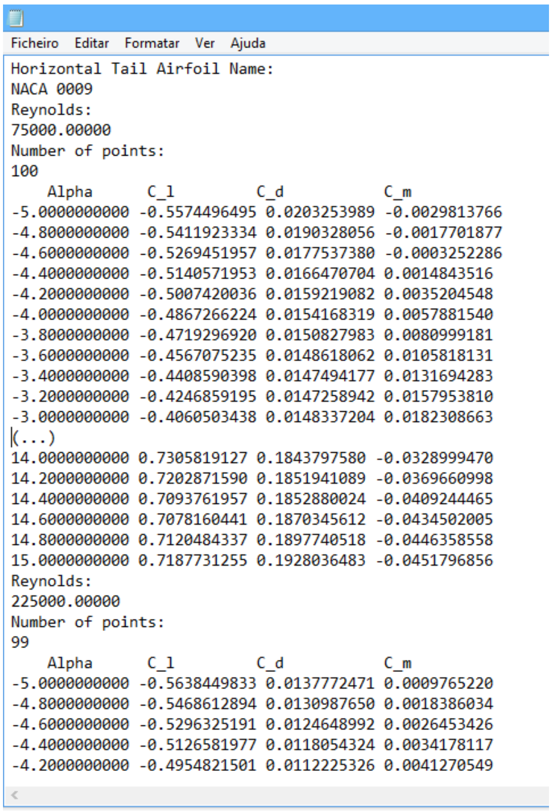
2.2.1.4 Aerodynamics

As far as the *Aerodynamics Data* window is concerned, the MTOP code requires the lifting surfaces (wing, horizontal stabilizer and vertical stabilizer) aerodynamic coefficients to be known for a set of different Reynolds numbers and angles-of-attack so that these can be accurately estimated at any flow condition. These aerodynamic coefficients are the airfoils':

- Lift coefficient (C_l);
- Drag coefficient (C_d);
- Pitching moment coefficient (C_m).

There are two ways of feeding the MTOP code with these coefficients:

- **Option 1:** The user clicks on *Load Aerodynamic Data* and the GUI will consecutively ask the user to select a ".txt" file with the airfoil's aerodynamic data for the wing, the horizontal stabilizer and the vertical stabilizer, respectively. Each ".txt" file must have the format shown in Figure 2.9.

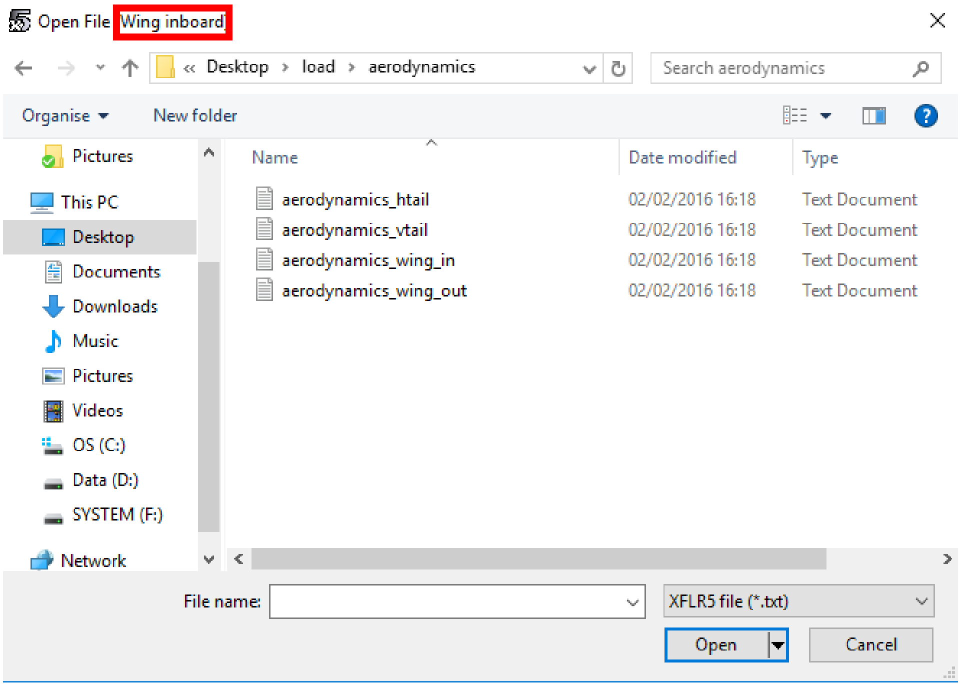


```
Horizontal Tail Airfoil Name:
NACA 0009
Reynolds:
75000.00000
Number of points:
100
    Alpha       C_l          C_d            C_m
-5.0000000000 -0.5574496495 0.0203253989 -0.0029813766
-4.8000000000 -0.5411923334 0.0190328056 -0.0017701877
-4.6000000000 -0.5269451957 0.0177537380 -0.0003252286
-4.4000000000 -0.5140571953 0.0166470704 0.0014843516
-4.2000000000 -0.5007420036 0.0159219082 0.0035204548
-4.0000000000 -0.4867266224 0.0154168319 0.0057881540
-3.8000000000 -0.4719296920 0.0150827983 0.0080999181
-3.6000000000 -0.4567075235 0.0148618062 0.0105818131
-3.4000000000 -0.4408590398 0.0147494177 0.0131694283
-3.2000000000 -0.4246859195 0.0147258942 0.0157953810
-3.0000000000 -0.4060503438 0.0148337204 0.0182308663
(...)
14.0000000000 0.7305819127 0.1843797580 -0.0328999470
14.2000000000 0.7202871590 0.1851941089 -0.0369660998
14.4000000000 0.7093761957 0.1852880024 -0.0409244465
14.6000000000 0.7078160441 0.1870345612 -0.0434502005
14.8000000000 0.7120484337 0.1897740518 -0.0446358558
15.0000000000 0.7187731255 0.1928036483 -0.0451796856
Reynolds:
225000.00000
Number of points:
99
    Alpha       C_l          C_d            C_m
-5.0000000000 -0.5638449833 0.0137772471 0.0009765220
-4.8000000000 -0.5468612894 0.0130987650 0.0018386034
-4.6000000000 -0.5296325191 0.0124648992 0.0026453426
-4.4000000000 -0.5126581977 0.0118054324 0.0034178117
-4.2000000000 -0.4954821501 0.0112225326 0.0041270549
```

Figure 2.9: Airfoils aerodynamic analysis data loading format example.

The name of the aerodynamic surface data being loaded is visible at the top of Figure 2.10.

- **Option 2:** The user can load the wing, horizontal stabilizer and vertical stabilizer airfoils' coordinates in the XFLR5 and perform a *Batch Analysis*, before opening the MTOP



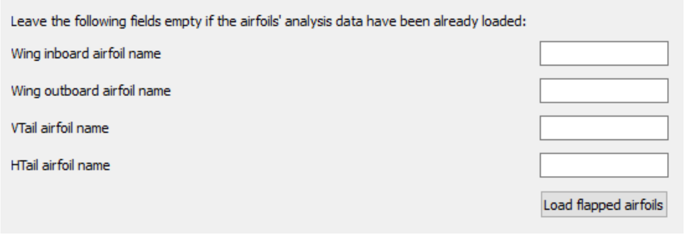
Name	Date modified	Type
aerodynamics_htail	02/02/2016 16:18	Text Document
aerodynamics_vtail	02/02/2016 16:18	Text Document
aerodynamics_wing_in	02/02/2016 16:18	Text Document
aerodynamics_wing_out	02/02/2016 16:18	Text Document

File name:

XFLR5 file (*.txt)

Figure 2.10: Aerodynamic data loading data.

code. Once these aerodynamic analyzes have been run, when the MTOP input data is being loaded, the user must write the airfoils' names in the corresponding fields (Figure 2.11).


Leave the following fields empty if the airfoils' analysis data have been already loaded:

Wing inboard airfoil name

Wing outboard airfoil name

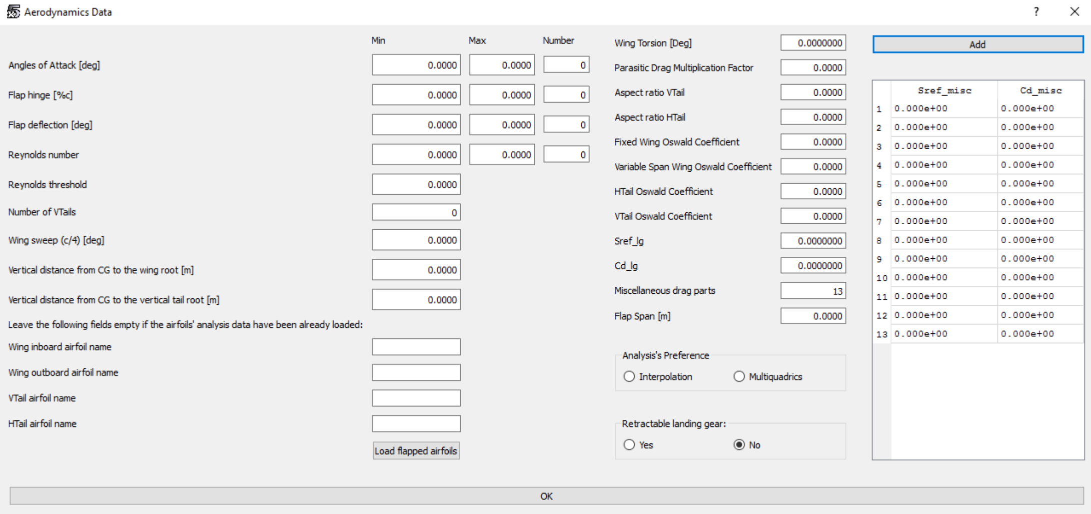
VTail airfoil name

HTail airfoil name

Figure 2.11: Fields to write the name of the airfoils previously analyzed in XFLR5.

The aerodynamic input window is shown in Figure 2.12. It should be observed that the number of minimum and maximum angle-of-attack and the number of minimum and maximum Reynolds number must match the ones defined in the airfoils aerodynamic coefficients file (**Option 1**) or the ones defined for the *Batch Analysis* (**Option 2**).

After loading the miscellaneous drag sources click on the *Add* button to be able to load the respective reference area and miscellaneous drag coefficients.



Aerodynamics Data

	Min	Max	Number
Angles of Attack [deg]	0.0000	0.0000	0
Flap hinge [%c]	0.0000	0.0000	0
Flap deflection [deg]	0.0000	0.0000	0
Reynolds number	0.0000	0.0000	0
Reynolds threshold	0.0000		
Number of VTails	0		
Wing sweep (c/4) [deg]	0.0000		
Vertical distance from CG to the wing root [m]	0.0000		
Vertical distance from CG to the vertical tail root [m]	0.0000		

Leave the following fields empty if the airfoils' analysis data have been already loaded:

Wing inboard airfoil name

Wing outboard airfoil name

VTail airfoil name

HTail airfoil name

Load flapped airfoils

Wing Torsion [Deg]	0.0000000
Parasitic Drag Multiplication Factor	0.0000
Aspect ratio VTail	0.0000
Aspect ratio HTail	0.0000
Fixed Wing Oswald Coefficient	0.0000
Variable Span Wing Oswald Coefficient	0.0000
HTail Oswald Coefficient	0.0000
VTail Oswald Coefficient	0.0000
Sref_lg	0.0000000
Cd_lg	0.0000000
Miscellaneous drag parts	13
Flap Span [m]	0.0000

Analysis's Preference

○ Interpolation ○ Multiquadrics

Retractable landing gear:

○ Yes ● No

Add

	Sref_misc	Cd_misc
1	0.000e+00	0.000e+00
2	0.000e+00	0.000e+00
3	0.000e+00	0.000e+00
4	0.000e+00	0.000e+00
5	0.000e+00	0.000e+00
6	0.000e+00	0.000e+00
7	0.000e+00	0.000e+00
8	0.000e+00	0.000e+00
9	0.000e+00	0.000e+00
10	0.000e+00	0.000e+00
11	0.000e+00	0.000e+00
12	0.000e+00	0.000e+00
13	0.000e+00	0.000e+00

OK

Figure 2.12: Aerodynamic Inputs.

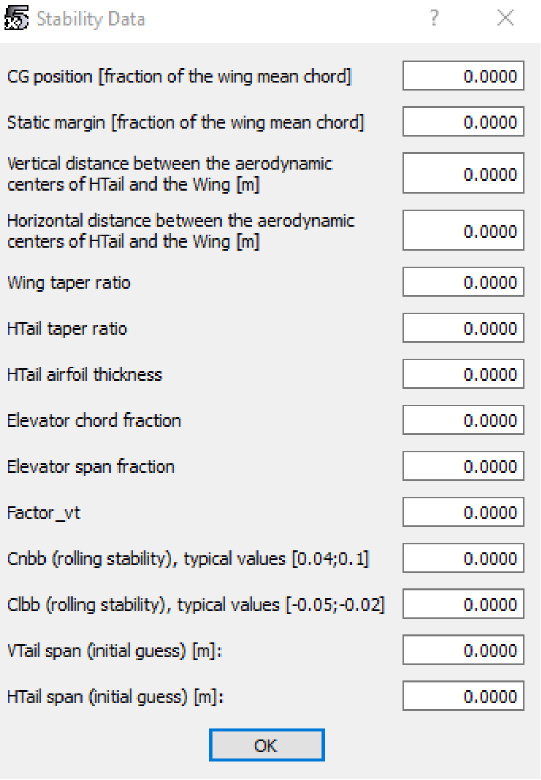
The aerodynamic inputs are described in Table 2.8.

Table 2.8: Aerodynamic inputs description.

Label	Description
Angles of attack	-
Flap hinge [%c]	-
Flap deflection [deg]	-
Reynolds number	-
Reynolds threshold	-
Number of VTails	number of vertical tails
Wing sweep (c/4) [deg]	wing sweep at the wing's quarter chord line
Vertical distance from CG to the wing root [m]	-
Vertical distance from CG to the vertical tail root [m]	-
Wing Torsion [deg]	-
Parasite Drag Multiplication Factor	factor that multiplies the parasite drag
Aspect Ratio VTail	vertical stabilizer aspect ratio
Aspect Ratio HTail	horizontal stabilizer aspect ratio
Fixed Wing Oswald Coefficient	-
Variable Span Wing Oswald Coefficient	-
HTail Oswald Coefficient	horizontal stabilizer Oswald coefficient
VTail Oswald Coefficient	vertical stabilizer Oswald coefficient
Sref_lg	landing gear reference area (for miscellaneous drag calculation)
Cd_lg	landing gear drag coefficient (for miscellaneous drag calculation)
Miscellaneous drag parts	number of miscellaneous drag sources
Flap Span [m]	-
Analysis's Preference	Interpolation or Multiquadrics
Retractable landing gear	Yes or No
Sref_misc	miscellaneous drag reference area (other than the landing gears)
Cd_misc	miscellaneous drag coefficient (other than the landing gears)

2.2.1.5 Stability

MTOP also requires some stability data, which can be loaded on the *Stability Data window*, shown in Figure 2.13.



Stability Data

CG position [fraction of the wing mean chord]	0.0000
Static margin [fraction of the wing mean chord]	0.0000
Vertical distance between the aerodynamic centers of HTail and the Wing [m]	0.0000
Horizontal distance between the aerodynamic centers of HTail and the Wing [m]	0.0000
Wing taper ratio	0.0000
HTail taper ratio	0.0000
HTail airfoil thickness	0.0000
Elevator chord fraction	0.0000
Elevator span fraction	0.0000
Factor_vt	0.0000
Cnbb (rolling stability), typical values [0.04;0.1]	0.0000
Clbb (rolling stability), typical values [-0.05;-0.02]	0.0000
VTail span (initial guess) [m]:	0.0000
HTail span (initial guess) [m]:	0.0000

Figure 2.13: Stability Data input window.

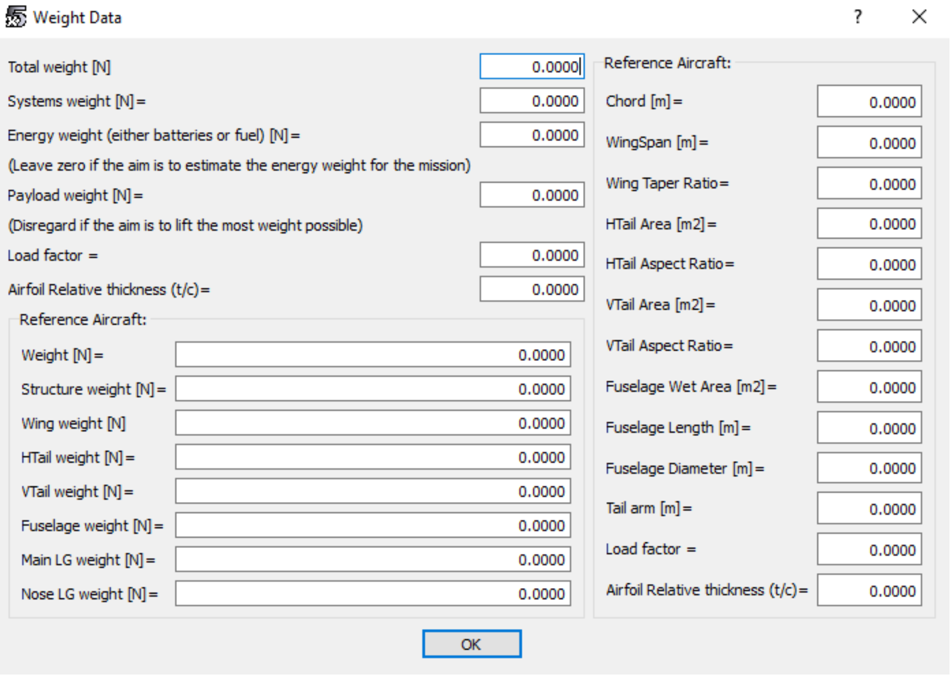
This window's inputs are explained in Table 2.9.

Table 2.9: Stability inputs description.

Label	Description
CG position [fraction of the wing mean chord]	-
Static margin [fraction of the wing mean chord]	-
Vertical distance between aerodynamic centers of HTail and Wing [m]	-
Horizontal distance between aerodynamic centers of HTail and Wing [m]	-
Wing taper ratio	-
HTail taper ratio	-
HTail airfoil thickness	-
Elevator chord fraction	-
Elevator span fraction	-
$Factor_{vt}$	-
$C_{n\beta}$ (rolling stability), typical values [0.04;0.1]	-
$C_{l\beta}$ (yawing stability), typical values [-0.05;-0.02]	-
VTail span (initial guess) [m]	-
HTail span (initial guess) [m]	-

2.2.1.6 Weight

When Weight Data is selected, the window shown in (Figure 2.14) is displayed. The first five inputs correspond to the actual aircraft settings. If the minimize energy mission mode is chosen, the Energy Weight should be left empty, whereas if the maximum payload mission mode is chosen, the Payload Weight should be left blank. The remaining inputs are inside two boxes called “Reference Aircraft”. These inputs correspond to a previously built aircraft with similar materials, manufacturing techniques and mission goals, which will be used to estimate the current aircraft structural weight.



Weight Data

Total weight [N] 0.0000
Systems weight [N]= 0.0000
Energy weight (either batteries or fuel) [N]= 0.0000
(Leave zero if the aim is to estimate the energy weight for the mission)
Payload weight [N]= 0.0000
(Disregard if the aim is to lift the most weight possible)
Load factor = 0.0000
Airfoil Relative thickness (t/c)= 0.0000

Reference Aircraft:
Weight [N]= 0.0000
Structure weight [N]= 0.0000
Wing weight [N] 0.0000
HTail weight [N]= 0.0000
VTail weight [N]= 0.0000
Fuselage weight [N]= 0.0000
Main LG weight [N]= 0.0000
Nose LG weight [N]= 0.0000

Reference Aircraft:
Chord [m]= 0.0000
WingSpan [m]= 0.0000
Wing Taper Ratio= 0.0000
HTail Area [m2]= 0.0000
HTail Aspect Ratio= 0.0000
VTail Area [m2]= 0.0000
VTail Aspect Ratio= 0.0000
Fuselage Wet Area [m2]= 0.0000
Fuselage Length [m]= 0.0000
Fuselage Diameter [m]= 0.0000
Tail arm [m]= 0.0000
Load factor = 0.0000
Airfoil Relative thickness (t/c)= 0.0000

Figure 2.14: Weight Data input window.

This window’s inputs are explained in Tables 2.10 and 2.11.

Table 2.10: Actual aircraft inputs description.

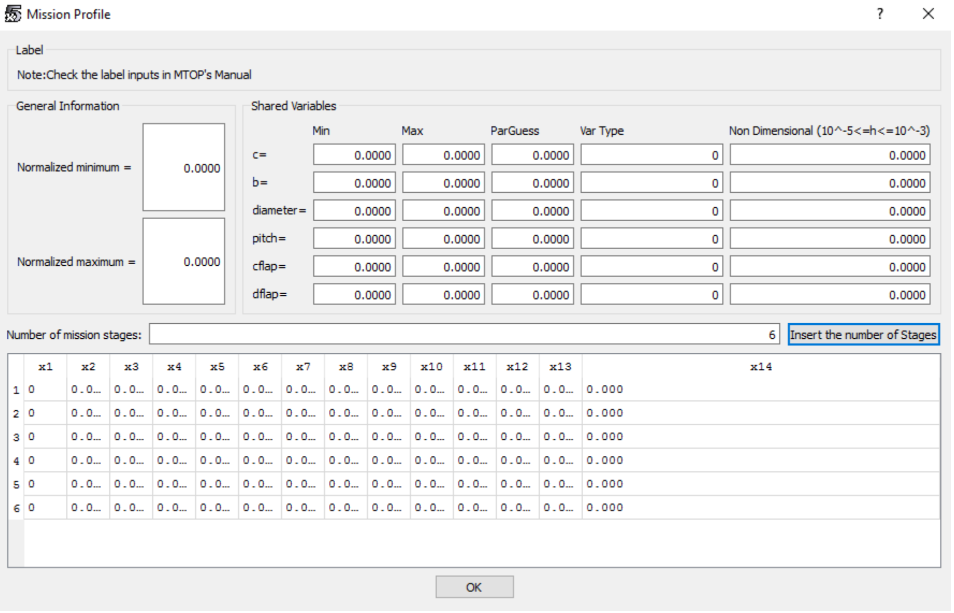
Label	Description
Total weight [N]	-
Systems weight (without accounting for engine or motor weight) [N]	-
Energy weight (either batteries or fuel) [N]	-
Payload weight [N]	-
Load factor	-
Wing airfoil Relative thickness (t/c)	wing airfoil relative thickness (thickness/chord)

Table 2.11: Reference aircraft inputs description.

Label	Description
Weight [N]	design take-off weight
Structure weight [N]	aircraft structural weight
Wing weight [N]	-
HTail weight [N]	horizontal stabilizer weight
VTail weight [N]	vertical stabilizer weight
Fuselage weight [N]	-
Main LG weight [N]	main landing gear weight
Nose LG weight [N]	nose landing gear weight
Chord [m]	mean chord
WingSpan [m]	-
Wing Taper Ratio	-
HTail Area [m2]	horizontal stabilizer area
HTail Aspect Ratio	horizontal stabilizer aspect ratio
VTail Area [m2]	vertical stabilizer area
VTail Aspect Ratio	vertical stabilizer aspect ratio
Fuselage Wet Area [m2]	fuselage outer skin area
Fuselage Length [m]	-
Fuselage Diameter [m]	-
Tail arm [m]	-
Load factor	-
Wing airfoil Relative thickness (t/c)	wing airfoil relative thickness (thickness/chord)

2.2.1.7 Mission

After filling the aforementioned input disciplinary windows and before the MTOP is able to run, the user has to load the mission profile requirements. For the MTOP code, each mission is divided in four possible mission stage categories: Take-off, Climb, Cruise and Descent. While each mission profile can only feature a single Take-off, there can be an arbitrary number of Climb, Cruise and Descent stages. The landing stage is disregarded because it is not considered relevant for the sake of minimizing energy or maximizing payload. First, it is necessary to input the number of stages required and after click on the button *Insert the number of Stages*. As a consequence, a table like the one shown in Figure 2.15 is displayed. Above this table, the user can find the code for assigning the right variables in the right cells. If there happens not to be a match between a particular parameter (e.g. x7, x8 and x9 in the Take-off stage), the corresponding cell should be left blank.



Label

Note:Check the label inputs in MTOP's Manual

General Information

Normalized minimum = 0.0000

Normalized maximum = 0.0000

Shared Variables

	Min	Max	ParGuess	Var Type	Non Dimensional (10^-5<=h<=10^-3)
c=	0.0000	0.0000	0.0000	0	0.0000
b=	0.0000	0.0000	0.0000	0	0.0000
diameter=	0.0000	0.0000	0.0000	0	0.0000
pitch=	0.0000	0.0000	0.0000	0	0.0000
cflap=	0.0000	0.0000	0.0000	0	0.0000
dflap=	0.0000	0.0000	0.0000	0	0.0000

Number of mission stages: 6

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14
1	0	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.000
2	0	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.000
3	0	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.000
4	0	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.000
5	0	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.000
6	0	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.0…	0.000

Figure 2.15: Mission profile input window.

Tables 2.13 through 2.16 feature the different mission stages inputs description.

Table 2.12: General Info and Shared Variables

Label	Description
Normalized minimum	-
Normalized maximum	-
c, b, diameter, pitch, cflap, dflap	Min, Max, ParGuess, Var Type, Non Dimensional

Table 2.13: Take-off mission stage.

Label	Variable	Description
X0	1	stage reference number
X1	to_steps	number of discretization steps
X2	h [m]	take-off altitude
X3	to_dist [m]	take-off distance
X4	miu_dyn	dynamic rolling friction coefficient
X5	V2/Vstall [m/s]	ratio between the lift-off velocity and the stall velocity
X6	Vwind [m/s]	wind velocity

Table 2.14: Climb (2) or (3) mission stage.

Label	Variable	Description
X0	2 or 3	stage reference number
X1	cb_steps	number of discretization steps
X2	Cl_cb_i	initial wing airfoil lift coefficient
X3	Cl_cb_f	final wing airfoil lift coefficient
X4	hmin[m]	minimum or initial altitude
X5	hmax[m]	maximum or final altitude
X6	Vwind[m/s]	wind velocity
X7	RC[m/s]	rate-of-climb
X8	Load Factor	-
X9	Climb	1-thrust setting=1; 2-fixed rate-of-climb

Table 2.15: Cruise (4) or (5) mission stage.

Label	Variable	Description
X0	4 or 5	stage reference number
X1	cz_steps	number of discretization steps
X2	Cl_cz	wing airfoil lift coefficient
X3	h [m]	altitude
X4	Vmin [m/s]	minimum velocity
X5	Vmax [m/s]	maximum velocity
X6	Vwind[m/s]	wind velocity
X7	Load Factor	-
X8	Range[km]	-
X9	Endurance[h]	-

Table 2.16: Descent mission stage.

Label	Variable	Description
X0	6	stage reference number
X1	dt_steps	number of discretization steps
X2	Cl_dt_i	initial wing airfoil lift coefficient
X3	Cl_dt_f	final wing airfoil lift coefficient
X4	hmax[m]	maximum or initial altitude
X5	hmin[m]	minimum or final altitude
X6	Vwind[m/s]	wind velocity
X7	Load Factor	-
X8	dt_angle[deg]	descent angle
X9	Ajust	1- adjust Cl_airfoil; 2- adjust angle of descent [deg]

2.2.2 Flapped airfoil generator

The flapped airfoil generator is not intrinsic to the MTOP code, but can feed the MTOP code whenever a variable camber morphing technique aims to be studied. In addition, the flapped airfoil generator can be used in isolation to determine the new set of airfoil coordinates of a given airfoil after it has been cambered, with the only required inputs being the flap chord and the flap deflection.

To access the flapped airfoil generator, window, the user should observe the following approach:

1. Open XFLR5;
2. Go to: "File" and then select "Aircraft Optimization" (or click in Ctrl+P), Figure 2.16;

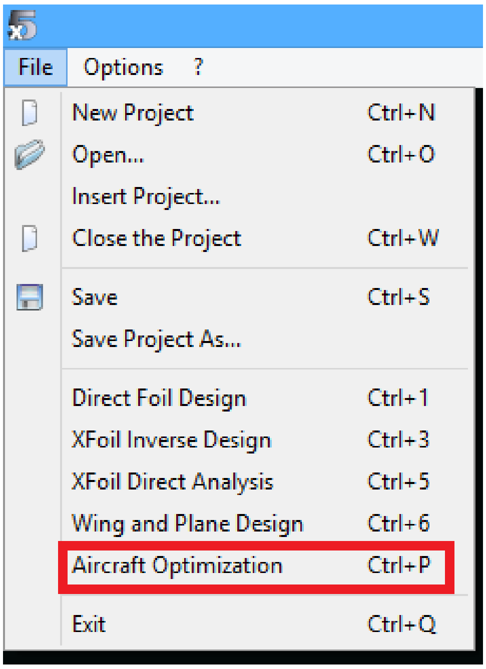

File
Options
?
New Project Ctrl+N
Open... Ctrl+O
Insert Project...
Close the Project Ctrl+W
Save Ctrl+S
Save Project As...
Direct Foil Design Ctrl+1
XFoil Inverse Design Ctrl+3
XFoil Direct Analysis Ctrl+5
Wing and Plane Design Ctrl+6
Aircraft Optimization Ctrl+P
Exit Ctrl+Q

Figure 2.16: XFLR5 Sub-menus.

3. Select "Flap" in the toolbar and then "Flap" (or click in Ctrl+F), Figure 2.17;

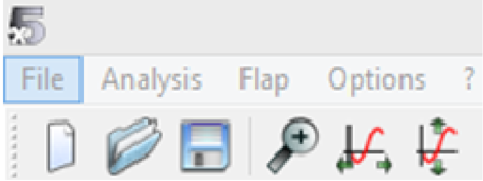


Figure 2.17: XFLR5 Toolbar.

4. Enter the inputs required to perform the analysis, Figure 2.18:

   choose amongst the three available methodologies to deflect the airfoil (their description can be found on the window's help);

number of different flap chords to be generated;

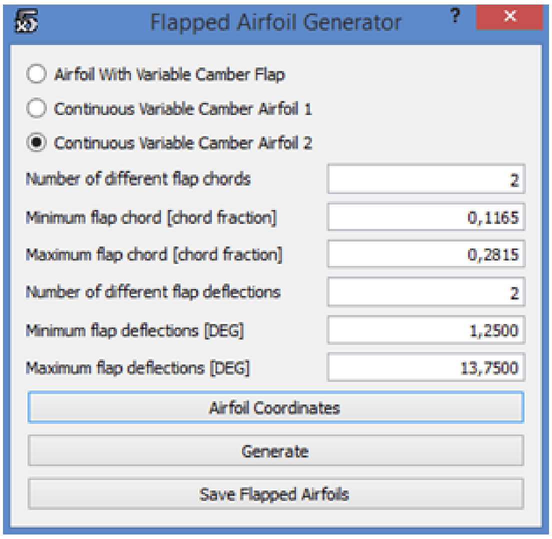
minimum flap chord (as a fraction of the airfoil chord);

maximum flap chord (as a fraction of the airfoil chord);

number of different flap deflections to be generated;

minimum flap deflection (in degrees);

maximum flap deflection (in degrees).



○ Airfoil With Variable Camber Flap

○ Continuous Variable Camber Airfoil 1

● Continuous Variable Camber Airfoil 2

Number of different flap chords 2

Minimum flap chord [chord fraction] 0,1165

Maximum flap chord [chord fraction] 0,2815

Number of different flap deflections 2

Minimum flap deflections [DEG] 1,2500

Maximum flap deflections [DEG] 13,7500

Figure 2.18: Flap menu.

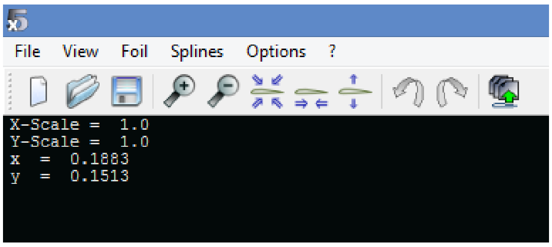
5. Select “Airfoil Coordinates” to select the “.dat” file with the airfoil coordinates for which the analysis is intended;

6. Click on “Generate” to perform the analysis;

7. Click on “Save Flapped Airfoils” and then select the folder to save the Airfoils generated.

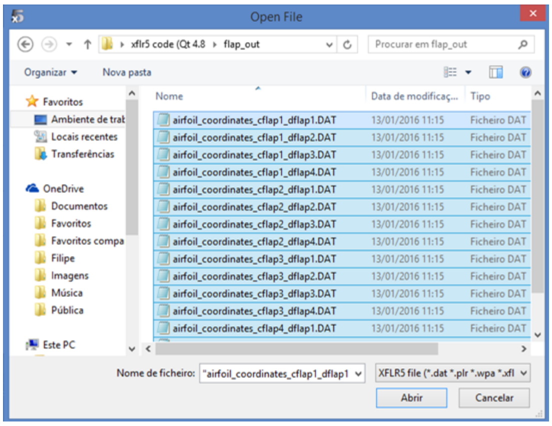
Once the flapped airfoils coordinates have been generated, if the user wants to perform an aerodynamic analysis in XFLR5, he/she should observe the following steps:

1. Go to: “File” and then select “Direct Foil Design” (or enter Ctrl+1). Select the folder’s icon to select the “.dat” files previously generated, Figure 2.19 and , Figure 2.20.



```
X-Scale =   1.0
Y-Scale =   1.0
x  =  0.1883
y  =  0.1513
```

Figure 2.19: Airfoils' import menu.



Nome	Data de modificaç...	Tipo
airfoil_coordinates_cflap1_dflap1.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap1_dflap2.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap1_dflap3.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap1_dflap4.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap2_dflap1.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap2_dflap2.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap2_dflap3.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap2_dflap4.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap3_dflap1.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap3_dflap2.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap3_dflap3.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap3_dflap4.DAT	13/01/2016 11:15	Ficheiro DAT
airfoil_coordinates_cflap4_dflap1.DAT	13/01/2016 11:15	Ficheiro DAT

Nome de ficheiro: "airfoil_coordinates_cflap1_dflap1

XFLR5 file (*.dat *.plr *.wpa *.xfl

Figure 2.20: Selection of airfoils.

2. Once all the airfoil coordinates files have been selected, the aerodynamic analysis can take place. Accordingly, the user can go to "XFoil Direct Analysis" (or enter Ctrl+5) and after choose the "Batch Analysis" (or enter Shift+F6) option on the "Analysis" toolbar, Figure 2.21.

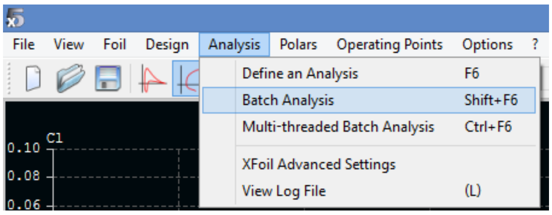
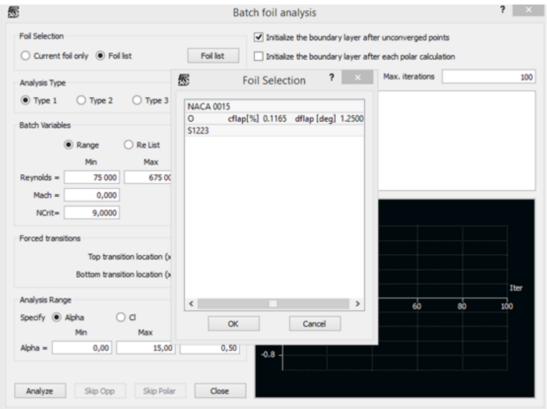


Figure 2.21: Analysis selection.

3. In the "batch foil analysis window", select the radio button "Foil List" and then click on "Foil List" to select the desired foils in the list and enter the Batch Analysis parameters, Figure 2.22.

4. Then just select all the the aerodynamics analysis files, Figure 2.23.


Batch foil analysis

Foil Selection: ○ Current foil only ● Foil list

☑ Initialize the boundary layer after unconverged points

☐ Initialize the boundary layer after each polar calculation

Max. iterations 100

Analysis Type: ● Type 1 ○ Type 2 ○ Type 3

Batch Variables: ● Range ○ Re List

	Min	Max
Reynolds =	75 000	675 00
Mach =	0,000	
NCrit=	9,0000	

Forced transitions

Top transition location (x

Bottom transition location (x

Analysis Range

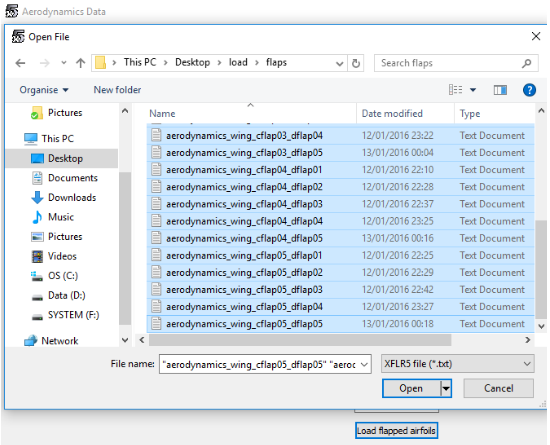
Specify ● Alpha ○ Cl

	Min	Max	
Alpha =	0,00	15,00	0,50

Foil Selection

NACA 0015
O cflap[%] 0.1165 dflap [deg] 1.2500
S1223

Figure 2.22: Batch analysis inputs window.


Aerodynamics Data

Open File

This PC › Desktop › load › flaps

Name	Date modified	Type
aerodynamics_wing_cflap03_dflap04	12/01/2016 23:22	Text Document
aerodynamics_wing_cflap03_dflap05	13/01/2016 00:04	Text Document
aerodynamics_wing_cflap04_dflap01	12/01/2016 22:10	Text Document
aerodynamics_wing_cflap04_dflap02	12/01/2016 22:28	Text Document
aerodynamics_wing_cflap04_dflap03	12/01/2016 22:37	Text Document
aerodynamics_wing_cflap04_dflap04	12/01/2016 23:25	Text Document
aerodynamics_wing_cflap04_dflap05	13/01/2016 00:16	Text Document
aerodynamics_wing_cflap05_dflap01	12/01/2016 22:25	Text Document
aerodynamics_wing_cflap05_dflap02	12/01/2016 22:29	Text Document
aerodynamics_wing_cflap05_dflap03	12/01/2016 22:42	Text Document
aerodynamics_wing_cflap05_dflap04	12/01/2016 23:27	Text Document
aerodynamics_wing_cflap05_dflap05	13/01/2016 00:18	Text Document

File name: "aerodynamics_wing_cflap05_dflap05" "aeroc

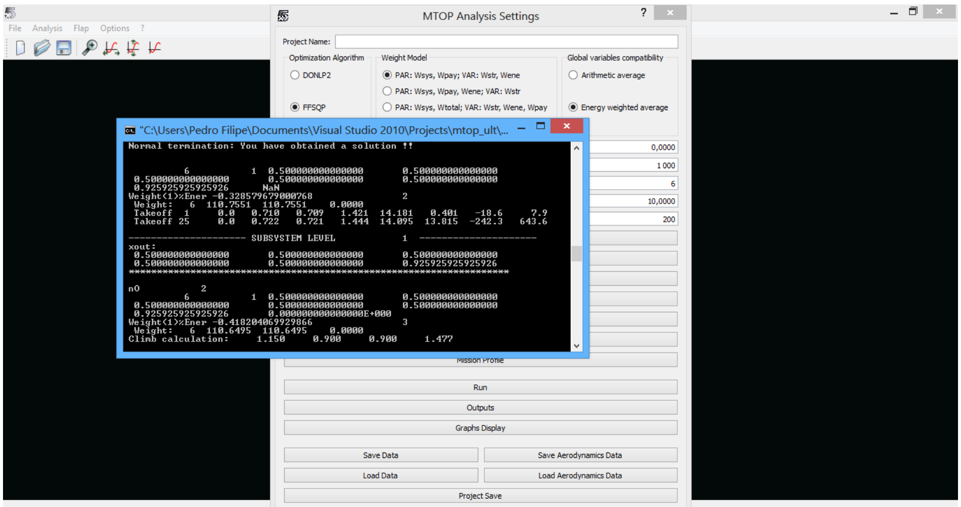
XFLR5 file (*.txt)

Figure 2.23: Flaps aerodynamic coefficients files.

2.2.3 Processing

Once all the inputs have been dully filled, the next step is to run the MTOP code. To do that the user should click on the button *Analyze* (Figure 2.3). As the analysis starts, a black window will appear before MTOP Main Menu, which will display many relevant data in real time (Figure 2.24).

Depending on the analysis settings, the running of the MTOP code can take from a few minutes up to several hours. However, the user should be aware that if the process gets stopped, i.e., if the MTOP output window is not updated for a long period of time, it has probably got to an unfeasible solution. If that is the case, the user should stop the process and evaluate the potential causes of such happening.



MTOP Analysis Settings

Project Name:

Optimization Algorithm: ○ DONLP2 ● FFSQP

Weight Model: ● PAR: Wsys, Wpay; VAR: Wstr, Wene ○ PAR: Wsys, Wpay, Wene; VAR: Wstr ○ PAR: Wsys, Wtotal; VAR: Wstr, Wene, Wpay

Global variables compatibility: ○ Arithmetic average ● Energy weighted average

0,0000
1 000
6
10,0000
200

"C:\Users\Pedro Filipe\Documents\Visual Studio 2010\Projects\mtop_ult\...

```
Normal termination: You have obtained a solution !!


          6          1  0.500000000000000       0.500000000000000
 0.500000000000000       0.500000000000000       0.500000000000000
 0.925925925925926       NaN
Weight(1)%Ener -0.328579679000768               2
 Weight:   6  110.7551  110.7551    0.0000
 Takeoff  1      0.0   0.710   0.709   1.421  14.181   0.401   -18.6     7.9
 Takeoff 25      0.0   0.722   0.721   1.444  14.095  13.815  -242.3   643.6

--------------------- SUBSYSTEM LEVEL           1  ---------------------
xout:
 0.500000000000000       0.500000000000000       0.500000000000000
 0.500000000000000       0.500000000000000       0.925925925925926
*******************************************************************

n0           2
          6          1  0.500000000000000       0.500000000000000
 0.500000000000000       0.500000000000000       0.500000000000000
 0.925925925925926       0.000000000000000E+000
Weight(1)%Ener -0.418204069929866               3
 Weight:   6  110.6495  110.6495    0.0000
Climb calculation:     1.150     0.900     0.900     1.477
```

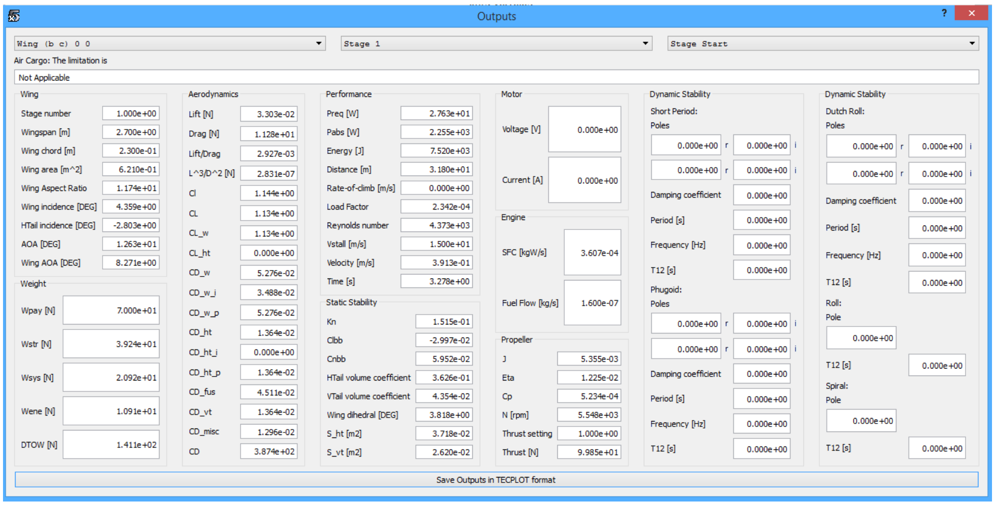
Figure 2.24: MTOP Analysis Running.

2.2.4 Outputs

After the analysis process is complete, the outputs are ready to be displayed. The user must click on the *Outputs* button to open the respective window (Figure 2.25).

This window has three combo boxes:

- the first one is to choose displaying the outputs of each different wing geometry (wing chord and wingspan combination);
- the second allows the user to choose between the different mission stages;
- the third one is to choose between displaying the output parameters at the beginning or at the end of the stage.



Wing (b c) 0 0

Stage 1

Stage Start

Air Cargo: The limitation is

Not Applicable

Wing

Stage number	1.000e+00
Wingspan [m]	2.700e+00
Wing chord [m]	2.300e-01
Wing area [m^2]	6.210e-01
Wing Aspect Ratio	1.174e+01
Wing incidence [DEG]	4.359e+00
HTail incidence [DEG]	-2.803e+00
AOA [DEG]	1.263e+01
Wing AOA [DEG]	8.271e+00

Weight

Wpay [N]	7.000e+01
Wstr [N]	3.924e+01
Wsys [N]	2.092e+01
Wene [N]	1.091e+01
DTOW [N]	1.411e+02

Aerodynamics

Lift [N]	3.303e-02
Drag [N]	1.128e+01
Lift/Drag	2.927e-03
L^3/D^2 [N]	2.831e-07
Cl	1.144e+00
CL	1.134e+00
CL_w	1.134e+00
CL_ht	0.000e+00
CD_w	5.276e-02
CD_w_i	3.488e-02
CD_w_p	5.276e-02
CD_ht	1.364e-02
CD_ht_i	0.000e+00
CD_ht_p	1.364e-02
CD_fus	4.511e-02
CD_vt	1.364e-02
CD_misc	1.296e-02
CD	3.874e+02

Performance

Preq [W]	2.763e+01
Pabs [W]	2.255e+03
Energy [J]	7.520e+03
Distance [m]	3.180e+01
Rate-of-climb [m/s]	0.000e+00
Load Factor	2.342e-04
Reynolds number	4.373e+03
Vstall [m/s]	1.500e+01
Velocity [m/s]	3.913e-01
Time [s]	3.278e+00

Static Stability

Kn	1.515e-01
Clbb	-2.997e-02
Cnbb	5.952e-02
HTail volume coefficient	3.626e-01
VTail volume coefficient	4.354e-02
Wing dihedral [DEG]	3.818e+00
S_ht [m2]	3.718e-02
S_vt [m2]	2.620e-02

Motor

Voltage [V]	0.000e+00
Current [A]	0.000e+00

Engine

SFC [kgW/s]	3.607e-04
Fuel Flow [kg/s]	1.600e-07

Propeller

J	5.355e-03
Eta	1.225e-02
Cp	5.234e-04
N [rpm]	5.548e+03
Thrust setting	1.000e+00
Thrust [N]	9.985e+01

Dynamic Stability

Short Period:

Poles

0.000e+00	r	0.000e+00	i
0.000e+00	r	0.000e+00	i

Damping coefficient	0.000e+00
Period [s]	0.000e+00
Frequency [Hz]	0.000e+00
T12 [s]	0.000e+00

Phugoid:

Poles

0.000e+00	r	0.000e+00	i
0.000e+00	r	0.000e+00	i

Damping coefficient	0.000e+00
Period [s]	0.000e+00
Frequency [Hz]	0.000e+00
T12 [s]	0.000e+00

Dynamic Stability

Dutch Roll:

Poles

0.000e+00	r	0.000e+00	i
0.000e+00	r	0.000e+00	i

Damping coefficient	0.000e+00
Period [s]	0.000e+00
Frequency [Hz]	0.000e+00
T12 [s]	0.000e+00

Roll:

Pole

0.000e+00

T12 [s]	0.000e+00

Spiral:

Pole

0.000e+00

T12 [s]	0.000e+00

Figure 2.25: Outputs window.

At the bottom of the window there is the button *Save Outputs in TECPLOT format* which generates a TECPLOT format file with all the listed output data. This is one possible way of generating parametric plots, which can make the analysis between the different wing geometries significantly more straightforward.

2.2.5 Saving and Loading Options

In the MTOP main menu (Figure 2.3) there are four buttons that have not yet been mentioned. These are:

1. Load Data - loads the input data ".txt" file;
2. Save Data - saves the previously loaded data in the user defined directory;
3. Save Aerodynamic Data - saves the airfoils'aerodynamic data into ".txt" files in the user defined directory;
4. Save Project - opens a window to choose a directory where the user wants to save all the input and output data in a folder with the respective project name.

2.3 Post-Processing

The handling of the numeric output is up to the user preferences. Since the output data is automatically written in various files, there are countless alternatives. The authors advise the users to consider using the TECPLOT or an analogous software package, once it enables a swift way of creating multidimensional plots.

Chapter 3

Disclaimer

The authors are not liable for any inadequate use of the current software. Its use should be limited to designers with a solid and comprehensive background on the design of unmanned aerial vehicles, namely in the fields of aerodynamics, flight mechanics, performance, stability and propulsion. Its use shall be limited to the preliminary stage of aircraft design and it should by no means be used to attest compliance with any certification specifications.

34

Bibliography

[1] Pedro F. Albuquerque, Pedro V. Gamboa, and Miguel A. Silvestre. Multidisciplinary and Multilevel Design Methodology of Unmanned Aerial Vehicles Using Enhanced Collaborative Optimizat. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 9(4):470–479, 2015. 1

[2] Joaquim R. R. A. Martins and Andrew B. Lambe. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal*, 51:2049–2075, 2013. 2

[3] Brian Roth and Ilan Kroo. Enhanced Collaborative Optimization: Application to an Analytic Test Problem and Aircraft Design. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC, Canada*, 2008. 2

[4] Brian Roth and Ilan Kroo. Enhanced Collaborative Optimization: A Decomposition-Based Method for Multidisciplinary Design. *ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008*, 2008. 2

[5] P. Spellucci. A SQP method for general nonlinear programs using only equality constrained subproblems. *Math. Prog.*, 82:413–448, 1998. 2

[6] P. Spellucci. A new technique for inconsistent problems in the SQP method. Math. Meth. of Oper. Res. *Math. Meth. of Oper. Res.*, 47:355–400, 1998.

[7] Jian L. Zhou, Andr L. Tits, and Craig T. Lawrence. User's Guide for FFSQP Version 3.7: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints1. *Electrical Engineering Department and Institute for Systems Research University of Maryland, College Park, MD 20742*, 1992. 2

Appendix F

Motor & Engine Specifications

Table F.1: Motor & Engine Specifications.

MOTOR	**Scorpion SII-4025-520KV**
Stator Diameter	40.0 mm
Stator Thickness	25.0 mm
No. of Stator Arms	12
Motor Wind	11 Turn Delta
Motor Wire 26-Strand	0.25 mm
Motor Kv	520 RPM / Volt
No-Load Current (Io)	@ 10V 1.40 Amps
Motor Resistance (Rm)	0.016 Ohms
Max Continuous Current	100 Amps
Max Continuous Power	2000 Watts
Weight	353 Grams (11.96 oz)
Outside Diameter	48.9 mm
Shaft Diameter	5.98 mm
Body Length	54.1 mm
Overall Shaft Length	85 mm
Max Lipo Cell	6s
Motor Timing	5deg
ENGINE	**3W-55i**
Cylinder Capacity	55 ccm
Power Rating	3.82 kW
Bore Diameter	44 mm
Stroke	35 mm
Weight	1905 g
Speed Range	1300 - 8500 RPM
Crankshaft	3 Ball bearings
Oil / Gasoline Ratio	1:50 - 1:80 Mix
Ignition Input Voltage:	6.0 to 8.0 V 6.0 V NiCD / NiMH or 7.2 V LiPo / Li Ion

Appendix G

MTOP Case Studies Mission's Design Variables

The optimized design variables values for each mission, optimization case are listed in Tables G.1 through G.3.

Table G.1: Optimized variables - MTOP case study mission I.

Case	Variable	TO	CB	CZ	DT
I.A	$\bar{c}[m]$	0.25			
	$b[m]$	2.65			
	$d_{prop}['']$	14.76			
	$p_{prop}['']$	9.80			
I.B	$\bar{c}[m]$	0.25			
	$b[m]$	3.55	3.41	2.65	2.65
	$d_{prop}['']$	14.84			
	$p_{prop}['']$	9.77			
I.C	$\bar{c}[m]$	0.25			
	$b[m]$	2.65			
	$d_{prop}['']$	14.77			
	$p_{prop}['']$	6.07	10.00	10.00	8.01
I.D	$\bar{c}[m]$	0.25			
	$b[m]$	2.65			
	$d_{prop}['']$	14.79			
	$p_{prop}['']$	9.76			
	$c_{flap}/\bar{c}$	0.29			
	$\delta_{flap}[deg]$	-5.00	-1.50	-5.00	15.00
I.E	$\bar{c}[m]$	0.25			
	$b[m]$	2.65			
	$d_{prop}['']$	14.78			
	$p_{prop}['']$	6.03	10.00	10.00	8.59
	$c_{flap}/\bar{c}$	0.30			
	$\delta_{flap}[deg]$	-5.00	-1.20	5.00	15.00
I.F	$\bar{c}[m]$	0.25			
	$b[m]$	3.55	3.53	2.93	2.65
	$d_{prop}['']$	14.86			
	$p_{prop}['']$	6.06	10.00	10.00	10.00
	$c_{flap}/\bar{c}$	0.36			
	$\delta_{flap}[deg]$	-5.00	-1.49	-4.19	-4.04

Table G.2: Optimized variables - MTOP case study mission II.

Case	Variable	TO	CB1	CZ1	LO1	CB2	CZ2	DT1	LO	CZ3	DT2
II.A	$\bar{c}[m]$	0.26									
	$b[m]$	2.94									
	$d_{prop}['']$	15.46									
	$p_{prop}['']$	10.00									
II.B	$\bar{c}[m]$	0.26									
	$b[m]$	3.55	3.05	2.65	3.55	3.05	3.24	2.65	3.55	2.65	2.65
	$d_{prop}['']$	15.70									
	$p_{prop}['']$	10.00									
II.C	$\bar{c}[m]$	0.26									
	$b[m]$	2.94									
	$d_{prop}['']$	15.47									
	$p_{prop}['']$	6.08	8.94	10.00	10.00	10.00	10.00	8.48	10.00	10.00	8.48
II.D	$\bar{c}[m]$	0.25									
	$b[m]$	2.97									
	$d_{prop}['']$	15.42									
	$p_{prop}['']$	10.00									
	$c_{flap}/\bar{c}$	0.26									
	$\delta_{flap}[deg]$	-5.00	1.18	-5.00	1.21	-0.52	-5.00	-5.00	1.21	-5.00	-5.00
II.E	$\bar{c}[m]$	0.25									
	$b[m]$	2.97									
	$d_{prop}['']$	15.42									
	$p_{prop}['']$	6.04	8.34	10.00	10.00	10.00	10.00	8.71	10.00	10.00	8.71
	$c_{flap}/\bar{c}$	0.26									
	$\delta_{flap}[deg]$	-5.00	-5.00	-5.00	1.20	-0.60	-5.00	-5.00	1.20	-5.00	-5.00
II.F	$\bar{c}[m]$	0.26									
	$b[m]$	3.55	3.10	2.65	3.55	3.11	3.55	2.65	3.55	2.65	2.65
	$d_{prop}['']$	15.63									
	$p_{prop}['']$	6.11	9.51	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
	$c_{flap}/\bar{c}$	0.38									
	$\delta_{flap}[deg]$	-5.00	-3.61	-5.00	4.19	-0.91	-4.36	-3.93	4.19	-5.00	-3.91

Table G.3: Optimized variables - MTOP case study mission III.

Case	Variable	TO	CB	CZ1	LT	CZ2	DT
III.A	$\bar{c}[m]$	0.27					
	$b[m]$	3.34					
	$d_{prop}['']$	20.06					
	$p_{prop}['']$	17.99					
III.B	$\bar{c}[m]$	0.28					
	$b[m]$	3.55	2.67	2.65	3.55	2.65	2.65
	$d_{prop}['']$	20.08					
	$p_{prop}['']$	17.98					
III.C	$\bar{c}[m]$	0.27					
	$b[m]$	3.34					
	$d_{prop}['']$	20.06					
	$p_{prop}['']$	18.00	16.54	18.00	18.00	18.00	18.00
III.D	$\bar{c}[m]$	0.27					
	$b[m]$	3.35					
	$d_{prop}['']$	20.06					
	$p_{prop}['']$	17.98					
	$c_{flap}/\bar{c}$	0.29					
	$\delta_{flap}[deg]$	-5.00	-5.00	-5.00	1.12	-5.00	15.00
III.E	$\bar{c}[m]$	0.27					
	$b[m]$	3.35					
	$d_{prop}['']$	20.06					
	$p_{prop}['']$	18.00	16.54	18.00	18.00	18.00	18.00
	$c_{flap}/\bar{c}$	0.29					
	$\delta_{flap}[deg]$	-5.00	-5.00	-5.00	1.12	-5.00	15.00
III.F	$\bar{c}[m]$	0.28					
	$b[m]$	3.55	2.76	2.65	3.55	2.65	2.65
	$d_{prop}['']$	20.08					
	$p_{prop}['']$	18.00	16.54	18.00	18.00	18.00	18.00
	$c_{flap}/\bar{c}$	0.22					
	$\delta_{flap}[deg]$	-5.00	-5.00	-5.00	1.14	-5.00	-5.00

Appendix H

Running Time

The total running time of the three mission profiles analyzed for each of the six optimization cases is listed in Table H.1. Mission I results were run on an Intel Core i7-3610QM CPU @ 2.30GHz; RAM 8.0GB, whereas the remaining simulations were ran on a Intel Core i7-6700K CPU @ 4GHz; RAM 16GB and sometimes other applications were running at the same time.

Table H.1: Running Time for each optimization case [s].

Optimization Case	Mission I	Mission II	Mission III
A	$6,513$	$35,647$	$34,005$
B	$5,825$	$33,832$	$13,029$
C	$7,934$	$40,499$	$26,434$
D	$8,718$	$20,813$	$23,501$
E	$8,933$	$25,422$	$25,837$
F	$11,544$	$42,223$	$22,073$

Appendix I

MTOP Outputs

Table I.1: MTOP Output Variables (I).

Discipline	Variable
Aircraft	Wingspan
	Wingspan
	Wing mean chord
	Wing area
	Wing aspect ratio
	Wing incidence
	Horizontal tail incidence
	Aircraft angle-of-attack
	Wing angle-of-attack
Aerodynamics	Lift force
	Drag force
	Lift-to-drag ratio
	Cubic lift to squared drag ratio
	Airfoil lift coefficient
	Aircraft lift coefficient
	Wing lift coefficient
	Horizontal tail lift coefficient
	Wing drag coefficient
	Wing induced drag coefficient
	Wing parasitic drag coefficient
	Horizontal tail drag coefficient
	Horizontal tail induced drag coefficient
	Horizontal tail parasitic drag coefficient
	Fuselage drag coefficient
	Vertical tail drag coefficient
	Miscellaneous drag coefficient
	Aircraft drag coefficient
Performance	Required power
	Absorbed power
	Energy
	Distance
	Rate-of-climb
	Load factor
	Reynolds number
	Stall velocity
	Aircraft velocity
	Time

Table I.2: MTOP Output Variables (II).

Discipline	**Variable**
Weight	Payload weight
	Structural weight
	Systems weight
	Energy weight
	Design take-off weight
Motor	Voltage
	Current
Engine	Specific fuel consumption
	Fuel flow
Propeller	Propeller advance ratio
	Propeller efficiency
	Propeller power coefficient
	Propeller rotations
	Thrust setting
	Thrust force
Static Stability	Static stability margin
	Rolling moment coefficient derivative with respect to the sideslip angle
	Yawing moment coefficient derivative with respect to the sideslip angle
	Horizontal tail volume coefficient
	Vertical tail volume coefficient
	Wing dihedral angle
	Horizontal stabilizer area
	Vertical stabilizer area
Dynamic Stability	Poles
	Damping
	Period
	Frequency
	Half-time decay

Appendix J

Publications

J.1 Journals

- P. F. Albuquerque, P. V. Gamboa and M. A. Silvestre, “Mission-Based Multidisciplinary Aircraft Design Optimization Methodology Tailored for Adaptive Technologies”, Journal of Aircraft, AIAA, [Accepted for publication in September, 2017].
- P. F. Albuquerque, P. V. Gamboa and M. A. Silvestre, “Multidisciplinary and Multilevel Design Methodology of Unmanned Aerial Vehicles Using Enhanced Collaborative Optimization”, International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering Vol. 9, No. 4, pp 470-479, 2015.
- P. F. Albuquerque, P. V. Gamboa and M. A. Silvestre, “Parametric Aircraft Design Optimisation Using Span, Mean Chord and Wing Aerofoil Lift Coefficient as Main Design Drivers”, Advanced Materials Research Vol. 1016, pp 365-369, 2014.

J.2 Conferences

- F. R. Fraqueiro, P. F. Albuquerque and P. V. Gamboa, “A Computer Application for Parametric Aircraft Design”, International Conference in Engineering of University of Beira Interior, University of Beira Interior, Covilhã, Portugal, 2015. [1]
- P. F. Albuquerque, P. V. Gamboa and M. A. Silvestre, “Multidisciplinary and Multilevel Aircraft Design Methodology using Enhanced Collaborative Optimization”, 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Dallas, TX, USA, 2015.[2]
- P. F. Albuquerque and P. V. Gamboa, “Preliminary Design of Unmanned Aerial Vehicles Using Multidisciplinary Design Methodologies”, 3rd LAETA Young Researchers Meeting, Coimbra, Portugal, 2015.
- P. F. Albuquerque, P. V. Gamboa and M. A. Silvestre, “Multidisciplinary and Multilevel Design Methodology of Unmanned Aerial Vehicles using Enhanced Collaborative Optimization”, World Academy of Science, Engineering & Technology, Lisbon, Portugal, 2015.
- P. F. Albuquerque, P. V. Gamboa and M. A. Silvestre, “Parametric Aircraft Design Optimisation Using Span, Mean Chord and Wing Aerofoil Lift Coefficient as Main Design Drivers”, International Conference in Mechanical and Aerospace Engineering, Science & Engineering Institute, Madrid, Spain, 2014.[3]

[1] Ranked as best English paper by reviewers on the parallel session of ICEUBI2015

[2] Shortlisted for AIAA Student Paper Competition

[3] Considered the best oral presentation of the Air Mechanical Engineering session

- P. V. Gamboa, M. A. Silvestre and P. F. Albuquerque, “Aircraft Design Methodology Using Span and Mean Aerodynamic Chord as Main Design Parameters”, International Conference in Engineering of University of Beira Interior, University of Beira Interior, Covilhã, Portugal, 2013.

J.3 Manuals

- P. F. Albuquerque, P. V. Gamboa and Filipe R. Fraqueiro, “PARROT - **P**arametric **A**i**R**c**R**aft **O**p**T**imization code User’s Manual”, University of Beira Interior, 2016.
- P. F. Albuquerque, P. V. Gamboa and Filipe R. Fraqueiro, “MTOP - **M**ultilevel aircraf**T** design **OP**timization code User’s Manual”, University of Beira Interior, 2016.