

# **Coupled Preliminary Design and Trajectory Optimization of Rockets using a Multidisciplinary Approach**

**Fábio Miguel Pereira Morgado**

Thesis to obtain the Master of Science Degree in

## **Aerospace Engineering**

Supervisors: Prof. Paulo Jorge Soares Gil  
Prof. André Calado Marta

### **Examination Committee**

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha  
Supervisor: Prof. Paulo Jorge Soares Gil  
Member of the Committee: Prof. João Manuel Gonçalves de Sousa Oliveira

**June 2019**



## **Acknowledgements**

I would like to thank to professor Paulo Gil and professor André Marta for supervising my work and providing valuable insights regarding the various topics of this Master Thesis.

To the RNL team for helping configuring the cluster. Without their help, this work would never be possible.

To my friends for believing in me and constantly motivating me during the last five years. A special thanks to my friends from crossfit for giving me motivation and discipline to overcome this phase in my life.

To my mother, father and brother for all their support, love, dedication and motivation. This would not be possible without you.



## Resumo

Neste trabalho, foi criada uma ferramenta para a optimização do design preliminar de um foguete. Utilizando uma abordagem multidisciplinar acoplada, a ferramenta é capaz de rapidamente encontrar os parâmetros óptimos de design e trajectória para a missão especificada.

A optimização do design é efectuada utilizando um algoritmo genético contínuo desenvolvido e testado no trabalho, com opção de processamento paralelo. Os modelos de massa e dimensionamento necessários para estimar a estrutura do foguete são criados a partir da regressão de dados históricos, ou retirados da literatura.

A optimização da trajectória é efectuada utilizando o princípio mínimo de Pontryagin. As equações de optimalidade são deduzidas e os valores óptimos são localizados utilizando o método de optimização por enxame de partículas. A curvatura da Terra e a contribuição da sua rotação são consideradas na simulação da trajectória. O modelo de arrasto é criado utilizando a data proveniente do software Missile Datcom, testando a influência da geometria do nariz com o número de Mach.

Os modelos são validados utilizando foguetes reais e os parâmetros do algoritmo genético são afinados facilitar a convergência da solução. Finalmente, a ferramenta é testada através da optimização de um pequeno foguete e comparando-o com o melhor foguete actual.

A ferramenta apresenta resultados promissores na optimização da trajectória e do design. Consegue lidar com os constrangimentos impostos e é capaz de realizar o desenho conceptual de um foguete e o cálculo da trajectória num espaço de tempo razoável.

**Palavras-chave:** Abordagem acoplada, Veículo de lançamento, Optimização de trajectória, Algoritmo genético, Enxame de partículas



## Abstract

A tool was developed to perform a rocket preliminary design by finding the optimal design and trajectory parameters for a specific mission. A multidisciplinary coupled approach was used to optimize both trajectory and design.

The design optimization is performed using a continuous genetic algorithm, able to perform parallel computation, also developed and benchmarked in this work. The mass and sizing models required to estimate the rocket structure are created using historical data regression or taken from literature.

The trajectory optimization is done using the Pontryagin's minimum principle. The optimality equations are deduced and the optimal values are found using a particle swarm optimization. The Earth's curvature and the contribution due to Earth's rotation is taken into account to simulate the trajectory. The drag model is created using data from Missile DATCOM software to test the nose geometry influence with Mach number.

The models are validated using real rockets and the influence of the coast phase in trajectory optimization is analyzed using the Vega rocket. The genetic algorithm is subjected to a simplified parameter tuning. Finally, the tool is tested by optimizing the design of a small launch vehicle and comparing it to the state-of-the-art rocket.

The tool shows promising results in both trajectory and design optimization. It handles the imposed constraints and is able to successfully perform a launch vehicle conceptual design and trajectory calculation in a reasonable time.

**Keywords:** Coupled approach, Launch vehicle, Trajectory optimization, Genetic algorithm, Particle swarm optimization



# Contents

List of Figures . . . . .	xii
List of Tables . . . . .	xiv
Nomenclature . . . . .	xvi
Glossary . . . . .	xxi
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Rocket Design Challenges . . . . .	2
1.3 Rocket Design and Optimization . . . . .	3
1.4 Objectives and Deliverables . . . . .	4
1.5 Thesis Outline . . . . .	4
<b>2 Rocket Fundamentals</b>	<b>5</b>
2.1 Rocket Performance . . . . .	5
2.2 Staging . . . . .	6
2.2.1 Serial Staging . . . . .	7
2.2.2 Parallel Staging . . . . .	9
2.3 Propulsion . . . . .	10
2.4 Trajectory . . . . .	12
2.4.1 Lift Off and Pitch Over Maneuver . . . . .	13
2.4.2 Gravity Turn . . . . .	13
2.4.3 Free Flight Phase . . . . .	14
2.5 Structure . . . . .	14
2.5.1 Structural System . . . . .	14
2.5.2 Buckling . . . . .	15
<b>3 Rocket Design and Optimization</b>	<b>17</b>
3.1 MDO Application to Launch Vehicles . . . . .	17
3.2 Optimization Algorithms . . . . .	19

3.2.1	Genetic Algorithm . . . . .	21
3.2.2	Particle Swarm Optimization . . . . .	22
3.3	Trajectory Optimal Control . . . . .	24
3.3.1	Direct methods . . . . .	24
3.3.2	Indirect methods . . . . .	25
<b>4</b>	<b>Optimal Rocket Design Procedure</b>	<b>27</b>
4.1	Dry Mass Estimation and Sizing . . . . .	27
4.1.1	Solid Propulsion Systems . . . . .	27
4.1.2	Liquid Propulsion Systems . . . . .	31
4.1.3	Payload Adapter and Fairing . . . . .	33
4.2	Trajectory Model . . . . .	35
4.2.1	Equations of Motion . . . . .	35
4.2.2	Free Flight Phase . . . . .	37
4.3	Atmospheric Model . . . . .	40
4.4	Drag Model . . . . .	41
4.5	Algorithm Development . . . . .	42
4.5.1	Genetic Algorithm Implementation . . . . .	43
4.5.2	Rocket Construction and Evaluation . . . . .	45
<b>5</b>	<b>MDO and Models Validation</b>	<b>51</b>
5.1	Mass Model Validation . . . . .	51
5.1.1	Solid Rocket: Vega . . . . .	51
5.1.2	Liquid Serial Staging Rocket: Proton K . . . . .	53
5.1.3	Parallel Staging Rocket: Ariane 5 ECA . . . . .	54
5.2	Testing the Trajectory Model . . . . .	56
5.3	Genetic Algorithm Benchmark . . . . .	59
<b>6</b>	<b>Preliminary Design of a Small Launch Vehicle</b>	<b>65</b>
6.1	Algorithm Setup . . . . .	66
6.2	Results . . . . .	68
<b>7</b>	<b>Conclusions</b>	<b>79</b>
7.1	Summary and Achievements . . . . .	79
7.2	Future Developments . . . . .	80
	<b>Bibliography</b>	<b>80</b>
	<b>Appendices</b>	<b>91</b>





# List of Figures

1.1	First rocket launch at Cape Canaveral [2]. . . . .	1
2.1	Mass definitions for serial staging [35]. . . . .	7
2.2	Variation of total payload ratio of a multi-stage rocket with performance parameter $\frac{\Delta v}{v_e}$ [30]. . . . .	8
2.3	Mass definitions for parallel staging [35]. . . . .	9
2.4	Typical flight sequence of a space launch vehicle. [43] . . . . .	12
2.5	Structural system of a liquid multi-stage rocket. [51] . . . . .	15
3.1	Scheme of the Multidisciplinary Feasible method [12]. . . . .	18
3.2	Classification of the main MDO formulations [12]. . . . .	19
3.3	Genetic algorithm procedure flowchart [71]. . . . .	22
3.4	Particle swarm optimization procedure flowchart [70]. . . . .	23
4.1	Comparison between real and predicted engine mass (under 500 kg). . . . .	28
4.2	Comparison between real and predicted engine mass (above 500 kg). . . . .	28
4.3	Comparison between real and predicted solid stage inert mass. . . . .	29
4.4	Comparison between real and predicted solid booster inert mass. . . . .	30
4.5	Fit between stage length and propellant grain length. . . . .	31
4.6	Comparison between real and predicted fairing length. . . . .	34
4.7	Comparison between real and predicted fairing mass. . . . .	35
4.8	Rocket state variables and forces during flight [34]. . . . .	36
4.9	Influence of Mach number in Drag coefficient [98]. . . . .	41
4.10	Render of the half power series nose cone. . . . .	42
4.11	Genetic Algorithm implementation using master-slave architecture. . . . .	44
4.12	Flowchart of rocket construction and trajectory simulation. . . . .	45
4.13	Data flow of the rocket construction block. . . . .	46
4.14	Data flow of the trajectory optimization block. . . . .	47
4.15	Flowchart of trajectory model. . . . .	48

5.1 Vega thrust vectoring. . . . .	57
5.2 Vega altitude. . . . .	57
5.3 Vega velocity. . . . .	58
5.4 Vega flight path angle. . . . .	58
5.5 Velocity loss due to drag. . . . .	58
5.6 Velocity loss due to gravity. . . . .	58
5.7 Velocity loss due to thrust vectoring. . . . .	58
5.8 Rocket total mass along generations. . . . .	62
5.9 Dynamic pressure during flight . . . . .	63
5.10 Axial load during flight. . . . .	63
5.11 Rocket acceleration during flight. . . . .	64
5.12 Rocket thrust during flight. . . . .	64
6.1 Rockets best total mass evolution. . . . .	68
6.2 Analysis of feasible and unfeasible two-stage rockets. . . . .	69
6.3 Analysis of feasible and unfeasible three-stage rockets. . . . .	69
6.4 Number of trajectory calls per generation. . . . .	70
6.5 Rocket altitude evolution. . . . .	71
6.6 Rocket velocity evolution. . . . .	72
6.7 Rocket flight path angle evolution. . . . .	72
6.8 Rocket thrust vectoring evolution. . . . .	73
6.9 Rocket velocity loss due to thrust vectoring. . . . .	73
6.10 Rocket velocity loss due to thrust drag. . . . .	74
6.11 Rocket velocity loss due to gravity. . . . .	74
6.12 Dynamic pressure before coast phase. . . . .	75
6.13 Rocket acceleration before coast phase. . . . .	75
6.14 Rocket axial load before coast phase. . . . .	75
6.15 Rocket thrust before coast phase. . . . .	75
6.16 Optimized rockets and Electron rocket dimensions. . . . .	76

# List of Tables

2.1	Common rocket propellants and respective $I_{sp}$ range [30]. . . . .	11
4.1	P-value of solid stage mass regression variables. . . . .	28
4.2	Statistics from engine dry mass regression. . . . .	29
4.3	Liquid propellant characteristics. . . . .	32
4.4	P-value of fairing length regression variables. . . . .	33
4.5	Statistics from fairing length regression. . . . .	34
4.6	Variable p-value of fairing mass regression. . . . .	34
4.7	Statistics from fairing mass regression. . . . .	34
4.8	List of parameters handled by the algorithm. . . . .	43
5.1	Vega rocket characteristics [39]. . . . .	52
5.2	Vega mass error analysis. . . . .	52
5.3	Vega sizing error analysis. . . . .	52
5.4	Proton K rocket characteristics [40]. . . . .	53
5.5	Proton K mass error analysis. . . . .	54
5.6	Proton K sizing error analysis. . . . .	54
5.7	Ariane 5 ECA rocket characteristics [41]. . . . .	55
5.8	Ariane 5 ECA mass error analysis. . . . .	55
5.9	Ariane 5 ECA sizing error analysis. . . . .	55
5.10	Mission specification for trajectory validation. . . . .	56
5.11	Trajectory parameters for trajectory validation. . . . .	56
5.12	Optimal trajectory parameters and final rocket state. . . . .	57
5.13	Optimization benchmark functions . . . . .	59
5.14	Optimization benchmark function solutions and selected parameters. . . . .	60
5.15	Comparison between implemented GA and DEAP's GA. . . . .	60
5.16	Convergence influence study of the DR parameter. . . . .	61
5.17	Mission specification for GA validation. . . . .	61
5.18	Constraints for GA validation. . . . .	61

5.19 Design parameters for GA validation. . . . .	62
5.20 Merlin engine characteristics. . . . .	62
5.21 Individual evaluations. . . . .	62
5.22 Optimal design parameters. . . . .	63
5.23 Optimal trajectory parameters. . . . .	63
6.1 Mission specification. . . . .	66
6.2 Trajectory parameters for optimization. . . . .	66
6.3 Propulsive and propellant parameters. . . . .	67
6.4 Design Parameters. . . . .	67
6.5 Imposed constraints. . . . .	67
6.6 Genetic algorithm parameters. . . . .	68
6.7 Particle swarm optimization parameters. . . . .	68
6.8 Two-stage rocket optimal design parameters. . . . .	70
6.9 Three-stage rocket optimal design parameters. . . . .	70
6.10 Rocket optimal trajectory parameters. . . . .	71
6.11 Comparison of design characteristics between optimized and Electron rocket. . . . .	76
6.12 Comparison of propulsive characteristics between optimized and Electron rocket. . . . .	76
A.1 Solid motor characteristics for regression [90] . . . . .	93
A.2 Solid stages and boosters characteristics for regression . . . . .	94
A.3 Fairing characteristics for regression . . . . .	94

# Nomenclature

## Greek Symbols

$\alpha$	Angle of attack
$\gamma$	Flight path angle
$\gamma_p$	Pitch maneuver angle
$\Delta t_c$	Coast duration
$\Delta t_T$	Last stage duration
$\Delta V$	Change of velocity
$\Delta V_{drag}$	Velocity loss due to drag
$\Delta V_{gravity}$	Velocity loss due to gravity
$\Delta V_{orbit}$	Ideal velocity to reach orbit
$\Delta V_{thrust}$	Velocity loss due to thrust vectoring
$\delta$	Latitude
$\eta_s$	Safety factor
$\lambda$	Time-dependent adjoint variables vector
$\lambda$	Molecular mean free path, mutation step-size, payload ratio
$\Lambda$	Mass ratio
$\mu_E$	Earth gravitational parameter
$\nu$	Time-independent adjoint variables vector
$\rho$	Density, distance from Earth center

$\sigma$	Structural ratio
$\Phi$	Boundary condition Function
$\phi$	Aerothermal Flux
$\chi$	Thrust vector angle
$\Psi$	Boundary condition vector

### Latin Symbols

$A$	Area
$a$	Acceleration
$A_c$	Propellant tank cylindrical section area
$A_s$	Propellant tank spherical section area
$c_1$	Cognitive parameter
$c_2$	Social parameter
$C_D$	Drag coefficient
$D$	Diameter, drag
$DR$	Domain reduction
$\mathbf{f}$	State vector
$Ft_u$	Allowable material strength
$g$	Gravitational acceleration
$g_0$	Gravity acceleration at Earth surface
$H$	Hamiltonian
$h$	Altitude
$I_{sp}$	Specific impulse
$J$	Objective function
$K_n$	Knudsen number
$L$	Characteristic length, lagrangian, length
$\dot{m}$	Mass flow

$M$	Mach number
$m$	Mass
$P$	Pressure
$p$	Best particle position
$P_b$	Burst pressure
$p_c$	Crossover rate
$p_m$	Mutation rate
$R$	Air specific gas constant
$R_e$	Earth radius
$S$	Characteristic area
$s$	Molecular speed ratio
$s_c$	Constraint penalty factor
$T$	Temperature, thrust
$t$	Time
$t_f$	Final flight time
$t_{cf}$	Final coast time
$t_{ic}$	Coast initialization time
$th$	Thickness
$th_c$	Propellant tank cylindrical section thickness
$th_s$	Propellant tank spherical section thickness
$u$	Control variables
$V$	Velocity, volume
$v$	Particle velocity
$V_e$	Exhaust velocity
$v_e$	Effective exhaust velocity
$V_{ER}$	Earth velocity contribution

$w$	Inertia parameter
$x$	State variables
$x$	Downrange, particle position

### Superscripts

'	Required value
*	Optimal value

### Subscripts

0	Initial
$a$	Atmospheric
$e$	Exit nozzle
$F$	Fuel
$f$	Final
$LE$	Liquid engine
$LS$	Liquid stage
$mat$	Material
$n$	Normal component
$O$	Oxidizer
$p$	Propellant
$pl$	Payload
$PLA$	Payload adapter
$PLF$	Payload fairing
$s$	Structural
$SM$	Solid motor
$SS$	Solid stage
$st$	Support structure
$t$	Tangential component

<i>tankF</i>	Fuel tank
<i>tankO</i>	Oxidizer tank
<i>tc</i>	Thrust chamber



# Acronyms

**AAO** All At Once

**ATC** Analytical Target Cascading

**BLISS** Bi-Level Integrated Systems Synthesis

**CO** Collaborative Optimization

**CSSO** Concurrent SubSpace Optimization

**DDO** Disciplinary Design Optimization

**DIRECT** Divided RECTangles

**DR** Domain Reduction

**FPI** Fixed Point Iteration

**GA** Genetic Algorithm

**HTPB** Hydroxyl-terminated Polybutadiene

**IDF** Individual Discipline Feasible

**IP** Interior-Point

**LEO** Low Earth Orbit

**LH2** Liquid Hydrogen

**LOX** Liquid Oxygen

**LV** Launch Vehicle

**MDA** Multidisciplinary Design Analysis

**MDF** Multidisciplinary Feasible

**MDO** Multidisciplinary Design Optimization

**MMH** Monomethylhydrazine

**N2O4** Nitrogen Tetroxide

**NLP** Nonlinear Programming Problem

**PBAN** Polybutadiene Acrylonitrile

**PMP** Pontryagin's Minimum Principle

**PSO** Particle Swarm Optimization

**RP1** Rocket Propellant - 1 (Kerosene)

**SA** Simulated Annealing

**SQP** Sequential Quadratic Programming

**SSTO** Single-Stage To Orbit

**SWORD** Stage-Wise decomposition for Optimal Rocket Design

**TPBVP** Two Point Boundary Value Problem

**UDMH** Unsymmetrical Dimethylhydrazine

# Chapter 1

## Introduction

Space launchers design is a task of great difficulty that involves a large number of components and subsystems, affecting the vehicle performance. Thus, small adjustments in the design parameters may have a great influence in the subsystems, flight path and payload capacity.

In this work, a tool is developed for coupled multidisciplinary optimization of preliminary rocket design, dividing the design into several disciplines to efficiently deal with the complexity.

### 1.1 Motivation

Over the past few years, the competitiveness in the space sector increased drastically. It is of major importance to build efficient rocket to reduce costs and increase space access. The costs are conditioned by the technological advancements, along with the payload, propellant and structural mass [1].

Space companies design new launchers while seeking the minimum cost configuration able to perform a set of reference missions, such as sending a satellite of some mass to orbit. Current technology allied to continuous development of Multidisciplinary Design Optimization (MDO) algorithms are powerful tool to build cheaper and better rockets.



Figure 1.1: First rocket launch at Cape Canaveral [2].

## 1.2 Rocket Design Challenges

Currently, a large number of different launchers are available in the market and several others are being developed using new technological and design concepts. Space launch vehicles are required to perform several activities, such as launch satellites from the Earth's surface to specified orbits, access the International Space Station, or simply for space tourism. The primary goal of the aerospace community is to reduce the design and mission costs, transforming space access into a routine procedure [3].

The design of a rocket is a very challenging activity. As it is an high-energy systems, it tends to be extremely complex when safety, reliability and performance are considered, leading to high costs. A substantial part of the overall launcher development is committed at the conceptual design phases [4], and at least 80% of the life-cycle costs are comprised by the chosen design concept [5]. In this phase, only a few characteristics of the system are fixed, making MDO techniques interesting as they can handle large design space in a multidisciplinary environment, improving system performance and decreasing design cycle cost and time [6].

To find the best launcher configuration, multiple configurations can be tested during the early design phases. The optimal launcher is selected by evaluating the performance and costs of the alternative designs.

A launch vehicle performance and costs are not only influenced by the vehicle characteristics, but also by the flight trajectory. An optimal flight trajectory minimizes the thrusting time and reduces the structural stress in the early flight phases, leading to a reduction in the launcher mass and cost. To find the optimal launcher design, tools capable of performing design and trajectory optimization are required, but they demand high computational costs due to the extensive numerical modeling, simulation and optimization [1].

The space launcher design is a multidisciplinary process, involving aerodynamics, propulsion, mass and sizing, trajectory and costs. As the number of parameters increases, the design search space largely expands, which may cause convergence problems in finding the best design. In addition, during the early design phases, the design constraints are not so strict and there are several design variables still undetermined. For this reason, low-fidelity models of the various disciplines are used, which may result in inefficient designs [1].

As suggested by [7] to reduce the complexity and life-cycle cost of space launchers, there is the need to significantly improve early systems analysis capability in the conceptual and preliminary design phases. This can be done by improving the computational methods for space search and optimization, improving the fidelity of disciplinary engineering models and improving the multidisciplinary integration of the engineering disciplines to reduce workload and design cycle time.

Currently, due to the commercialization of space travel, private and public companies are emerging and providing valuable contribution to space technology. The company SpaceX, founded in 2002, was the first company to recover the first stage of an orbital rocket (Falcon 9) [8], drastically reducing the mission costs.

## 1.3 Rocket Design and Optimization

The design of space launch vehicles involves the interaction of diverse engineering disciplines which often have conflicting objectives and demand a vast search space to find the global optimum. This requires adapted design multidisciplinary tools which allow to integrate the constraints inherent to each discipline and to ease the compromise search.

Traditionally, launch vehicle design optimization use the Multidisciplinary Feasible (MDF) method. This method consists in splitting the design problem according to the different disciplines and associating a global optimizer at the system level, while complying with all discipline constraints [9, 10, 11]. However, the handling of all the design variables by the single optimizer may induce a very large search space, posing several problems such as the requirement of an appropriate initialization and a good knowledge of the design variable search domain in order to converge [12]. This led to the creation of MDO frameworks involving several levels of optimization, such as the use of Collaborative Optimization (CO) [13] or the Bi-Level Integrated Systems Synthesis (BLISS) [14].

The need for specialized MDO methods for space launch vehicles led to the development of new optimization methods. One example, is the Stage-Wise decomposition for Optimal Rocket Design (SWORD) method [12]. While classical MDO methods tackle the trajectory as a whole, the SWORD method decomposes the rocket according to the different stages in order to improve the efficiency of the MDO process.

The optimization process has to handle different kinds of errors and uncertainties [15]. The optimal solution is conditioned by noise in the evaluation of objective functions and approximation errors due to the use of meta-models. Three major issues have to be addressed when using MDO: model fidelity, solution optimality, and solution robustness [16].

The computational advances over the last few decades enabled the use of better models. Nonetheless, high-fidelity models generally imply expensive computation costs, making reduced models using metamodelling techniques a reliable choice for preliminary design. Over the past years, there has been a shift from response surface models to alternative approximation methods [17], such as neural-network metamodel [18], support vector regression [19] and kriging modelling [20].

The optimal solution may be hard to find due to the large number of variables and constraints and the presence of local minima. This problem is frequently addressed by using metaheuristics (heuristic methods applicable to a wide set of different problems) to find the optimal design, such as evolutionary algorithms. Several works use Genetic Algorithm (GA) to explore the design space and optimize the vehicle, while being able to handle multi-objective problems [21, 22].

The use of approximation models are sufficient to correctly estimate the design features of the vehicle [23]. However, the trajectory optimization problem requires numerical optimization methods to be implemented. Trajectory optimization is a key point in designing a launch vehicle [24] and may be solved either by using direct methods (e.g., pseudospectral knotting [25] and collocation methods [26]) or indirect methods utilizing Pontryagin's Minimum Principle (PMP) [27].

## 1.4 Objectives and Deliverables

The main objective of this work is the creation of an algorithm capable of designing and/or optimizing rockets, according to the given mission specifications and design variables to optimize. The optimization process led to the development of aerodynamic and mass models, as well as trajectory optimization. One of the main problems that had to be overcome was the lack of computational power to run the algorithm. Therefore, a parallelization option had to be included.

The expected deliverables are:

1. A state-of-the-art overview of the different techniques used in multi-disciplinary and trajectory optimization;
2. Development and validation of mass, aerodynamic and 2-D trajectory models;
3. Development and validation of a numerical tool to perform trajectory and design optimization using a master-slave architecture;
4. Research of current small rocket launcher technology, followed by the design and performance analysis of a small launch vehicle using the created optimization algorithm.

## 1.5 Thesis Outline

The first three chapters present the historical and theoretical background to better comprehend the work performed. In chapter 1, an overall view of the rocket design optimization theme is provided, along with the motivation for this research. The fundamental disciplines and equations for rocket design are explored in chapter 2, explaining the Tsiolkovski rocket equation, the different rocket configurations and systems and providing an overview of the general trajectory. The historical development of MDO techniques is presented in chapter 3. Additionally, it provides information on current optimization techniques and trajectory control. An introduction to the genetic algorithm and particle swarm optimization is also presented.

Chapter 4 describes the construction of the mass and sizing, drag and trajectory models, along with the development of the optimization algorithm to reduce the total rocket mass for a specific mission. The models and the algorithm are tested and tuned in chapter 5. The tool is tested in chapter 6 by performing a conceptual rocket design optimization of a small rocket. Thus, a comparison with the Electron rocket is made.

Lastly, chapter 7 draws the conclusion of the study. It includes a brief summary of the thesis and recommendations for future research.

## Chapter 2

# Rocket Fundamentals

Since the beginning of the space age, rocketry has evolved from a single-engine to multi-engine rockets, increasing the rockets weight-lifting capability by ten thousand times and leading to the construction of Saturn V [28]. The Saturn V rocket still holds the record for the heaviest payload launched and largest payload capacity to Low Earth Orbit (LEO) of 140 tonnes [29].

Currently, there are several types of concepts regarding Single-Stage To Orbit (SSTO) and multi-stage rockets to perform a mission, such as the use of boosters to aid at lift-off and the use of reusable or expendable stages. However, all of these concepts share one thing: rocket and boosters develop thrust by the rearward ejection of mass at very high velocity, experiencing an increase in momentum proportional to the momentum carried away in the exhaust.

### 2.1 Rocket Performance

A mission requires the rocket to achieve a specified velocity to deliver the payload into the desired orbit. Disregarding external forces, the velocity change is determined by the Tsiolkovsky rocket equation [30]

$$\Delta V = V_f - V_0 = v_e \ln \frac{m_0}{m_f}, \quad (2.1)$$

where  $\Delta V$  is the maximum change of velocity,  $v_e$  the effective exhaust velocity,  $m_0$  the initial rocket mass and  $m_f$  the final rocket mass. A simple model for preliminary design is considering  $v_e$  constant for each rocket stage, but a more accurate approximation can be made by varying the parameter according to the atmospheric pressure [31].

In reality, the rocket is subjected to external forces, the most important being drag and gravity, causing energy losses throughout the mission profile. The energy losses may be expressed in terms of velocity losses, transforming the required mission velocity change into [30]

$$\Delta V = \Delta V_{orbit} + \Delta V_{losses} = \Delta V_{orbit} + \Delta V_{drag} + \Delta V_{gravity} + \Delta V_{thrust}, \quad (2.2)$$

where  $\Delta V_{orbit}$  is the ideal velocity required for the rocket to achieve in order to reach the desired orbit, and  $\Delta V_{drag}$  and  $\Delta V_{gravity}$  are the velocity losses due to drag and gravity, respectively. The term  $\Delta V_{thrust}$  is the wasted velocity due to steering, so the vehicle can correct its trajectory by using gimbaled thrust systems or vernier thrusters.

By definition, the gravity loss is

$$\Delta V_{gravity} = \int_{t_0}^{t_f} g \sin \gamma \, dt, \quad (2.3)$$

where  $g$  represents the gravitational acceleration and  $\gamma$  the flight path angle. The initial time  $t_0$  and final time  $t_f$  are the boundaries of the time interval to calculate the velocity loss, usually comprising the entire flight duration. The losses due to gravity are only null when the gravity vector is orthogonal to the velocity vector. As the gravitational acceleration component changes with altitude and flight angle, the losses need to be calculated for every time step considered.

The drag loss also vary in time and is given by

$$\Delta V_{drag} = \int_{t_0}^{t_f} \frac{D}{m} dt, \quad (2.4)$$

where  $D$  is the drag force, and  $m$  the rocket mass.

The velocity loss induced by steering is

$$\Delta V_{thrust} = \int_{t_0}^{t_f} \left( \frac{T}{m} - \frac{T}{m} \cos \chi \right) dt, \quad (2.5)$$

where  $\chi$  is the angle between the velocity vector and thrust vector, and  $T$  is the rocket thrust at the given time.

Determining the required velocity for a rocket to perform a mission is, as it will be seen, an iterative problem. To accelerate convergence, velocity losses have to be initially estimated. A reasonable preliminary estimation is considering the losses typically between 1.5 to 2 km/s [32], predominantly due to gravity, but better approximations can be used during the optimization process.

## 2.2 Staging

A single-stage-to-orbit Launch Vehicle (LV) avoids the costs and complexity of staging, but the payload able to carry is too small for economically feasible operations. Staging allows the vehicle not to transport all the structure to orbit, thus saving propellant. Using a multi-stage configuration, for launches to LEO, the payload is typically about 1% to 3.5% of the vehicle mass [33]. Depending on the technology used, the

rocket can carry up to 6% of the total mass [34].

To reduce lift-off mass, two types of staging can be used: serial staging and parallel staging. The equations for both staging types are carefully explained in [30].

## 2.2.1 Serial Staging

In serial staging, the stages are stacked upon each other and the thrust is provided by one stage at a time. When the propellant depletes, the engines are turned off and the stage discarded from the LV, reducing the dead weight. Usually, the first stage is the largest stage while the subsequent upper stages decrease in size.

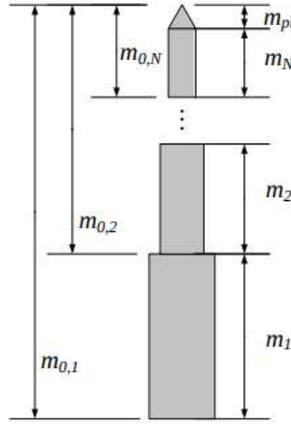


Figure 2.1: Mass definitions for serial staging [35].

Analyzing a N-stage rocket, the  $k_{th}$  stage mass is given as

$$m_k = m_{p,k} + m_{s,k}, \quad (2.6)$$

where  $m_s$  represents the structural mass and  $m_p$  the propellant mass.

The payload of any stage is equivalent to the mass sum of the higher stages, with the exception of the last stage ( $m_{pl,N} = m_{pl}$ , with  $m_{pl}$  being the payload mass), as illustrated in figure 2.1, is given by

$$m_{pl,k} = m_{0,k+1}. \quad (2.7)$$

where  $m_{0,k}$  is the sum of the  $k_{th}$  stage mass and payload mass.

To ease the rocket analysis, it is common the use of dimensionless mass ratios. The structural ratio is a dimensionless measure of how much of the stage mass is structural. The stage structural mass comprises only the dry mass, and the structural ratio of the  $k_{th}$  stage is defined as

$$\sigma_k = \frac{m_{s,k}}{m_k}. \quad (2.8)$$

The structural factor typically varies between 0.05 to 0.15. Current space launcher vehicles suggest that the first stages have a structural factor of 0.05 to 0.08, while values of 0.1 to 0.15 are common for the upper stages [34].

It is also convenient to define the payload ratio, that will give information of how much mass is necessary to carry a specific payload. The payload ratio of the  $k$ th stage is defined as

$$\lambda_k = \frac{m_{0,k+1}}{m_k}, \quad (2.9)$$

and the total payload ratio, that indicates how much of the initial mass of the vehicle is payload, is defined as

$$\lambda_T = \frac{m_{pl}}{m_{0,1}}. \quad (2.10)$$

The payload ratio is a measure of the rocket's usefulness, and the structural coefficient is a measure of the degree of optimization of the engineering design. Heavy designs or non-optimized propellant tanks will increase the structural ratio.

The mass ratio of any particular stage can be given as a function of the structural ratio and the payload ratio as

$$\Lambda_k = \frac{m_{0,k}}{m_{f,k}} = \frac{1 + \lambda_k}{\sigma_k + \lambda_k}, \quad (2.11)$$

where  $m_{f,k}$  is the sum of the  $k$ th stage structural mass and payload mass.

For a multi-stage rocket with a total of  $N$  stages, the ideal velocity increments is the sum of the individual stage contribution. The Tsiolkovsky's rocket equation (2.1) can be rewritten as

$$\Delta V = \sum_{k=1}^N v_{e,k} \ln \frac{m_{0,k}}{m_{f,k}} = \sum_{k=1}^N v_{e,k} \ln \Lambda_k. \quad (2.12)$$

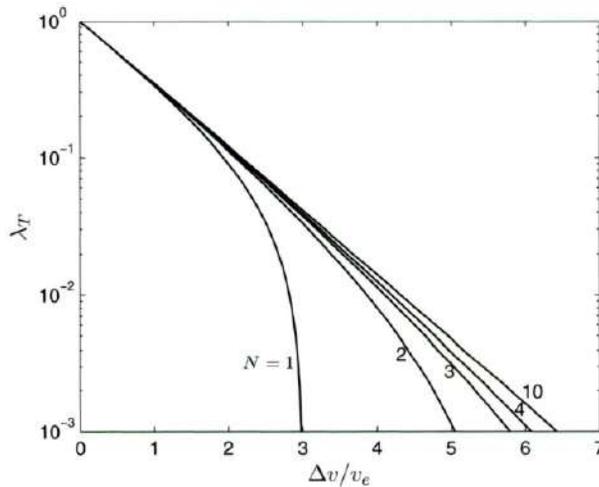


Figure 2.2: Variation of total payload ratio of a multi-stage rocket with performance parameter  $\frac{\Delta v}{v_e}$  [30].

Although the rocket performance and efficiency increase as new stages are included in the rocket design, as illustrated in figure 2.2, the complexity, risk and construction cost substantially rise, diminishing returns. Rockets typically have 2 to 3 stages, as the increase in performance due to staging is small when using more stages [30].

## 2.2.2 Parallel Staging

In parallel staging, the parallel boosters and the first stage burn simultaneously to assist with launch. It has the advantage of increasing the thrust in the early flight phase to account for an increased payload, without changing the fundamental design of the core rocket [31]. There is a vast number of modern launchers employing parallel staging, usually by attaching boosters to the first stage of the core vehicle. Generally, boosters have very high thrust, implying a high mass-flow rate, so they burn for a shorter time when compared to the core's first stage [31].

When the parallel boosters and the core first stage burn simultaneously, it is called zeroth stage. After the boosters detachment, the remaining propellant in the core's first stage is called first stage of the rocket and has to be taken in consideration separately from the zeroth stage [30].

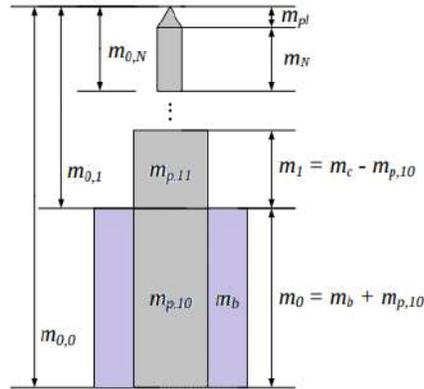


Figure 2.3: Mass definitions for parallel staging [35].

Observing figure 2.3, the core's rocket first stage has a total initial mass  $m_c$  and the boosters have a total mass of  $m_b$ . During the zeroth stage, only the propellant mass  $m_{p,10}$  at the core is consumed. Simultaneously, the boosters burn a total propellant mass  $m_{p,b}$ . Thus, the structural and payload ratios of the zeroth stage are given by

$$\sigma_0 = \frac{m_{s,b} + m_{s,1}}{m_b + m_{s,1} + m_{p,10}} \quad (2.13)$$

and

$$\lambda_0 = \frac{m_{0,1} - m_{s,1}}{m_b + m_{s,1} + m_{p,10}}, \quad (2.14)$$

respectively. The term  $m_{s,b}$  stands for the booster dry mass, and  $m_{s,1}$  for the core's first stage dry mass. Similarly, the ratios for the rocket's first stage are

$$\sigma_1 = \frac{m_{s,1}}{m_{s,1} + m_{p,11}} \quad (2.15)$$

and

$$\lambda_1 = \frac{m_{0,2}}{m_{s,1} + m_{p,11}}. \quad (2.16)$$

From the first stage onward, the rocket is built using serial staging configuration. The Tsiolkovsky rocket equation for parallel staging is equivalent to the serial staging configuration, in section 2.2.1. The average exhaust velocity of the zeroth stage can be written as

$$v_{e0} = \frac{v_{eb}\dot{m}_b + v_{e1}\dot{m}_1}{\dot{m}_b + \dot{m}_1}, \quad (2.17)$$

where  $v_{eb}$  and  $v_{e1}$  are the effective exhaust velocity of the boosters and core's first stage, respectively. The term  $\dot{m}_b$  represents the booster's propellant mass flow and  $\dot{m}_1$  the core's stage propellant mass flow.

The boosters are usually attached in a circumferential arrangement and radially symmetric to prevent the rocket from tipping. The maximum number of boosters for an existing launcher is 9 for the Delta II 7900 series [36], where six of the boosters are ignited at take-off and the remaining at mid-air, after the detachment of the depleted boosters. The boosters do not necessarily need to be disposed in a symmetric position, such is the case of Atlas V, that compensates the asymmetry by tilting the engine [37].

## 2.3 Propulsion

Launch vehicles consume propellant to generate the thrust required to complete the mission. The energy source most commonly used for propulsion is chemical combustion, based on energy released inside a combustion chamber [38]. After the chemical reaction, the stationary gases contain enormous amounts of heat energy. The thrust is then generated by transforming the heat energy into kinetic energy by accelerating the gas through the nozzle.

There are many types of rocket propulsion systems, but the most used are the liquid and solid. Independently of the chemical propulsive system used, thrust is a result of the change of the gas momentum due to the transformation of heat into kinetic energy, defined as

$$T = \dot{m}V_e + A_e(P_e - P_a), \quad (2.18)$$

where  $\dot{m}$  is the mass flow rate through the nozzle,  $V_e$  the exhaust velocity,  $A_e$  the exit nozzle area, and  $P_e$  and  $P_a$  the exit nozzle pressure and atmospheric pressure, respectively. The thrust is highly dependent of the atmospheric pressure, changing accordingly with the rocket altitude. For this reason, the lower stages

usually have a lower nozzle area, while upper stages have nozzles optimized for vacuum propulsion. The thrust equation can also be rewritten as function of the effective exhaust velocity

$$T = \dot{m}v_e \quad (2.19)$$

with

$$v_e = V_e + \frac{A_e(P_e - P_a)}{\dot{m}}. \quad (2.20)$$

A simple propulsive model for preliminary rocket design is to consider a constant thrust. However, a more complex model can be used by varying thrust as a function of time.

To compare different propellants and engines, the specific impulse parameter is used. It has units of second, describing the total impulse delivered per unit weight of propellant, given by

$$I_{sp} = \frac{T}{\dot{m}g_0} = \frac{v_e}{g_0}, \quad (2.21)$$

where  $g_0 = 9.80665 \text{ m/s}^2$  is the acceleration of gravity at the Earth's surface. Typical  $I_{sp}$  values for solid propellant motors vary between 180 to 270 seconds, while liquid propellant engines may vary between 260 to 475 seconds [30]. Some of the most common rocket propellants are shown in table 2.1.

Propulsion system	Propellant	Isp [sec]
Solid	PBAN	260-263
	HTPB	260-265
Liquid	N2O4/MMH	260-310
	N2O4/UDMH	260-310
	RP1/LOX	300-350
	LH2/LOX	455

Table 2.1: Common rocket propellants and respective  $I_{sp}$  range [30].

The use of liquid engines has serious disadvantages comparatively to solid motors. The main disadvantages are the complex design of the commonly used turbo-pump feeding systems and the storage difficulty of the oxidizer due to its instability, increasing manufacturing costs. However, they present the capability of being throttled, shut down and restarted, vital to correct trajectory after an orbital or sub-orbital coast phase [38].

As for solid motors, they do not require a propellant feeding system, and the solid propellant is easily storable and relatively safe to handle after it is produced. On the other hand, the specific impulse is lower than liquid engines and the motor cannot be controlled once ignited. The solid engine continues to provide thrust until all the propellant is exhausted. The thrust profile can be defined by the shape of the propellant charge in a solid rocket motor [38].

Modern space launchers usually have solid boosters, providing a very high thrust during boost phase while being simple and cheap to assemble. The upper stage generally have liquid engines due to orbit

precision requirements. As for the remaining stages, some rockets use solid motors, such as the Vega [39], but actual trend is to use liquid engines, for example the Falcon 9 [8], Proton-M [40] and Ariane 5 [41]. Thus, space companies are progressively using less harmful propellants, such as Liquid oxygen (LOX) and highly refined kerosene (RP1) instead of derivatives of hydrazine in combination with nitrogen oxides (N<sub>2</sub>O<sub>4</sub>/MMH and N<sub>2</sub>O<sub>4</sub>/UDMH), which are toxic and carcinogenic [38].

## 2.4 Trajectory

Space launchers have to reach a specific orbit to deliver the desired payload, which is done by following an ascent trajectory path. The ascent trajectory is influenced by the launch site, the propulsive characteristics of the LV, the aerodynamic and gravitational forces acting on the vehicle during flight and the parameters of the injection orbit [30]. It also needs to take into account the safety and operational constraints to reduce malfunction risk.

The trajectory path has a major influence in the design and performance of the LV, increasing production and operating costs. If the trajectory is not optimized, it may have significant effects on the payload mass that can be carried to a specific orbit [42]. The vehicle also needs to be able to perform trajectory corrections due to unexpected factors, such as the wind or engine malfunctions. The control is mainly done by thrust vectoring and throttling.

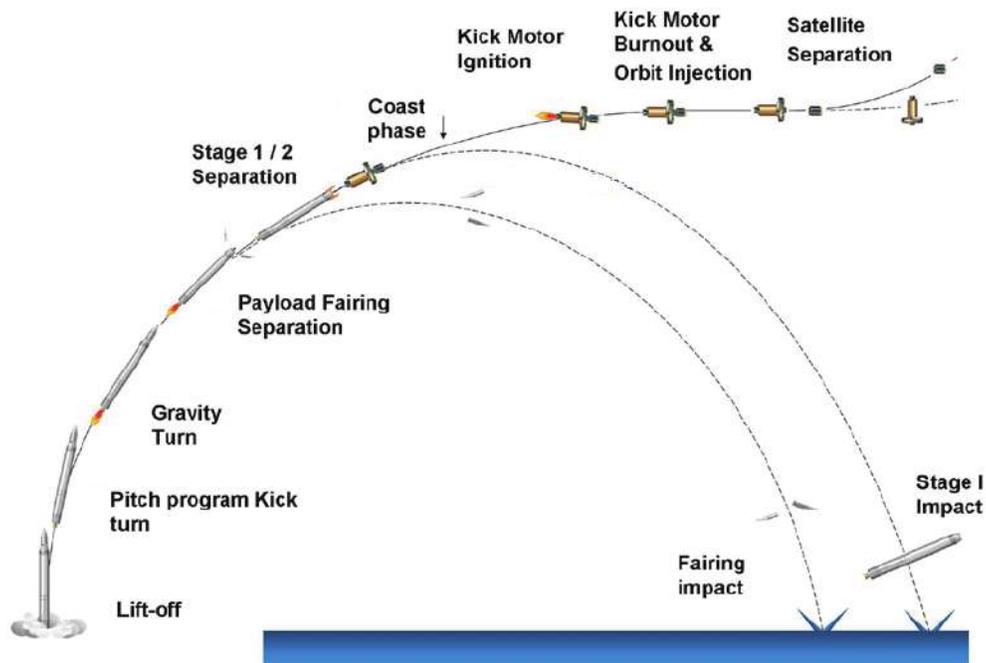


Figure 2.4: Typical flight sequence of a space launch vehicle. [43]

The trajectory can be divided into several phases as shown in figure 2.4 [30]. The respective phases are described in the next sections.

### **2.4.1 Lift Off and Pitch Over Maneuver**

The majority of space LVs take off from the ground launch pad, in the vertical direction, where the predominant forces acting in the rocket are thrust and weight. During the vertical ascent, the vehicle gains both speed and altitude, while maintaining a zero angle of attack to avoid lateral loads [44]. The rocket tries to leave atmosphere as soon as possible, to reduce drag losses. However, a steep ascent leads to more gravity losses as more energy is required to overcome gravity. Hence, the vehicle performs a pitch over maneuver at a certain instant, for the vehicle to start rotating gradually towards horizontal position [45].

As soon as the rocket clears any service towers and gains sufficient speed, the pitch over maneuver starts by using vernier thrusters or gimbaling the main engines, deviating the thrust vector from the symmetry axis and generating a torque about the center of gravity. This action produces a small pitch and establishes the initial flight path angle to start turning the rocket, usually by using a gravity turn trajectory explained in section 2.4.2. It is best to perform the initial pitch over at low dynamic pressure, as the vehicle will briefly have a small angle of attack during the pitch time, producing a minimal lift force. If the dynamic pressure is high, the slightest angle of attack can produce destructive transverse loads on the vehicle [31].

Due to technological advancements, the length of the vertical segment has tended to decrease, starting the pitch over maneuver earlier and reducing the gravity losses, which can be as much as 20% of the total velocity required [31].

### **2.4.2 Gravity Turn**

Orbital speed is fundamentally horizontal so the vehicle needs to rotate and gain horizontal speed. This is generally done by using a gravity turn trajectory, where the vehicle uses only gravity to steer onto the desired trajectory. The gravity turn trajectory starts right after the pitch over maneuver nudges the vehicle away from the vertical attitude, only ending when the vehicle leaves atmosphere.

The use of a gravity turn allows the vehicle to maintain a practically null angle of attack throughout the atmosphere, while it accelerates through the maximum dynamic pressure zone. This allows the minimization of the transverse aerodynamic stress on the LV, reducing the risk of structural failure and allowing the design of lighter LVs [46]. To hold the null angle of attack, the thrust vector is kept aligned with the velocity vector of the vehicle during the entire gravity turn maneuver.

The gravity turn maneuver ends when dynamic pressure becomes negligible, allowing the vehicle to have a non-zero angle of attack without damaging the structure. At this point, the shroud protecting the payload is dropped off to reduce the vehicle dry mass, and the free flight phase begins.

### **2.4.3 Free Flight Phase**

Starting the free flight phase, a more efficient trajectory path than the gravity turn can be followed because the vehicle does not require to hold the null angle of attack, as there will be no more significant aerodynamic loads applied [44]. By resorting to thrust vectoring and throttling, the vehicle can be controlled more efficiently to reach the desired orbit. Thus, it allows to reduce the overall propellant needed, leading to cost reduction.

Typically, during this phase, the rocket tries to gain enough velocity to perform a coast to the final orbital altitude by reducing the flight path angle, decreasing the gravity losses, and by consuming the propellant as fast as possible, to reduce mass. Afterwards, the rocket is injected into the specified orbit by restarting the engine, which will provide a small impulse, ideally as short as possible [31]. The inclusion of a coast arc can lead to substantial propellant savings, and duration usually increases as the injection altitude increases [47]. The vehicle may perform several coast arcs, depending on the propulsive technology used.

## **2.5 Structure**

The rocket structure has to be able to withstand the external forces applied during flight, but also has to be lightweight as the performance and costs of the vehicle are highly dependent on the structural mass. The choice of the proper materials is of extreme importance for aerospace applications, as they need to comply with several constraints due to safety operation requirements [48], while being one of the major factors in the rocket's performance. Also, the materials need to withstand exposure to high or cryogenic temperatures due to the aerodynamic heating during ascent and cryogenic propellant storage, respectively. Typical materials are aluminum and titanium alloys [49].

### **2.5.1 Structural System**

The structural system comprises the propellant tanks and primary structures, as illustrated in figure 2.5. The main function of the structural system is to protect the rocket from the external loads by distributing them through the entire frame. Generally, the rocket body is long and slender, providing low aerodynamic drag during atmospheric flight.

At the top of the body, a conical or tapered shell is used to provide a low-drag shape and to protect enclosed payloads during the atmospheric flight. If the shell encloses the payload, it is called shroud or payload fairing, otherwise it is known as nose cone, used on top of the strap-on boosters.

During ascent, the shroud is subjected to high heat fluxes and must incorporate heat-resistant materials, such as carbon fibers. Being at the extreme forward end of the vehicle, the shroud does not need to sustain as much axial load as other structures, turning stiffness relatively more important than strength, to maintain the shape of the payload compartment. The fairing design also needs to take in account high-intensity

sound waves that reverberate around the launch pad after engine ignition, which could be intense enough to damage the payload [50].

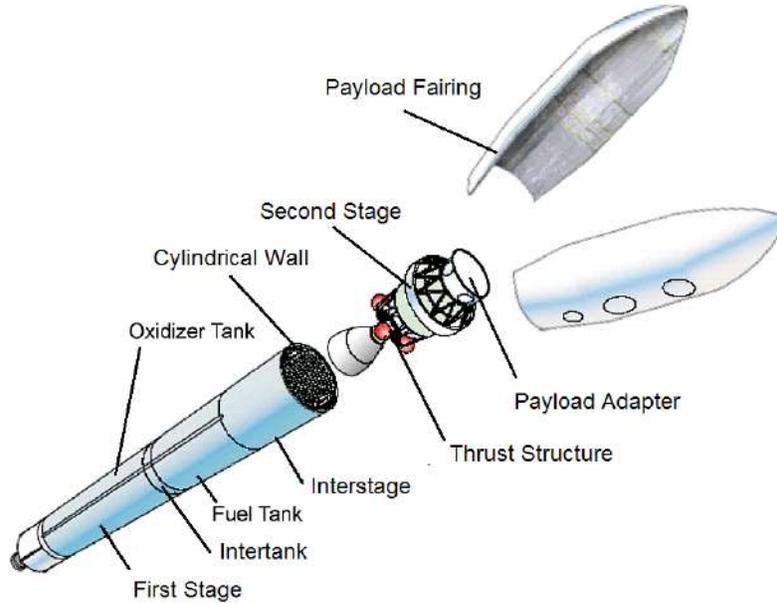


Figure 2.5: Structural system of a liquid multi-stage rocket. [51]

## 2.5.2 Buckling

During flight, a rocket is subjected to compressive forces as a result of gravity and inertia while thrusting. To prevent the rocket from collapsing and to protect it from external forces, rockets have a cylindrical thin wall structure. The wall has to be as thin as possible to decrease the rocket mass. However, thinning the structure leads to loss of the strength and, therefore, have to be tested before-hand. A possible problem with thin structures is the occurrence of buckling. Only buckling due to axial compression is considered and is explained in this section.

Buckling occurs when the structure under compressive load reaches a critical point, where it begins to deform. The deformed wall can still carry the loading, but it is lower than the critical load value. However, when the rocket structure reaches the critical point, structural failure is almost guaranteed to occur.

For a thin elastic cylindrical shell of radius  $R$ , thickness  $th$ , and Young modulus  $E$ , the linearized buckling equations, the critical stress is given by [52]

$$\sigma_{crit} = \frac{E}{\sqrt{3(1-\nu^2)}} \left( \frac{th}{R} \right), \quad (2.22)$$

where  $\nu$  is the material Poisson's ratio. For an aluminum-alloy,  $\nu = 0.32$ . The critical stress is independent of the structure length. However, this equation only works for cylindrical shells of intermediate length, and rocket stages are assumed to belong in the category. The critical load is given by

$$L_{crit} = \sigma_{crit}\pi((R^2) - (R - th)^2). \quad (2.23)$$

If the axial load reaches the critical load, the structure buckles. It is common to use a safety factor, providing a safety margin to prevent structural failure. Typically, rocket structural walls have a minimum safety factor of 1.4 [48]. To increase the critical load, a rocket cylinder may have internal stiffeners, but they will not be on the scope of this work.

## Chapter 3

# Rocket Design and Optimization

Designing a launch vehicle involves several engineering disciplines. In traditional engineering design, experts in the various disciplines adopted the Disciplinary Design Optimization (DDO) method, where an optimal solution is obtained for every subsystem. However, it may lead to a suboptimal design and requires a large amount of time [53].

To solve this problem, Multidisciplinary Design Optimization (MDO) methods were created. MDO deals with the coupled design problem as a whole by integrating all the disciplines at the same time and finding the global optimal design solution. The complexity of the problem increases substantially but can lead to a decreased computational cost [54].

### 3.1 MDO Application to Launch Vehicles

In literature, classical MDO methods applied to the design of space LVs typically decompose the design problem into the different disciplines, namely, aerodynamics, propulsion, structure, weight and sizing, costs and trajectory [12]. Each of the disciplines can be subjected to design and coupling constraints, depending on the mission requirements. The disciplinary constraints can be difficult to satisfy and considerably limit the feasible search domain.

The flight trajectory is a dynamic system coupled with all the other disciplines and subjected to strict equality constraints that has to be optimized for each rocket design configuration. Trajectory optimization requires a trajectory simulation with stage separations, which is complex to integrate. The use of MDO methods allows the combination of the design variables and trajectory optimization, making it suitable for space launchers design [12].

Currently, the most used MDO method for general design optimization is the MDF method. The MDF uses a single-level optimization formulation, i.e., requires only one optimizer at the system-level and a Multidisciplinary Design Analysis (MDA) to solve the interdisciplinary coupling equations, performed at each

iteration of the optimization process [12]. The MDA is typically performed by using the Fixed Point Iteration (FPI) method [55], consisting in a loop between the different discipline analysis until the coupling constraints are complied and convergence is achieved. The MDF schematic is represented in figure 3.1.

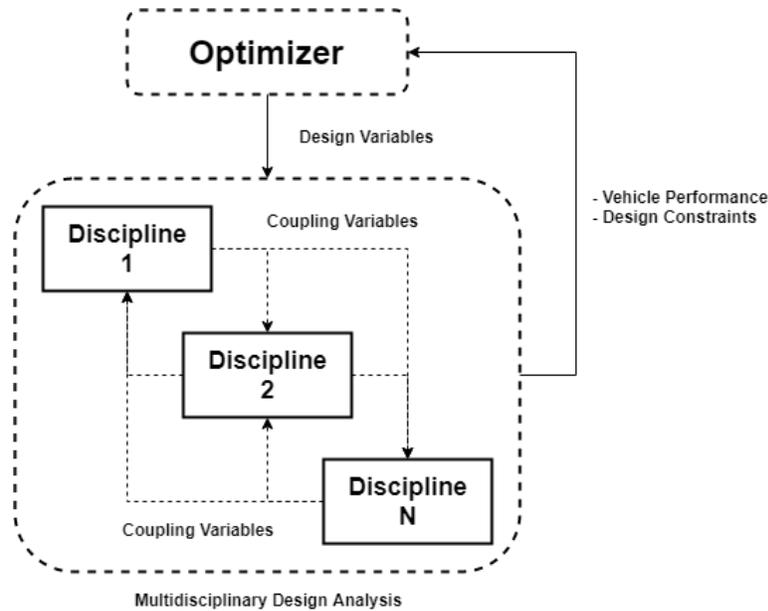


Figure 3.1: Scheme of the Multidisciplinary Feasible method [12].

Using the MDF formulation, the optimization variables are the design variables passed to the different disciplines. The disciplines are analyzed sequentially due to the coupling between downstream and upstream disciplines. At the end of each iteration, the optimizer evaluates the design performance and verifies if the design complies with the given constraints. The main advantages of using the MDF method is its simplicity and each iteration gives a feasible solution, even if stopped prematurely [56].

The method also presents important drawbacks. The computational cost of using MDF is high, as the MDA has to be executed at each iteration of the optimization process and does not take advantages of the possible parallelization between different disciplines. Thus, if a gradient-based optimizer method is used, a complete MDA is necessary at every point where the derivatives have to be evaluated. Also, the method is not guaranteed to converge [53].

Other single-level MDO methods used in the industry are the Individual Discipline Feasible (IDF) [57] and the All At Once (AAO) [58] methods. Contrarily to the MDF method, these methods do not use a MDA to solve the coupling. Instead, the optimizer alone takes into account the coupling variables, reducing computational cost. If using AAO, the optimizer also handles the state variables. These two methods present the advantage of being implementable in a network, for parallelization, since the analysis and evaluation of each subsystem can be made simultaneously. However, may not be applicable if using complex subsystems and the given solution is only feasible at convergence, when the consistency of the couplings between subsystems is guaranteed [53].

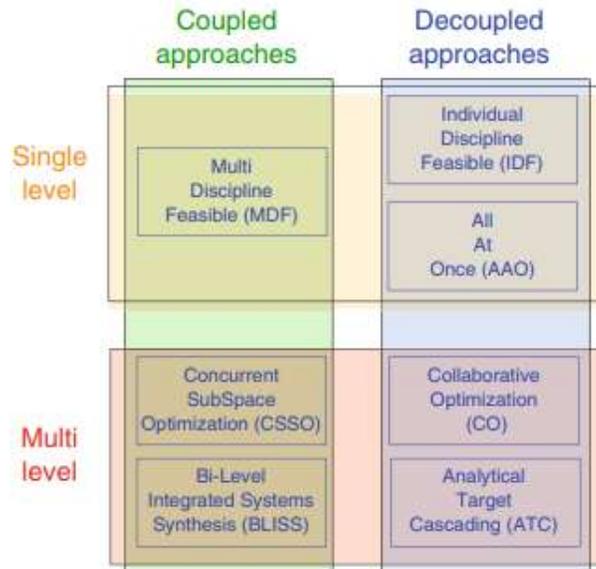


Figure 3.2: Classification of the main MDO formulations [12].

The MDO methods can also present multiple optimizers, which are explained in detail in [12]. Some examples are the Bi-Level Integrated Systems Synthesis (BLISS) [14] and the Concurrent SubSpace Optimization (CSSO) [59] methods, which also use MDA to solve the interdisciplinary coupling equations. Other methods, such as the Collaborative Optimization (CO) [13] and the Analytical Target Cascading (ATC) [60] methods use decoupled formulation, allowing the parallel analyses of the disciplines and reduced computational cost, while being able to find the coupled solution. However, decoupled approaches need to be carefully formulated to assure constraints compatibility between subsystems [58].

There also are specialized MDO methods for LV design optimization, such as the SWORD [12] method. The SWORD formulation allows the decomposition of the design problem according to the different stages, improving the efficiency of the MDO process. Using SWORD, the subsystems are not the disciplines, as in the classical MDO methods. Instead, it uses the different stages as subsystems, while incorporating all the required engineering disciplines for each stage.

For the tool, the MDF formulation was chosen. Being the most used optimization method, its simplicity allows an easy integration of the engineering disciplines, and the FPI method allows the convergence between them. Thus, using a MDF formulation allows to get a feasible design, even if the optimization stops prematurely due to an external error or command.

## 3.2 Optimization Algorithms

MDO methods require one or several optimization algorithms to work. Traditional optimization techniques search for the best solutions using gradients and/or random guesses. These algorithms are the tools to find

the best solution under given constraints, according to the optimization objective. The classical optimization algorithms can be divided into gradient-based and gradient-free algorithms, and choosing between them may have a large impact in the computational time [61].

The use of gradient-based algorithms imply the differentiation of the objective and constraint functions with respect to the design variables. Although being widely used due to the quick convergence when near a minimum, these algorithms present several drawbacks. Gradient-based algorithms can only optimize continuous parameters, requires the functions to be differentiable and, being a local method, can only find the local optima. However, they are guaranteed to find the local minimum in convex search space problems [62]. There are diverse methods to compute the gradient. The simplest sensitivity analysis method is finite-differences. If the evaluation is affected by round-off errors, complex-step method can be used [62]. In either case, the cost of evaluating the gradient is proportional to the number of design variables.

When the MDO method implements MDA to solve the coupling equations, sensitivity analysis methods can perform poorly. The MDA must be solved for each component of the function and constraint sensitivities, which can be costly. In order to reduce the MDA calls, semi-analytic sensitivity analysis methods should be used whenever possible. One example is the use of Coupled-Adjoint Sensitivity Analysis Method [63] which can handle hundreds of design variables, impractical for the use of finite-differences or the complex-step method. The cost of computing sensitivities using the coupled-adjoint procedure is almost independent of the number of variables.

To initialize a gradient-based algorithm, an initial-guess is made. At every iteration, the search direction and step-size are determined to calculate the next parameters set. The most commonly used gradient-based methods are the Sequential Quadratic Programming (SQP) and Interior-Point (IP). More information about gradient-based methods, can be found in [64].

The gradient-free algorithms, contrarily to the gradient-based algorithms, require only the availability of the function values. The key strength of gradient-free algorithms is the ability to solve highly discontinuous and noisy functions. Some algorithms even present the possibility of using discrete or mixed search spaces. These algorithms can perform global search. Furthermore, they allow the inclusion of a degree of randomness, not to get stuck in local minima [65]. The biggest drawback of using gradient-free algorithms is the increase in the number of evaluation calls required and the slow convergence when near a local optimum, leading to an increase in computational cost. There are two types of gradient-free algorithms: deterministic and stochastic.

In deterministic algorithms, the solution is fully determined by the parameter values and the initial conditions. A classical example of local-search deterministic algorithm is the Nelder-Mead simplex algorithm [66], capable of solving non-linear, unconstrained optimization problems. It is also possible to perform global optimization, using a global-search deterministic algorithm, such as the Divided RECTangles (DIRECT) method [67], by using an hyper-dimensional adaptive meshing scheme to search the entire design space.

Stochastic algorithms have some inherited randomness. Given the same initial conditions and parame-

ters, the solution may differ every time the algorithm is executed. Over the last two decades, there has been an increasing interest in heuristic approaches, which are typically inspired by natural phenomena. A few advantages are the ease implementation and being well suited for discrete optimization problems. Some heuristic methods also allow multi-objective optimization [68].

Some of the most popular stochastic algorithms are the Simulated Annealing (SA) [69], inspired in metallurgic annealing where heating and controlled cooling are used to achieve an optimal configuration of the material, the Particle Swarm Optimization (PSO) [70], which applies the concept of social behaviour in a group of organisms, and Genetic Algorithms (GA) [71], based on the concept of Darwin's theory of evolution.

Contrarily to the SA algorithm, the PSO and GA algorithms are population methods, where a set of individuals is created and updated throughout the generations. The initial population can be created randomly or can be performed an heuristic initialization. However, using exclusively heuristics results in little diversity and may fail to find the optimum [72].

Gradient-free algorithms typically perform unconstrained optimization, although most of the engineering problems are constrained. A common method to address the constraints is the use of penalty functions, converting the constrained problem into an unconstrained one by penalizing infeasible solutions, reducing their fitness values in proportion to their degrees of constraint violation [73]. The most common penalty methods are the static penalty (penalty parameters are constant) and dynamic penalty (penalty parameters are dependent on the current iteration).

From the gradient-free algorithms, the GA and PSO have been widely used in space industry, not only in design optimization [10], but also in trajectory optimization [74]. They are used in this work, and therefore, will be presented in the succeeding sections.

### **3.2.1 Genetic Algorithm**

The Genetic Algorithm is inspired in Darwin's theory of evolution, by the inclusion of selection, crossover and mutation techniques [71]. The basic procedure to implement a GA is illustrated in Fig. 3.3.

To start the procedure, an initial population has to be created and filled with individuals with a set of properties, which can be altered and mutated. Typically, the population is randomly created to improve the population diversity, or by using heuristic techniques, to initialize the optimization with good solutions [72]. During the optimization process the population size does not change. For each generation, the individuals are evaluated and ranked according to their fitness values, i.e, according to the objective function.

After the evaluation, a portion of the existing population is selected to breed a new generation. Usually, mating pools are created by extracting a random sample from the population. The parents are selected from the pool by using one of the several selection operators proposed in literature [75], which can be a random selection or based on fitness (e.g. tournament selection or roulette-wheel selection).

Once the parents have been selected, the next generation is created by mixing the selected parent designs in a process known as crossover. The created children always share the characteristics of the

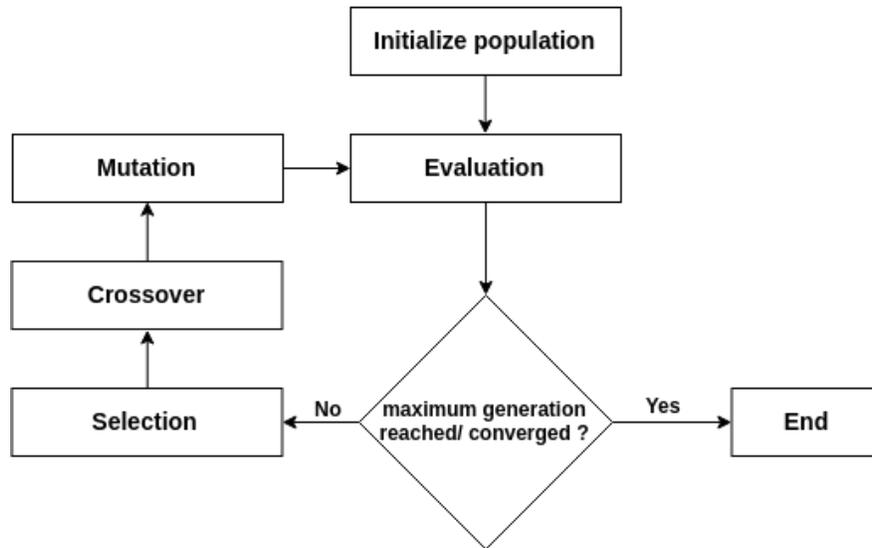


Figure 3.3: Genetic algorithm procedure flowchart [71].

selected parents and are expected to present good properties.

To maintain diversity in the population, the children can be modified by using a mutation operator. The mutation consists in randomly changing the properties of the individual, avoiding the individuals from becoming too similar to each other, which could result in slowing or even stopping evolution. The probability of a mutation to occur can be static or can be tuned along the optimization process. Usually, the probability of mutation and the corresponding variation are large at the beginning of the search process to explore more of the search space, decreasing progressively through generations [76].

When the mutation operation concludes, a new generation is created, and the procedure is repeated until convergence. Since GA is a probabilistic method due to randomness insertion, it is important to run the problem multiple times when studying the characteristics.

The GA are now being widely used for engineering design problems [22, 77], presenting the ability to combine discrete, integer and continuous variables, no requirement for an initial design, and the ability to address non-convex, multi-modal and discontinuous functions. A GA was developed to optimize the rocket design, allowing an easy implementation and parallel optimization.

### 3.2.2 Particle Swarm Optimization

Particle Swarm Optimization algorithms model the social behavior of animal groups, such as flocking behavior of birds and the schooling behavior of fish. The population, or swarm, uses information obtained from each individual, or particle, as well as from the information obtained by the swarm as a whole, to reach the optimum solution. The general PSO flowchart is illustrated in figure 3.4.

The algorithm starts by generating an initial population, randomly distributed throughout the design space. At each iteration, the individual fitness is evaluated, using the specified objective function. If fit-

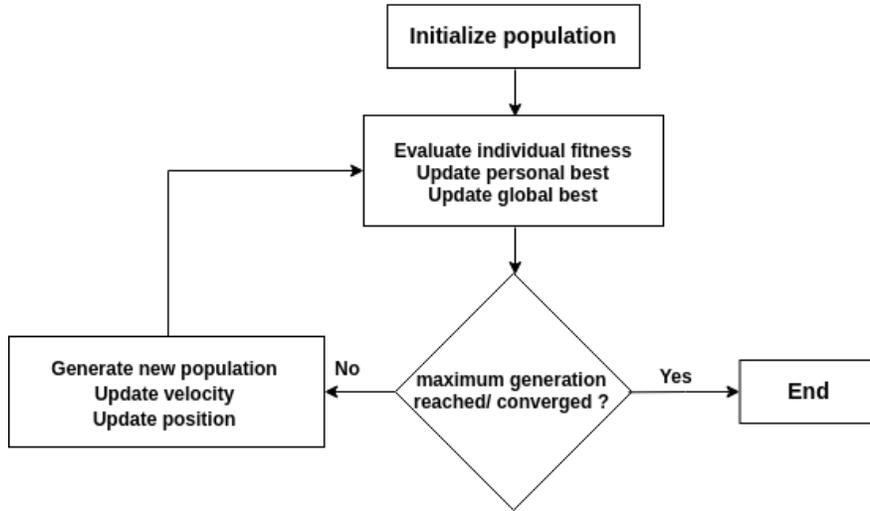


Figure 3.4: Particle swarm optimization procedure flowchart [70].

ness improves, the individual and global best positions are updated. The particles movement are guided by their own best known position, as well as the swarm's best known position. The position of each particle is then updated at every iteration, using

$$x_i^{q+1} = x_i^q + v_i^q \Delta t, \quad (3.1)$$

where  $x_i^q$  and  $v_i^q$  refers to the position and velocity vector of the  $i$ th individual at the  $q$ th iteration. The individual position is given as a point in the N-dimensional search space, where N is the number of variables, and the velocity vector directs the particle to its next position. The time increment  $\Delta t$  is typically taken as unity [70].

In the first iteration, a random velocity vector is assigned for each particle, which is updated at each iteration using

$$v_i^{q+1} = wv_i^q + c_1r_1 \frac{p_i - x_i^q}{\Delta t} + c_2r_2 \frac{p^g - x_i^q}{\Delta t}, \quad (3.2)$$

where  $w$  is the inertia parameter,  $c_1$  and  $c_2$  are the trust parameters (cognitive and social parameters respectively), and  $r_1$  and  $r_2$  are random numbers between 0 and 1. Additionally,  $p_i$  and  $p^g$  are the best position stored by the  $i$ th particle and by the swarm, respectively.

The  $c_1$  and  $c_2$  parameters indicates the degree of confidence in the best solution found by each individual particle and by the swarm as a whole, respectively. The inertia weight parameter  $w$  controls the search behavior of the algorithm by determining the contribution rate of a particle's previous velocity. Using larger inertia values facilitates a global search, while using small values eases local search [78]. Several methods allow the weight to be dynamically adjusted through the optimization [79].

The PSO algorithm was chosen to optimize the rocket trajectory, as it allows to accurately calculate the required trajectory path using indirect methods (section 3.3.2), while minimizing the propellant consumption

and the rocket total mass [27]. A precise trajectory calculation is important, as it allows to better analyze the external forces acting in the rocket, assuring the structural integrity.

### 3.3 Trajectory Optimal Control

The optimal rocket is obtained not only by improving the engines and reducing the structural mass, but also by optimizing the ascent trajectory. Rocket design and performance analysis are highly coupled to the trajectory optimization problem of finding the best path to a desired orbit [42]. The most important objectives for trajectory optimization are the maximization of transported payload, and minimization of the thrusting time taken to reach orbit, reducing fuel consumption [80].

Computing a multi-stage rocket trajectory is not trivial. The trajectory has to be divided into several phases to accommodate discontinuities and changes in the dynamic equations, due to the jettison of dry mass or the ignition of a motor. It also has to take into account boundary conditions, as to reach the desired orbit, and path constraints, to assure the well-function of the rocket (e.g. dynamic pressure, maximum load and buckling) [80].

After specifying the rocket characteristics, the optimization of its ascending trajectory consists of determining the optimal control law that achieves the objective at orbit injection. Historically, the numerical solution has been pursued using different methods and can be divided into indirect methods, where the necessary and sufficient conditions for optimality are analytically constructed and solved numerically, and indirect methods, where the optimization problem is converted into a Nonlinear Programming Problem (NLP) [24].

In general, direct methods are more robust, as they tend to converge to the desired result, even with a poor initial guess. However, although indirect methods are harder to initialize, the results are more accurate, which is critical for aerospace application. The need to find a good initial solution led to the development of effective heuristic methods in the last decades [81].

#### 3.3.1 Direct methods

The direct methods are the most used in the LV trajectory optimization, due to being easier to implement and initialize, comparatively to the indirect methods. In a direct method, the optimal control problem is transformed into a NLP [24].

There are several methods to transform an optimal control problem into an NLP, such as the use of direct shooting [82] and direct collocation methods [26]. In a direct shooting method, only the control is parameterized and numerical integration is used to propagate the trajectory in the different arcs. In direct collocation methods, the state and the control are parameterized, and piecewise polynomials are used to satisfy the differential equations at collocation points. Further information about direct methods can be found in [24, 83].

The use of direct methods present the advantage of easy implementation and the user does not have to be concerned with the deduction of the adjoint equations. However, it presents the disadvantage of producing less accurate solutions than indirect methods, which can be crucial in a space flight mission [84].

### 3.3.2 Indirect methods

Indirect methods use the theory of optimal control to transform the optimization problem into a Two Point Boundary Value Problem (TPBVP) by introducing adjoint variables, which is then solved numerically, usually using shooting methods. To turn the problem into a TPBVP, the differential equations for the adjoint variables, the control equation and the boundary conditions (transversality conditions) have to be analytically deduced and solved, in compliance with the Pontryagin's Minimum Principle (PMP) [85].

These equations are deduced from the Hamiltonian function [24, 84], given as

$$H = \lambda^T \mathbf{f} + L, \quad (3.3)$$

where  $\lambda^T \mathbf{f}$  is the adjoint variables conjugate to the state equations, and  $L$  is the Lagrangian of the system. The adjoint differential equations are deduced using

$$\frac{d\lambda}{dt} = - \left( \frac{\partial H}{\partial \mathbf{x}} \right)^T, \quad (3.4)$$

where  $x$  represent the state variables.

The optimal control is determined by minimizing the Hamiltonian with respect to the control variables  $u$ ,

$$\left( \frac{\partial H}{\partial \mathbf{u}} \right)^T = 0, \quad (3.5)$$

and by assuring the Legendre-Clebsch condition ( $\frac{\partial^2 H}{\partial \mathbf{u}^2}$  has to be positive semidefinite).

Finally, the transversality conditions can be deduced by solving

$$(\Phi_t + H)|_{t=t_f} = 0, \quad (3.6)$$

where  $\Phi$  is the boundary condition function, given by

$$\Phi = J + \nu^T \Psi(x_f), \quad (3.7)$$

being  $J$  the objective function, and  $\nu^T \Psi$  is the time-independent adjoint variable conjugate to the imposed boundary conditions.

The method tries to minimize the objective function, while complying with the optimality constraints. Common objectives are the time of flight and the propellant consumption.

The primary advantages of using indirect methods are their high solution accuracy, while assuring it

satisfies the first-order optimality conditions. However, it also present significant drawbacks, such as a small region of convergence, the requirement of deducting the analytical equations for optimality and initial guess of the adjoint variables [84]. To mitigate the drawbacks of the indirect methods, it is possible to use direct methods (such as direct collocation methods [64]), or conduct an heuristic search using stochastic algorithms to find a good initial guess for the adjoint variables [27], such as PSO.

The indirect method was chosen to control the rocket during the free-flight phase. Although is harder to initialize than direct methods, it provides more accurate results, critical for the rocket to complete the mission [64]. Thus, to ease the initialization, a PSO algorithm is used to search the optimal trajectory parameters.

## Chapter 4

# Optimal Rocket Design Procedure

As discussed in chapter 3, designing and optimizing a rocket requires the use of several disciplines. For preliminary design of rockets, to avoid expensive computation costs from high-fidelity models, simple models of the disciplines were used. The models were evaluated to check their feasibility.

The construction of the models integrated in the algorithm, together with the algorithm itself, are carefully explained in this chapter. The disciplines are divided in modules, that can be replaced by higher-fidelity models in the future, improving the accuracy of the solution.

### 4.1 Dry Mass Estimation and Sizing

The dry mass sizing module calculates the dry mass and the sizing of each stage integrated in the LV. Heuristic models based on historical data [86, 87, 88, 89] are used to estimate the dry mass and the size of the components. The relations are inferred from historical data.

Simple mass models tend to use a low number of parameters, so some may reach a relative error of 35% [90]. The dry mass model is dependent of the parameters chosen as design input. A simple model using the minimum number of parameters possible is developed in this section for solid stages and fairings, suitable for a preliminary rocket design.

#### 4.1.1 Solid Propulsion Systems

The majority of mass estimation works are focused mainly on liquid propulsion systems. However, the dry mass of a solid stage can be expressed as a linear function of the propellant mass  $m_p$  [88],

$$m_{SS} = 0.135m_p. \quad (4.1)$$

However, equation (4.1) presents low accuracy when calculating the mass of solid stages [90]. This

issue motivated the creation of a simple new heuristic model based on solid motors historical data in table A.1, using thrust and mass propellant as parameters. Several regression methods were tested and the multivariable linear regression presented the best fit, calculated as

$$m_{SM} = 0.313823T + 0.077471m_p, \quad (4.2)$$

where  $T$  is the engine thrust, in kN, and  $m_{SM}$  the solid motor mass, in kg.

Table 4.1 provides the significance of each variable used in the model. The p-value tests the null hypothesis that the variable has no correlation with the predicted variable, i.e., there is insufficient evidence to conclude that the variable is significant at the population level. Usually, the boundary value between high and low p-values is 0.05 [91]. A variable that has a low p-value is probably meaningful to the model, as changes in the variable because changes in the predictor's value are related to changes in the response of the model. On the other hand, a high p-value suggests that changes in the variable may not associated with changes in the response, indicating a strong null hypothesis.

Variable	P-value
Thrust	0.644998
Propellant Mass	0.004539

Table 4.1: P-value of solid stage mass regression variables.

The propellant mass variable has a low p-value, suggesting that it may be an important variable to the model. The engine thrust presents a high p-value, suggesting that it could be discarded from the model. However, both variables will be taken into consideration to increase fidelity.

The comparison between the real inert mass and the mass obtained from the model can be observed in figures 4.1 and 4.2.

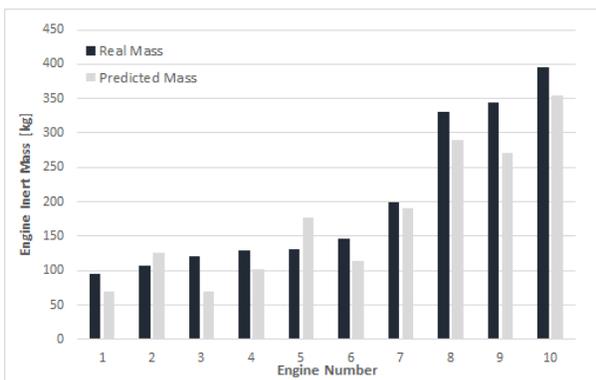


Figure 4.1: Comparison between real and predicted engine mass (under 500 kg).

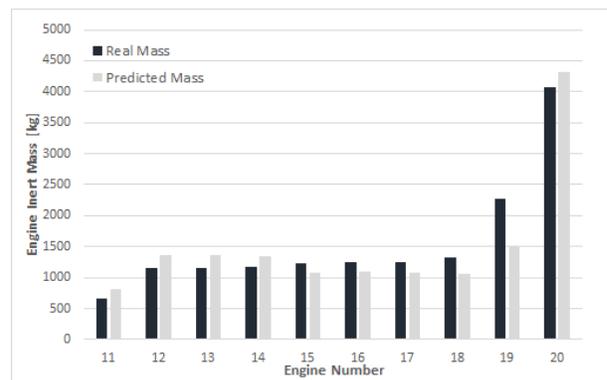


Figure 4.2: Comparison between real and predicted engine mass (above 500 kg).

Observing the adjusted R-square value in table 4.2, the high value presented states that the data points fit in the obtained regression. Observing figures 4.1 and 4.2, the majority of the predictions present an error

Multiple R	0.9853
R Square	0.9708
Adjusted R Square	0.913
Standard Error [kg]	231.70
Observations	20

Table 4.2: Statistics from engine dry mass regression.

below 20%, being suited to be used in a preliminary design.

To compare the mass predicted by the performed regression with equation (4.1), the stage outer-shell has to be taken into account, as equation (4.1) includes the shell mass. The inert stage mass is given as

$$m_{SS} = m_{SE} + \rho_{mat} \times L_{stage} \times \pi \times \left[ \frac{D_{stage}^2}{4} - \left( \frac{D_{stage}}{2} - th \right)^2 \right], \quad (4.3)$$

where  $\rho_{mat}$  is the material density,  $th$  is the thickness of the wall and  $L_{stage}$  and  $D_{stage}$  are the stage length and diameter, respectively. Using the solid stage mass, propellant mass and thrust data gathered in table A.2, it is possible to compare the predictions of both models. The thickness value of the outer wall is extremely difficult to find in literature. Hence, although is highly dependent on the material properties, the value of  $th = 4.7 \text{ mm}$  was used for the calculations, similar to the Falcon 9 wall thickness [8].

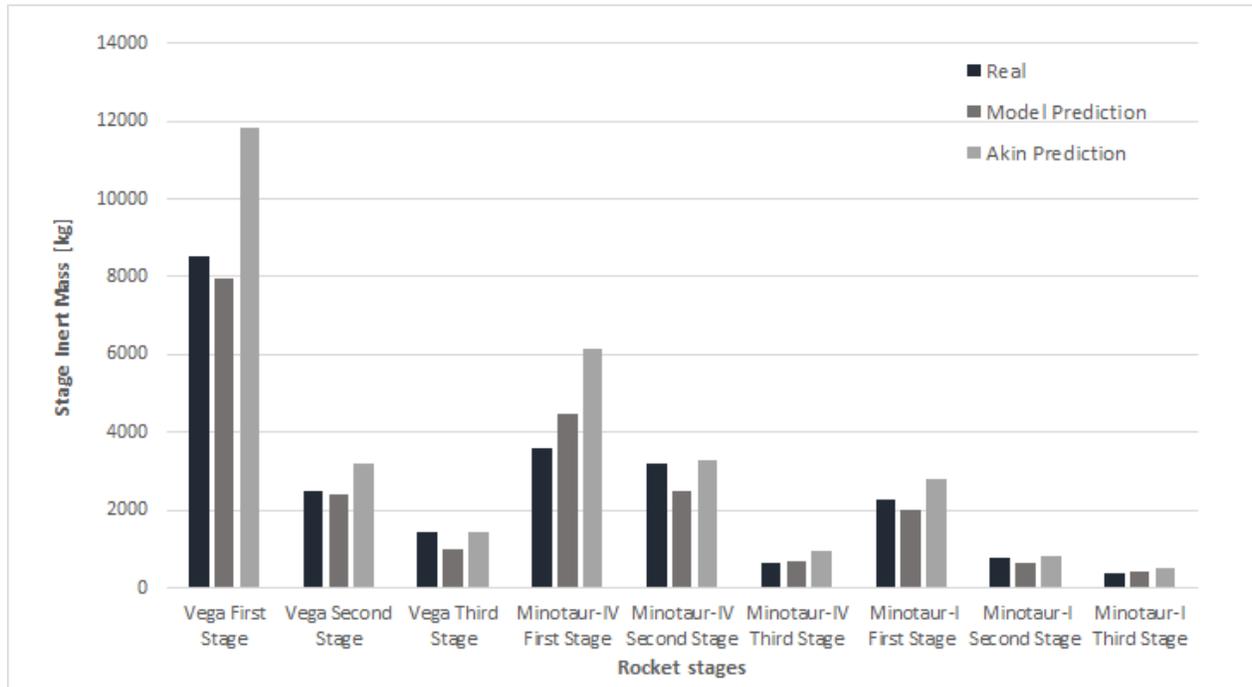


Figure 4.3: Comparison between real and predicted solid stage inert mass.

Observing figure 4.3, the regression model provides more accurate predictions than equation (4.1) in the majority of the cases studied. As such, the regression model will be used to calculate the inert mass of the solid core stages.

Using both formulas to calculate boosters inert mass, in figure 4.4, equation (4.1) provides better predictions than the regression model, therefore it will be used to calculate the boosters inert mass during optimization process.

Comparing the structural ratio of the different solid stages, the actual tendency is to use the regression model for structural ratio below 0.1, and to use equation (4.1) for structural ratio above 0.1. However, for simplification, the decision for choosing between formulas will be solely due to the type of rocket element (i.e. core stage or booster).

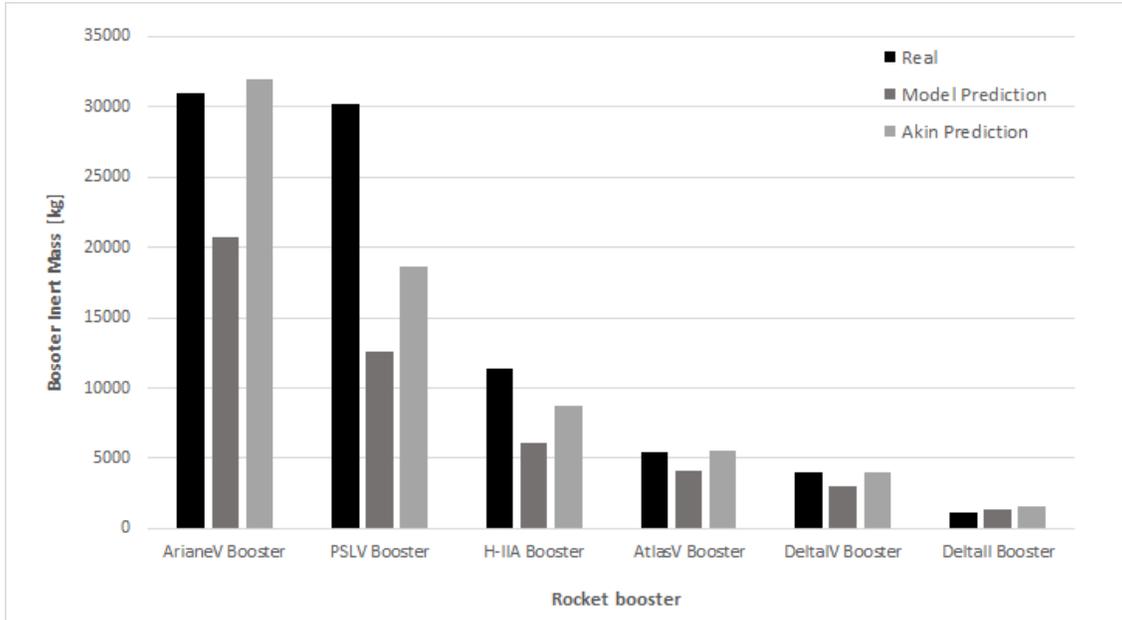


Figure 4.4: Comparison between real and predicted solid booster inert mass.

The solid stage length is calculated using the stages and boosters length data in table A.2 to perform a regression. The predictor used was the solid propellant grain length, calculated as

$$L_{grain} = \frac{m_p}{\left(\frac{D_{stage}}{2}\right)^2 \times \pi \times \rho_p}, \quad (4.4)$$

where  $\rho_p$  is the propellant density. The propellant is assumed to be HTPB ( $\rho_{HTPB} \approx 1800 \text{ kg/m}^3$ ). The correlation between the propellant grain length and the stage length can be observed in figure 4.5.

According to figure 4.5, the correlation can be represented by a linear regression. The high R-square value indicates that the linear model fits the data well. The regression expression is

$$L_{SolidStage} = 1.6495 \times L_{grain}. \quad (4.5)$$

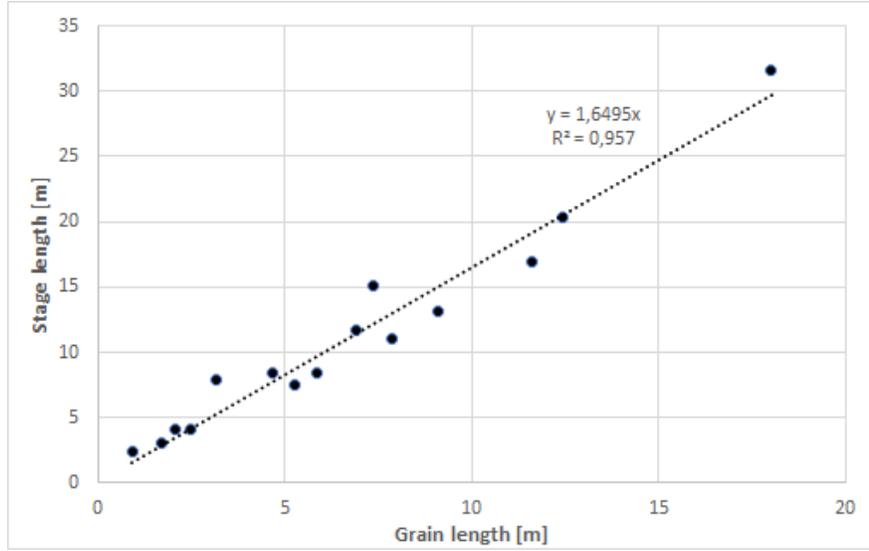


Figure 4.5: Fit between stage length and propellant grain length.

#### 4.1.2 Liquid Propulsion Systems

A bi-propellant liquid rocket engine is composed by two tanks to store the fuel and the oxidizer, a thrust chamber, a structure to support the propulsive engine and other components to connect all the elements. In the scope of this work, the pump-pressure fed system was chosen as it is the main system used for high-thrust applications. The mass and sizing models for liquid stages can be found in [90]. It uses empirical equations from liquid engines historical data found in literature and has an average error of 23% for the engine dry mass and 16% for the engine length, relative to the engines tested in [90].

The total engine mass is the sum of the system components as

$$m_{LE} = m_{tc} + m_{tankO} + m_{tankF} + m_{st}, \quad (4.6)$$

where  $m_{tc}$ ,  $m_{tankO}$ ,  $m_{tankF}$  and  $m_{st}$  are the masses of the the thrust chamber, oxidizer tank, fuel tank, and support structure, respectively.

The thrust chamber is composed by the propellant injectors, igniter, a combustion chamber, an exhaust nozzle and a cooling system. The mass can be estimated by [86]

$$m_{tc} = \frac{T}{g_0(25.2 \log(T) - 80.7)}. \quad (4.7)$$

To calculate the support structure mass, an empirical equation is used [92],

$$m_{st} = 0.88 \times 10^{-3} \times (0.225T)^{1.0687}. \quad (4.8)$$

Regarding the tanks, the mass was estimated using physics-based equations, assuming cylindrical tanks

with semi-spherical ends. The mass of the tank is calculated by

$$m_{tank} = (A_c \times th_c + A_s \times th_s) \times \rho_{mat}, \quad (4.9)$$

where  $A_c$  and  $A_s$  are the surface area of the cylindrical and spherical sections, respectively, and  $th_c, th_s$  the wall thickness. The wall thickness of the cylindrical and spherical section are defined, respectively, by

$$th_c = \frac{0.5P_b D_{tank}}{F_{tu}} \quad (4.10)$$

and

$$th_s = \frac{0.25P_b D_{tank}}{F_{tu}}, \quad (4.11)$$

with  $F_{tu}$  the allowable material strength,  $P_b$  the burst pressure, and  $D_{tank}$  the tank diameter, assumed to be equal to the stage diameter. The burst pressure is calculated from [90]

$$P_b = \eta_s \lambda_b P_{tank} = \eta_s \lambda_b (10^{-0.10688(\log(V_{tank}) - 0.2588)}) \times 10^6, \quad (4.12)$$

with a safety factor  $\eta_s$  equal to 2, and a ratio between the maximum expected operating pressure and the tank pressure  $\lambda_b$  equal to 1.2, as recommended in [90] to guarantee the tank structural integrity. The tank pressure approximation provided by [86] was used to reduce the model variables. The volume of each tank is determined by

$$V_{tank} = \frac{m_{O/F}}{\rho_{O/F}}, \quad (4.13)$$

where  $M_{O/F}$  and  $\rho_{O/F}$  are the mass and density of the oxidizer/fuel, respectively. The propellant density and common oxidizer/fuel ratios are listed in table 4.3.

Propellant	Oxidizer Density [kg/m <sup>3</sup> ]	Fuel Density [kg/m <sup>3</sup> ]	O/F Ratio
N2O4/MMH	1450	880	2.16
N2O4/UDMH	1450	793	2.61
LOX/RP-1	1140	806	2.56
LOX/LH2	1140	71	6

Table 4.3: Liquid propellant characteristics.

The liquid stage mass is calculated by accounting the outer-shell mass, using

$$m_{LS} = m_{LE} + \rho_{mat} \times L_{stage} \times \pi \times \left[ \frac{D_{stage}^2}{4} - \left( \frac{D_{stage}}{2} - th \right)^2 \right]. \quad (4.14)$$

To calculate the length of the liquid stage, it is accounted as the sum of the tanks and thrust chamber lengths,

$$L_{LS} = L_{tc} + L_{tankO} + L_{tankF}, \quad (4.15)$$

where  $L_{tc}$  is the thrust chamber length and  $L_{tankO}, L_{tankF}$  are the oxidizer and fuel tank lengths, respec-

tively. The thrust chamber length is calculated using [86]

$$L_{tc} = 3.042 \times 10^{-5}T + 327.7. \quad (4.16)$$

The tank length is determined by dividing it into the length of the cylindrical section and the length of the spherical ends. The length of the spherical section is equal to the diameter of the tank  $D_{tank}$ . The length of the cylindrical section is deduced by removing the volume of the spherical section from the tank volume  $V_{tank}$  (equation (4.13)). The total tank length is therefore given by

$$L_{tank} = D_{tank} + \frac{V_{tank} - \frac{4}{3}\pi\left(\frac{D_{tank}}{2}\right)^3}{\pi\left(\frac{D_{tank}}{2}\right)^2}. \quad (4.17)$$

However, if the propellant is able to be stored in a spherical tank, the diameter and length of the tank are

$$L_{tank} = D_{tank} = 2 \times \left(\frac{V_{tank}}{\frac{4}{3}\pi}\right)^{-\frac{1}{3}}. \quad (4.18)$$

### 4.1.3 Payload Adapter and Fairing

The payload adapter mass ( $m_{PLA}$ ) regression can be found in [87], where a power regression best fits the data, constituted of 10 adapters from Ariane 5, VEGA, Soyuz and Atlas families. It is given by

$$m_{PLA} = 0.0477536 \times (m_{PL})^{1.01317}, \quad (4.19)$$

where  $m_{PL}$  is the payload mass.

The fairing length equation is developed by performing a linear regression with the fairing data collected in table A.3. The length is estimated using the maximum payload to LEO for each rocket family and the fairing diameter, considered to be equal to the last stage diameter if not specified otherwise. It is given by

$$L_{PLF} = 3.939 \times 10^{-4}m_{PL} + 1.924244D_{PLF}, \quad (4.20)$$

where  $D_{PLF}$  is the fairing base diameter. Table 4.4 shows the significance level of the predictive variables.

Variable	P-value
Fairing Diameter	0.000379
Payload Mass	0.001789

Table 4.4: P-value of fairing length regression variables.

Observing both variables, the p-value is below 0.05. Both variables are assumed to be statistically significant and must remain in the model.

The high adjusted R squared in table 4.5 indicates that the data fits the regression in equation (4.20), as observed in figure 4.6. After predicting the fairing length, the payload fairing mass can be calculated. Once

Multiple R	0.9932
R Square	0.9864
Adjusted R Square	0.8597
Standard Error [m]	1.7960
Observations	10

Table 4.5: Statistics from fairing length regression.

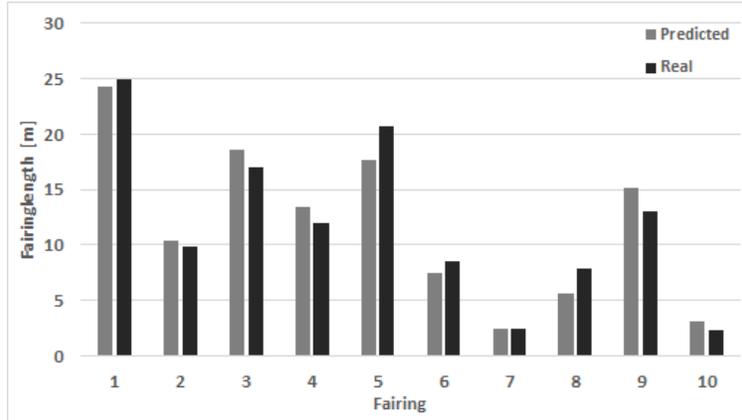


Figure 4.6: Comparison between real and predicted fairing length.

again, an empirical equation is deduced by performing a linear regression using the fairing diameter and length data in table A.3, obtaining

$$M_{PLF} = 31.0721D_{PLF} \times L_{PLF}. \quad (4.21)$$

Variable	P-value
Fairing Diameter $\times$ Fairing Length	1.836e-09

Table 4.6: Variable p-value of fairing mass regression.

For the regression, only the variable  $D_{PLF} \times L_{PLF}$  is considered, as the fairing mass is directly proportional to the fairing surface area. The significance level of the variable is indicated in table 4.6. The p-value is extremely low, meaning the variable is of major importance to the model.

Multiple R	0.9922
R Square	0.9846
Adjusted R Square	0.8734
Standard Error [kg]	288.7176
Observations	10

Table 4.7: Statistics from fairing mass regression.

The high adjusted R square value in table 4.7 indicates a good data fitting. Both the regressions obtained present a slight standard error. However, as shown in figures 4.6 and 4.7, the estimated values are relatively close to the observed values, so the formulas can be used to estimate the fairing dimensions and mass.

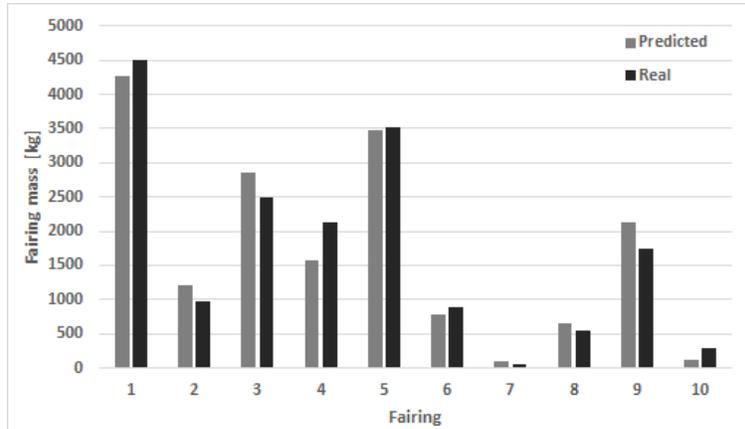


Figure 4.7: Comparison between real and predicted fairing mass.

## 4.2 Trajectory Model

Launch vehicle design and performance analysis are highly coupled to the trajectory analysis problem of finding the optimal path for minimum lift-off weight, while fulfilling the imposed flight constraints and inserting the payload into the desired orbit [42].

During staging, the algorithm needs to take into account the discontinuities due to the inert mass drop and thrust changes. The ideal trajectory profile is based on reaching the final velocity, altitude and orientation as the upper stage completes its injection burn, while minimizing velocity losses due to the drag, gravity and thrust vectoring.

The trajectory path is designed to avoid any aerodynamic side load, which could break the rocket. This is the main reason why the angle of attack is kept at zero while the rocket flies through the atmosphere using the gravity turn maneuver. Although is not optimal, it is widely used in the trajectory of several rocket design efforts [46, 44].

For this work, in order to simplify the optimization problem, the trajectory was divided in different flight phases, as stated in section 2.4, comprising of vertical ascent, pitch over, gravity turn and free flight phase. The pitch over maneuver will simply be represented as a small discontinuity step in the flight path angle, enough to properly start the gravity turn. After the gravity turn, the trajectory optimization is solved by defining an Hamiltonian function and applying the Pontryagin's Minimum Principle [27]. A PSO algorithm is then implemented to explore the search space for the optimal solution of the problem [27].

### 4.2.1 Equations of Motion

To completely describe a launch vehicle dynamics, the translational and rotational motion need to be accounted. To simplify the equations of motion, the rocket is assumed to be a variable-mass rigid body flying in a 2-D plane model, as illustrated in figure 4.8. To calculate the trajectory path, the rotational motion

of the vehicle can be disregarded [30]. Hence, the forces acting in the rocket are applied at the center of mass during flight and the equations of motion can be described by Newton's law  $F = ma$ .

The rocket's active stage produces a thrust  $T$  with an angle  $\chi$  in respect to the velocity vector  $V$ . In the context of trajectory analysis and optimization, the thrust direction can be assumed as always aligned to the vehicle longitudinal axis ( $\chi \equiv \alpha$ ) [27]. The aerodynamic drag force  $D$  has the opposite direction of the velocity vector, while the lift force  $L$  is directed perpendicularly. The force of gravity applied on the vehicle is  $mg$ , where  $m$  is the vehicle mass and  $g$  the local gravitational acceleration, which points to the center of the Earth at all times.

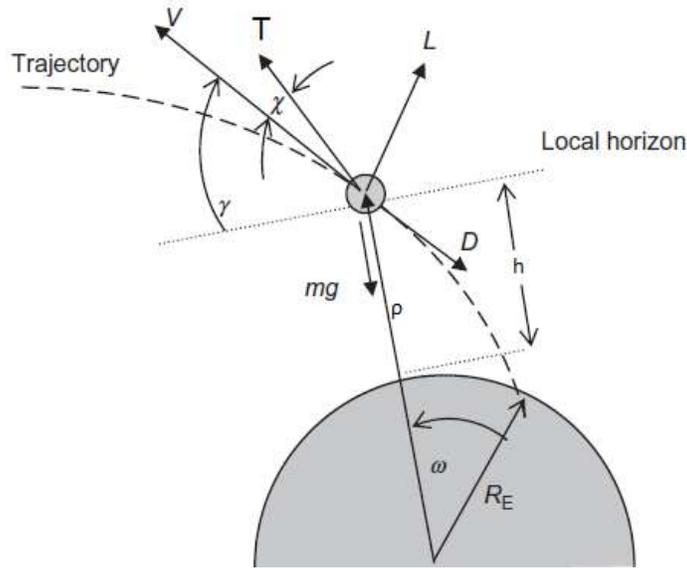


Figure 4.8: Rocket state variables and forces during flight [34].

The lift force is neglected, as it is held closely to zero during the powered ascent through the atmosphere [93]. The Coriolis and centripetal acceleration due to the Earth rotation are also neglected during trajectory simulation, as the accelerations are small when compared to the acceleration due to aerodynamic forces [94]. The contribution due to Earth rotation is taken into consideration before the start of the simulation. A reference system that rotates with the Earth and has the origin in its center, is used to better describe the rocket motion [27].

The forces acting in the vehicle can be divided into forces tangent and normal to the trajectory. The normal direction points toward the center of the trajectory curvature at all times.

The equations of motion are carefully deduced in [34]. For a flat earth model, the normal acceleration is

$$a_n = -V \frac{d\gamma}{dt}. \quad (4.22)$$

To account for the curvature of the Earth, polar coordinates with origin at the Earth center are used, correct-

ing the normal acceleration to

$$a_n = -V \frac{d\gamma}{dt} + \frac{V^2}{R_e + h} \cos \gamma, \quad (4.23)$$

where  $R_e = 6.371 \times 10^6$  m is the radius of the earth and  $h$  is the vehicle altitude. The tangential acceleration, along the trajectory path, is given as

$$a_t = \frac{dV}{dt}. \quad (4.24)$$

The equations of motion for the tangential and normal direction, respectively, are

$$\dot{V} = \frac{T}{m} \cos \alpha - \frac{D}{m} - g \sin \gamma \quad (4.25)$$

and

$$V\dot{\gamma} = -\left(g - \frac{V^2}{R_e + h}\right) \cos \gamma + \frac{T}{m} \sin \alpha. \quad (4.26)$$

The equations for downrange distance  $x$  and altitude  $h$  are

$$\dot{x} = \frac{R_e}{R_e + h} V \cos \gamma \quad (4.27)$$

and

$$\dot{h} = V \sin \gamma. \quad (4.28)$$

During trajectory simulation, the local gravity acceleration is considered, being updated according to the vehicle altitude. To solve the equations of motion, numerical methods must be implemented, as the vehicle mass, thrust, drag and gravitational acceleration vary in time.

It is important to notice that the equations stand for the entire flight duration of the vehicle. However, thrust deflection is only active after the end of gravity turn, during the free flight phase.

## 4.2.2 Free Flight Phase

As the vehicle leaves the tangible atmosphere, the thrust does not need to be aligned with the velocity vector anymore since there will be no considerable aerodynamic loads applied. A better trajectory can be found by throttling and/or deflecting the thrust, ending when the vehicle reaches orbit, ideally using all the available propellant. The optimal trajectory proposed is an extension of the work performed in [27]. For the explanation of the proposed method, a coast phase is assumed as they are typically required for a great variety of missions [42].

The free flight phase can be treated as a TPBVP, in which the vehicle initial position corresponds to the end of gravity turn, and the final position to the insertion in the specified orbit. The optimal controls for the optimized trajectory are found using indirect methods, by applying the Pontryagin's Minimum Principle [30]. The boundary value problem is then solved by using a shooting method to find the Lagrangian multipliers [24]. Using a shooting method has the advantage that all kinds of constraints are allowed and very accurate

results can be obtained.

The problem objective is to reduce the propellant consumption, which is equivalent to minimize the thrusting time. The proposed objective function is to reduce the final impulse time to reach circular orbit, expressed as

$$J = t_f - t_{cf}, \quad (4.29)$$

where  $t_f$  and  $t_{cf}$  are the final flight time and the final coast time, respectively. If the coast phase is not accounted, the parameter  $t_{cf}$  can be replaced by the last stage initial thrusting time, transforming the problem objective function into the minimization of the last stage thrust time.

The initial states value of the TPBVP  $(t_0, x_0, h_0, V_0, \gamma_0)$  are provided by the end of the gravity turn. For reference, the subscripts "0" and "f" will be used to refer to the initial and final recorded variable value and, when specified, of the corresponding flight arc. The flight arcs are divided by the discontinuities in mass and thrust through the flight. The terminal boundary constraints are

$$\Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{bmatrix} = \begin{bmatrix} h_f - h' \\ V_f - V' \\ \gamma_f - \gamma' \end{bmatrix} = 0, \quad (4.30)$$

where  $h'$ ,  $V'$ , and  $\gamma'$  are the final state values, for the vehicle to reach the desired orbit. The final time  $t_f$  and final downrange  $x_f$  are unknown.

The Earth rotation is taken into account by summing the required velocity at the end of the flight [94]. The rotation effect in the rocket velocity for an equatorial orbit, considering a spherical Earth, is given by

$$V_{ER} = \frac{2 \times 6.371 \times 10^6 \times \pi}{86164} \times \cos \delta, \quad (4.31)$$

where  $\delta$  is the latitude of the launch site. If the rocket is launched at the equator, the rotation contribution would be 465m/s. The contribution velocity vector is always perpendicular to the direction of the rocket and Earth center of mass. Performing a vector sum, the final velocity and flight path angle due to the rotation are respectively given by

$$\gamma'' = \arctan \frac{V' \sin \gamma'}{V' \cos \gamma' - V_{ER}} \quad (4.32)$$

and

$$V'' = \sqrt{(V' \cos \gamma' - V_{ER})^2 + (V' \sin \gamma')^2}, \quad (4.33)$$

where  $V''$  and  $\gamma''$  are the desired rocket final velocity and flight path angle.

The terminal boundary constraints have to be updated with the new values to account with Earth's rotation. Otherwise, the terminal values remain unchanged, for a non-rotating trajectory model.

The Hamiltonian for each flight arc is set as

$$H = L + \boldsymbol{\lambda}^T \mathbf{f} = \lambda_x \dot{x} + \lambda_h \dot{h} + \lambda_V \dot{V} + \lambda_\gamma \dot{\gamma}, \quad (4.34)$$

and the boundary condition function as

$$\Phi = J + \boldsymbol{\nu}^T \boldsymbol{\Psi}(x_f) = (t_f - t_{cf}) + \boldsymbol{\nu}^T \boldsymbol{\Psi}(x_f), \quad (4.35)$$

where  $\boldsymbol{\lambda}$  is the adjoint or costate variable conjugate to the state equations and  $\boldsymbol{\nu}$  is the time-independent adjoint variable conjugate to the boundary conditions. Due to the Weierstrass-Erdmann corner conditions [27], the adjoint variables are continuous across successive flight arcs.

The necessary conditions for optimality (Euler-Lagrange equations) are carefully explained in [95]. To minimize the Hamiltonian, the conditions (4.36) need to be satisfied.

$$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = 0, \quad (4.36a)$$

$$\dot{\lambda}_h = -\frac{\partial H}{\partial h} = \frac{1}{(R_E + h)^2} V \lambda_\gamma \cos(\gamma) - \left( 2\mu_E \lambda_V \sin \gamma + \frac{2\mu_E \lambda_\gamma \cos \gamma}{V} \right) \frac{1}{(R_E + h)^3}, \quad (4.36b)$$

$$\dot{\lambda}_V = -\frac{\partial H}{\partial V} = -\lambda_h \sin \gamma - \lambda_\gamma \left[ \cos \gamma \left( \frac{1}{R_E + h} + \frac{\mu_E}{(R_E + h)^2 V^2} \right) - \frac{T}{m} \frac{1}{V^2} \sin \alpha \right], \quad (4.36c)$$

$$\dot{\lambda}_\gamma = -\frac{\partial H}{\partial \gamma} = -V \lambda_h \cos \gamma + \mu_E \lambda_V \frac{\cos \gamma}{(R_E + h)^2} + \lambda_\gamma \sin \gamma \left( \frac{V}{(R_E + h)} - \frac{\mu_E}{(R_E + h)^2 V} \right). \quad (4.36d)$$

For the optimization problem, the control variable used is the thrust deflection  $\chi$ , which is equal to the angle of attack  $\alpha$ , as stated above. Another possible control variable would be the thrust throttling, but due to the difficult of implementing it in the rocket design [96], it will not be taken into consideration.

The control equation can be written in terms of the adjoint and state variables through the Pontryagin's Minimum Principle,

$$\alpha = \arg \min_{\alpha} H, \quad (4.37)$$

which is the equivalent to solve

$$\frac{\lambda_\gamma}{V} \sin \alpha + \lambda_V \cos \alpha = 0, \quad (4.38)$$

with  $\sin \alpha = -\frac{\lambda_\gamma}{V} \left[ \left( \frac{\lambda_\gamma}{V} \right)^2 + \lambda_V^2 \right]^{-\frac{1}{2}}$  and  $\cos \alpha = -\lambda_V \left[ \left( \frac{\lambda_\gamma}{V} \right)^2 + \lambda_V^2 \right]^{-\frac{1}{2}}$  in order to verify Pontryagin's Minimum Principle, which states that the Hamiltonian must be minimized with respect to the control ( $\frac{\partial^2 H}{\partial \alpha^2} \geq 0$ ).

The coast time and the burn time of the last stage are unspecified. According to [27], the transversality condition is given by

$$H_f^{\text{last stage}} + H_f^{\text{coast}} - H_0^{\text{last stage}} = 0 \quad \text{and} \quad H_f^{\text{last stage}} < 0. \quad (4.39)$$

Recognizing the costate equations (equation (4.36)) as homogeneous in  $\lambda$ , the optimization algorithm is capable of finding the initial value of  $\lambda$  such that  $\lambda = k_\lambda \lambda^*$  ( $k_\lambda > 0$ ), where the superscript "\*" denotes the actual optimal value of the adjoint variable. The same proportionality holds between  $\lambda$  and  $\lambda^*$  at any time  $t$ , meaning the costate initial values can be sought in the interval  $-1 \leq \lambda_k \leq 1$ , reducing the search space.

To deal with the optimization constraints, a popular approach consists in using the penalty function method to penalize the objective function, transforming the constrained problem into an unconstrained one. The use of the penalty function allows to build a single objective function, able to be minimized by the PSO algorithm. The new objective function is expressed as

$$J' = J + \sum_{c=1}^3 s_c \|\mathbf{x}_{c,f} - \mathbf{x}'_c\| + s_4 \|H_f^{\text{last stage}} + H_f^{\text{coast}} - H_0^{\text{last stage}}\|, \quad (4.40)$$

where  $s_c$  denotes the constraint penalty factor,  $\mathbf{x}_f$  the final state vector and  $\mathbf{x}'$  the required state vector. The original objective function  $J$  is penalized if the rocket does not reach the required altitude, velocity or flight path angle and if the transversality condition is not verified.

### 4.3 Atmospheric Model

The atmospheric properties vary with the altitude and affect the vehicle performance. The atmospheric temperature, density and pressure are modelled by interpolating the data provided in the 1976 U.S. Standard atmosphere, which covers altitudes up to 1000 km [97].

During atmospheric flight, the flow regimes vary from continuum to free-molecular flow. The flow regime is determined by the Knudsen number, which helps to determine whether statistical mechanics or the continuum mechanics formulation of fluid dynamics should be used to compute the aerodynamic forces.

The Knudsen number is defined as

$$Kn = \frac{\lambda}{L}, \quad (4.41)$$

where  $\lambda$  is the molecular mean free path length, and  $L$  is the characteristic length, typically the nose radius of the last stage. Using the Knudsen number, the flow regimes can be classified as [30]:

$$\left\{ \begin{array}{ll} \text{Continuum flow,} & \text{if } Kn \leq 0.01, \\ \text{Transitional flow,} & \text{if } 0.01 < Kn < 10, \\ \text{Free-molecular flow,} & \text{if } 10 \leq Kn. \end{array} \right. \quad (4.42)$$

The tangible atmosphere ends when the vehicle enters the free-molecular regime, which occurs near the Kármán line ( $h \approx 100$  km). At this point, the vehicle performance is no longer affected by Earth atmosphere. Thus, the shroud protecting the payload can be dropped, reducing dead mass.

## 4.4 Drag Model

At low altitudes, the vehicle experiences a significant atmospheric drag while in the tangible atmosphere. Drag is given as function of the vehicle flight speed  $V$ , the mass density  $\rho$  and a characteristic surface area  $S$ , as

$$D = C_D \frac{1}{2} \rho S V^2, \quad (4.43)$$

where  $C_D$  is the drag coefficient. The drag coefficient is a function of the flight Mach number  $M$ , angle of attack  $\alpha$ , Knudsen number  $Kn$  and of the nose cone geometry.

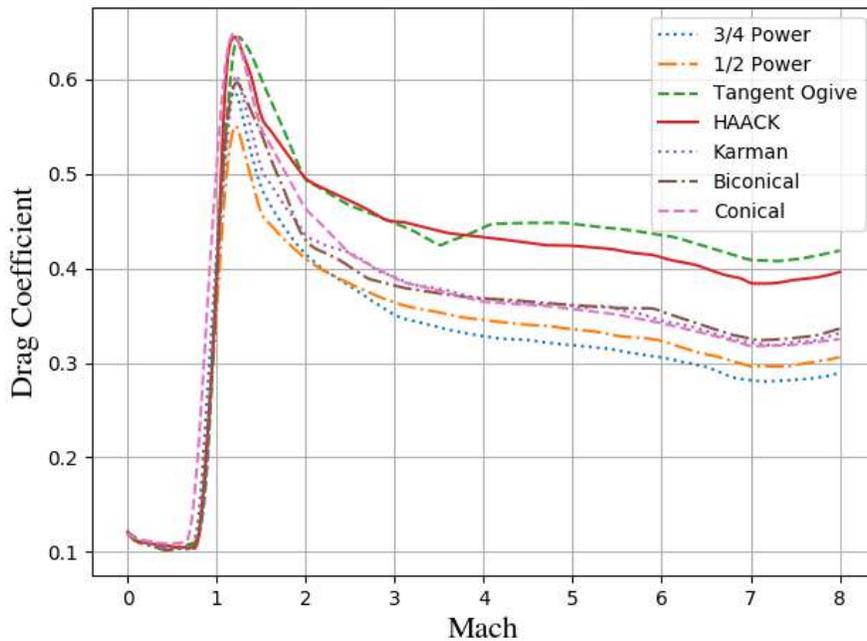


Figure 4.9: Influence of Mach number in Drag coefficient [98].

During the atmospheric flight, the attitude control system always maintains a zero angle of attack to minimize the transverse loads. The drag coefficient  $C_D$  is approximated as a function only of Mach number and Knudsen Number. For different geometries, the data was obtained from [98], using Missile DATCOM software to estimate  $C_D$ . By the hypersonic independence limit,  $C_D$  becomes almost invariant with  $M$  after a critical Mach number, generally between 4 and 6 [99], as observed in figure 4.9.

If not specified otherwise, the chosen default geometry is the 1/2 Power, illustrated in figure 4.10, due to presenting the lowest drag coefficient near the speed of sound. However, to use the data from figure 4.9, corrections need to be made. The DATCOM software does not use the Knudsen number in the estimates, as it only considers continuum flow, so it needs to be corrected [95] using

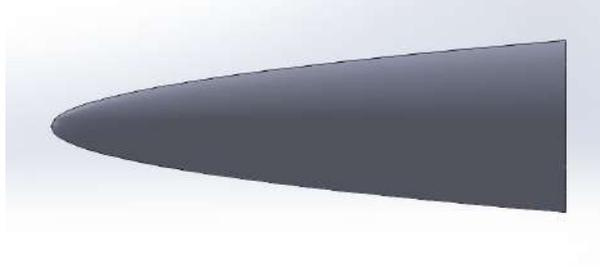


Figure 4.10: Render of the half power series nose cone.

$$C_D = \begin{cases} C_{Dc}(M), & \text{if } Kn < Kn_c, \\ C_{Dfm}, & \text{if } Kn > Kn_f, \\ C_{Dc} + (C_{Dfm} - C_{Dc}) \left[ \frac{1}{3} \log_{10}(A Kn) + B \right], & \text{if } Kn_c < Kn < Kn_f, \end{cases} \quad (4.44)$$

where  $C_{Dc}(M)$  is the drag coefficient in the continuum flow limit, function of Mach number and the nose geometry, and  $C_{Dfm}$  is the drag coefficient in the free molecular limit, where  $Kn > Kn_f$ , and is given by the cold-wall approximation [100]

$$C_{Dfm} = 1.75 + \frac{\sqrt{\pi}}{2s}, \quad (4.45)$$

where  $s = \frac{V}{\sqrt{2RT}}$  denotes the molecular speed ratio, with  $T$  being the air temperature and  $R = 287.058$  J/kgK being the air specific gas constant, The Knudsen numbers  $Kn_c$  and  $Kn_f$  represent the Knudsen limits of continuum flow and free-molecular flow, respectively, which depend exclusively on the vehicle geometry. The constants  $(A, B)$  are selected to perform a smooth bridging between the continuum and free-molecular flow regimes.

One important parameter to estimate is the dynamic pressure,

$$q = \frac{1}{2} \rho V^2. \quad (4.46)$$

Rocket trajectories are usually constrained by the maximum dynamic pressure, which is when the atmospheric forces are maximum, risking the vehicle structural integrity. The maximum value occurs at altitudes around 10 km. The vehicle acceleration needs to be carefully controlled [95], normally by throttling down the engines or by carefully choosing the solid grain shape [38].

## 4.5 Algorithm Development

The rocket design and optimization process was done by developing an optimization algorithm. Traditional methods are based on exhaustive searches or gradient-based algorithms to optimize the continuous design variables. However, discrete variables have usually to be assumed. Thus, the systems are optimized

assuming the fixed value and the optimization is then repeated assuming a different discrete parameter value, which can be time-consuming [77]. The GA was chosen to optimize the rocket design, allowing an easy implementation. The final objective of the optimization is to minimize the rocket launch mass, reducing manufacturing and operative costs.

The main part of the algorithm was developed in Python, while the trajectory model was implemented using the C language, to enhance speed. The code was developed using native libraries, with the exception of the PSO algorithm, which was taken from python library PyGMO [101]. PyGMO (Parallel Global Multi-objective Optimizer) is a scientific library for massively parallel optimization, used at the European Space Agency in the context of interplanetary trajectory optimization and orbit determination.

To initialize the algorithm, the user has to specify the mission parameters, initial conditions and the fixed design variables. A list of the parameters that need to be specified is presented in table 4.8.

All parameters presented in table 4.8 can be specified by the user if, for instance, the user wants to perform trajectory simulation/optimization using real rocket data. However, for optimization, most parameters can be handled by the optimizer.

Mission Parameters	Design Parameters	
Payload Mass	Number of Stages	<b>Propulsion</b>
Initial Altitude ( $h_0$ )	Number of Boosters	Oxidizer/Fuel Ratio
Initial Velocity ( $V_0$ )	*** Fairing Mass	Propellant Density
Initial Flight Path Angle ( $\gamma_0$ )	Nose Geometry	Nozzle Area
Final Altitude ( $h_f$ )		Specific Impulse
Final Velocity ( $V_f$ )	<b>Stage Relative</b>	* Number of Engines
Final Flight Path Angle ( $\gamma_f$ )	Stage Type (Solid/Liquid)	* Engine Thrust (Vacuum)
	* Stage Thickness	
<b>Trajectory Parameters</b>	* Stage Diameter	
** Pitch Angle (vertical launch)	*** Stage Length	
Pitch Maneuver Initialization (vertical launch)	*** Total Mass	
** Coast Initialization	*** Propellant Mass	
** Coast Duration	*/*** Change in Velocity ( $\Delta V$ )	
** Last Stage Burn Duration		
* - Can be specified by the Genetic Algorithm optimizer ** - Can be specified by the Particle Swarm Optimization algorithm *** - Can be calculated using heuristics or physical equations		

Table 4.8: List of parameters handled by the algorithm.

#### 4.5.1 Genetic Algorithm Implementation

A continuous genetic algorithm was built to handle the optimization process. The GA performs four actions: evaluation, selection, crossover, and mutation. These actions are critical to optimize the design at each generation.

To accelerate the optimization process, the GA was built with parallelization option, by using the Python library MPI4Py. The MPI4Py (Message Passing Interface) allows Python programs to exploit multiple proces-

sors, by supporting point-to-point (sends, receives) and collective (broadcasts, scatters, gathers) communications. For parallelization, a master-slave architecture was used. The master node scatters the population individuals throughout the slave nodes, which perform the individual evaluation to assess the fitness. The implementation of the GA algorithm using a master-slave architecture is illustrated in figure 4.11.

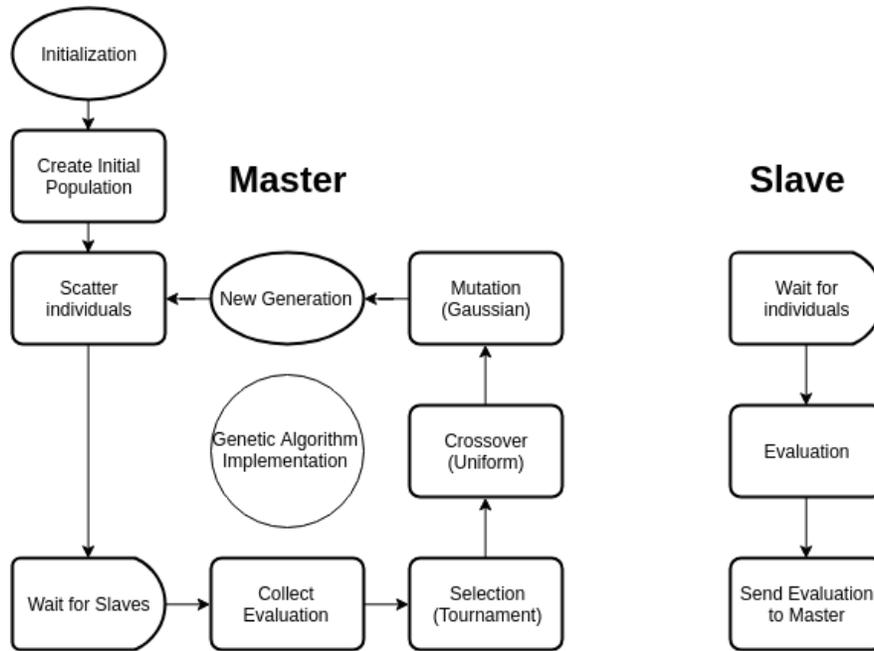


Figure 4.11: Genetic Algorithm implementation using master-slave architecture.

After choosing the parameters to optimize in table 4.8, the algorithm proceeds to initialize the population with the defined design parameters. The master node then scatters the parameters through all the slave nodes defined, which can be from one (only the master node performs evaluation) to the number of individuals in the population. In this context, each node corresponds to a CPU core. The slaves then perform the rocket mass evaluation by calculating the mass and the trajectory (section 4.5.2). Afterwards, the fitness values are sent to the master node to create the new generation.

The population initialization is performed using a latin hypercube method. The latin hypercube design is one of the most used methods to fill the space design and using this method allows the population to cover most part of the design region, which may not happen when using random sampling [102]. The maximin criteria was chosen for the sampling, maximizing the smallest distance between any two design points, spreading them evenly over the entire design region.

To define the new generation children, the parents need to be selected from the actual generation, through the selection process. For this algorithm, a tournament selection was chosen, consisting on randomly extracting a limited number of individuals to compete, where the best individual survives and is selected to be a parent. Succeeding the selection, an uniform crossover may be performed, with both parents having equal probability to provide their genetic information for each gene of the offspring. Thus, the children

created do not share the same genetic information, i.e., a parent only provides the gene to one children. Afterwards, a Gaussian mutation changes the children genes, by adding a centered normally distributed noise. For this optimization the chosen crossover rate and mutation rate were  $p_c = 0.75$  and  $p_m = 0.5e^{-DR}$ , inspired by [76], where  $DR$  is the number of domain reductions which will be tuned in section 5.3. The parameter  $DR$  starts at zero and is incremented every generation.

Using a Gaussian mutation operator requires choosing a standard deviation (or mutation step-size). The use of static step-sizes can lead to an inferior algorithm performance, as different values might be optimal at different stages of the evolutionary process. For instance, large step-sizes can be good at early stages, helping the exploration of the search space, while small step-sizes might be used at later generations to tune the suboptimal solution [76]. For this reason, a similar approach to the mutation rate was performed, using  $\lambda = 1.0e^{-DR}$ . The benchmark of the GA and the parameters tuning will be presented in section 5.3.

#### 4.5.2 Rocket Construction and Evaluation

The assembly of the rocket model and the analysis performance is made during the genetic algorithm evaluation operation by the slave nodes. The evaluation algorithm is described by the flowchart in figure 4.12.

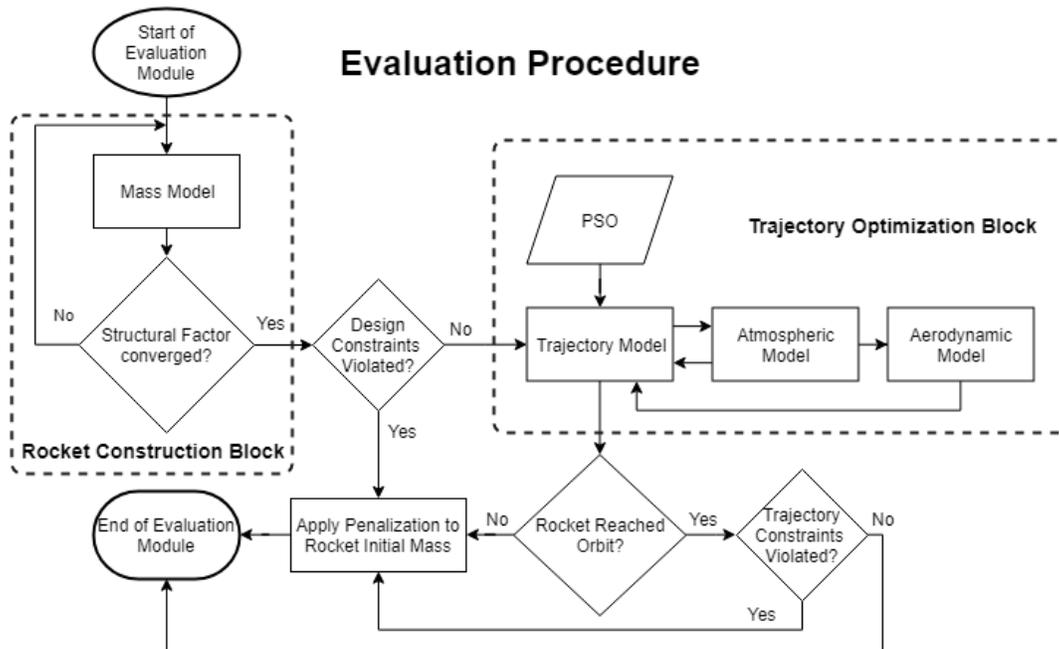


Figure 4.12: Flowchart of rocket construction and trajectory simulation.

The evaluation module can be divided in two blocks: rocket construction block, where the mass and sizing of the rocket is calculated, and the trajectory optimization block, where the optimal trajectory is calculated using the PSO algorithm. After each block, the algorithm verifies the constraints, penalizing the objective function if they are violated.

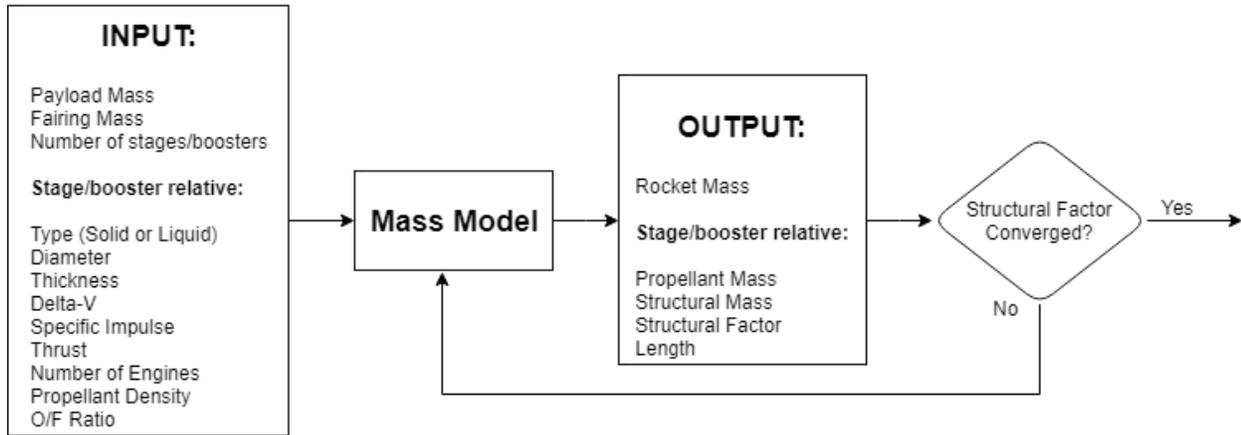


Figure 4.13: Data flow of the rocket construction block.

Firstly, the algorithm proceeds to calculate the mass and dimensions of the rocket using the mass model. The data flow of the construction block is represented in figure 4.13.

The mass model calculates the stages masses and dimensions sequentially, starting by the last stage. Using the user inputs, the mass model calculates the propellant mass using the Tsiolkovsky rocket equation (equation (2.1)), assuming no structural mass at the start of the first iteration. The structural mass is then calculated in parallel with the stage length, using the expressions in section 4.1. This is an iterative process, as adding structural mass requires an increase of propellant mass to achieve the desired  $\Delta V$ . The loop ends when the structural factor converges, progressing to the next stage. In the end, the rocket mass is known by summing the stages, payload and fairing mass.

Before starting the trajectory optimization, the design constraints are checked for violations. If any constraint is violated, the rocket mass suffers a penalty and the individual evaluation ends without performing the trajectory simulation. The design constraints implemented are  $1.2 \leq TWR \leq 2$  at lift-off, with  $TWR$  representing Thrust-to-Weight ratio, to guarantee the successful rocket ascent and preventing the first stage to have too much thrust, avoiding unnecessary aerodynamic loads.

The rocket  $\Delta V$  is limited to  $8500 \text{ m/s} \leq \Delta V \leq 10000 \text{ m/s}$ , since orbital velocity for LEO missions are typically around  $7.0 \text{ km/s}$  to  $7.8 \text{ km/s}$  and velocity losses between  $1.5 \text{ km/s}$  to  $2.0 \text{ km/s}$ . An excessive  $\Delta V$  reflects in a massive rocket, while a rocket with a short  $\Delta V$  is unable to reach the specified orbit. This values can be altered according to the designated mission.

Both constraints allow to diminish the search space, facilitating the search for feasible designs. If no constraints are violated, the trajectory optimization is performed using a PSO algorithm to find the optimal trajectory parameters. The data flow of the trajectory optimization block is represented in figure 4.14.

The trajectory model uses a fourth order explicit Runge-Kutta method (RK4) to generate the numerical trajectory solution. At each time step, the trajectory model communicates with the atmospheric and aerodynamic model to receive information of the atmospheric pressure, drag force, dynamic pressure and the

Knudsen number, which will directly influence the rocket performance. The different flight phases used in the model are the vertical take-off, pitch maneuver, gravity turn and free-flight phase. The phases and equations used in the construction of the trajectory model are described in section 2.4 and 4.2, respectively.

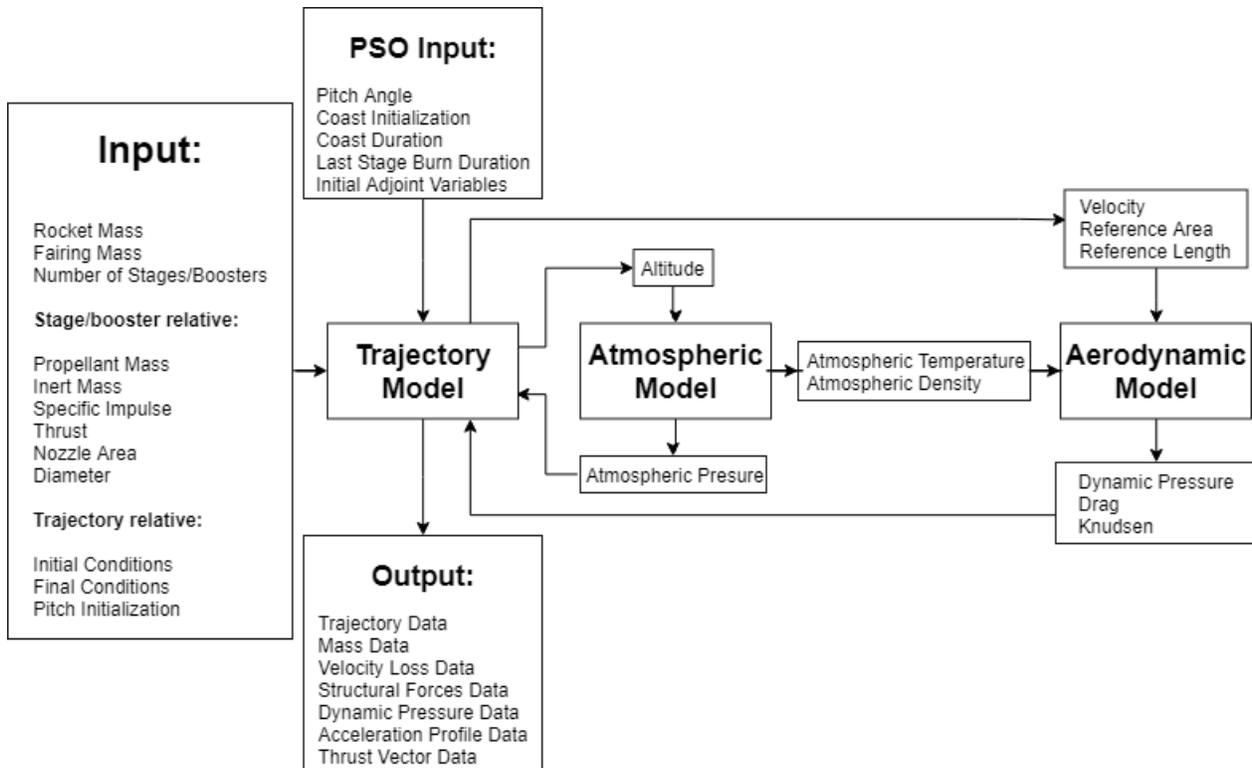


Figure 4.14: Data flow of the trajectory optimization block.

The PSO algorithm provides the trajectory parameters needed to find the optimal path. A trajectory simulation is performed for each candidate solution created by the PSO, until the rocket reaches orbit or when the maximum number of iterations is reached. Each particle is initiated with a parameter set represented by the unknown initial costate values  $(\lambda_{0h}, \lambda_{0V}, \lambda_{0\gamma})$ , the coast duration  $\Delta t_c$  and initialization time  $t_{ic}$ , last stage duration  $\Delta t_T$  and the pitch angle  $\gamma_p$  for the pitch maneuver. At the end of each simulation, the objective function is evaluated, while taking the boundary constraints and optimality conditions in consideration to progressively optimize the trajectory parameters through iterations.

Figure 4.15 illustrates the general trajectory algorithm scheme for a rocket ground launch. Although not illustrated, the algorithm verifies at every time-step if the rocket hits the ground. If the event is verified, the algorithm immediately terminates the trajectory to test the next parameters set. Thus, the algorithm limits the rocket acceleration when using liquid stages ( $a \leq 5g_0$ ) to protect the payload by throttling down the engines [1].

The default rocket trajectory is to perform a vertical lift off, if not specified otherwise. Thus, the default time to start the pitch maneuver is 5 seconds. The pitch maneuver phase is dependent on the rocket acceleration and time for tower clearance. However, the calculated path is just for reference and, even

if using an underestimation of the pitch maneuver initialization time, a real rocket can easily correct the trajectory when aerodynamic forces are negligible (Saturn V rocket only proceeds to guidance correction after the first stage flight [103]).

During the gravity turn and free flight, the stages burn time are monitored. When the propellant tank is depleted, the stage inert mass is dropped, while the next rocket stage begins thrusting. The condition chosen to end the gravity turn was the aerothermal flux to reach a value below  $1135 \text{ W/m}^2$  [12], where the fairing can be jettisoned without warming the payload. The aerothermal flux is evaluated using

$$\phi = \frac{1}{2}\rho V^3. \quad (4.47)$$

This condition allows the rocket to jettison the fairing typically between 100 km to 120 km, which are common altitudes for real rockets [104].

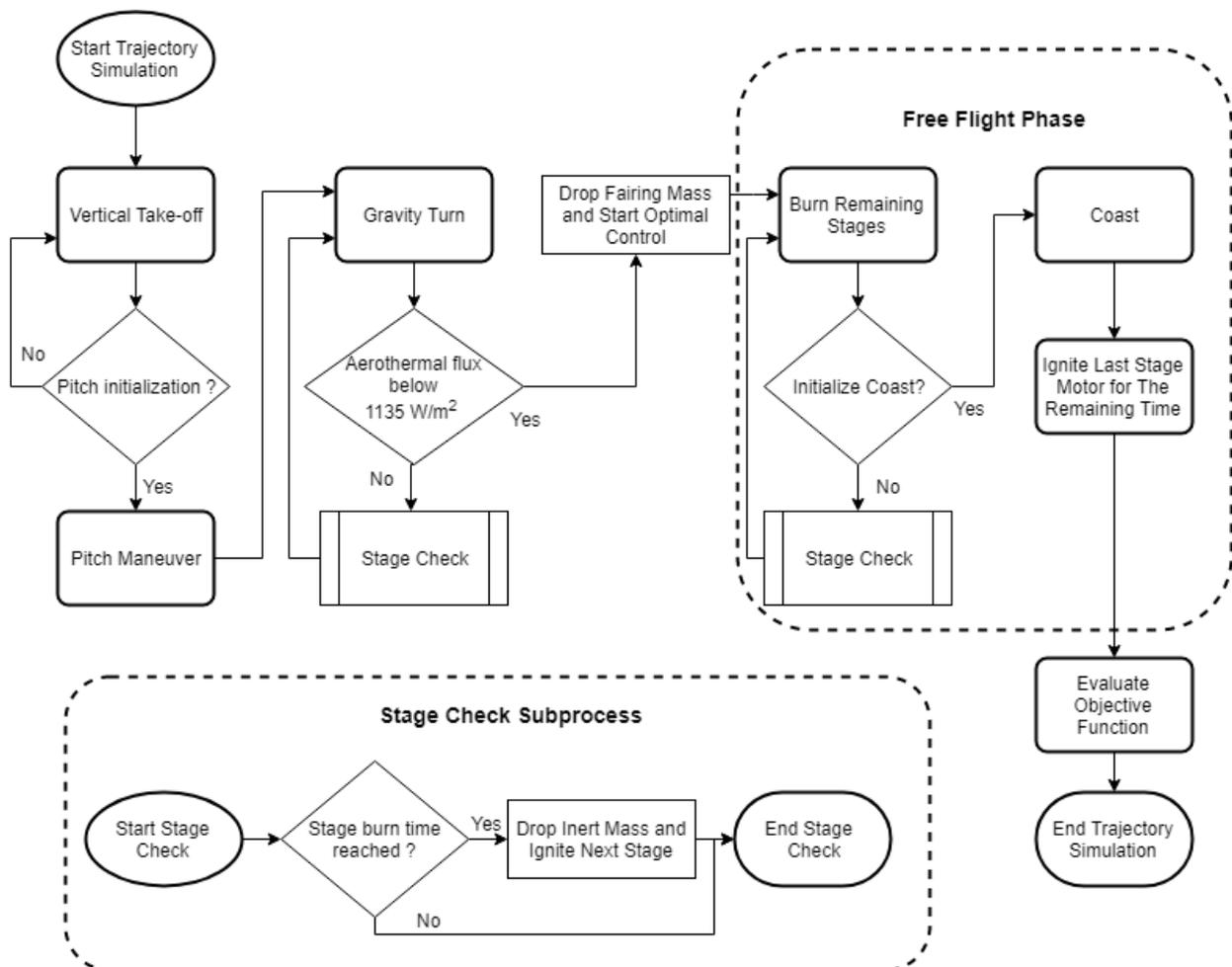


Figure 4.15: Flowchart of trajectory model.

Afterwards, the optimal control, given by the costate variables, is initiated. The rocket continues to thrust until the start of the coast phase, which only occurs during the last stage thrusting. The stage then proceeds

to burn, until the burn time  $\Delta t_T$  is reached.

Finished the trajectory optimization, the algorithm verifies if the rocket has successfully reached orbit. Thus, the algorithm verifies if buckling is imminent and if the maximum dynamic pressure affecting the rocket surpasses the maximum admissible value ( $q \leq 55000 \text{ N/m}^2$  [35]), penalizing the mass if the constraints are violated. The buckling equations are expressed in section 2.5.2.

After verifying all the imposed constraints, the algorithm proceeds to send the mass and optimal parameters information to the master node. A new generation of individuals is then created and the whole process is repeated, until the algorithm reaches the maximum number of iterations or if the optimal solution is not improved by 0.1% over 10 generations.



# Chapter 5

## MDO and Models Validation

To assess the quality of the developed algorithm, the models formulated in chapter 4, as well as the overall procedure, have to be validated. This chapter comprises in the validation of the mass model, using three rockets as reference: Vega, Proton K and Ariane 5 ECA. The trajectory model is tested afterwards, using the Vega rocket, and the coast influence in the trajectory is assessed. Finally, the genetic algorithm is tested by executing a constrained optimization using the Falcon 9 rocket as reference.

### 5.1 Mass Model Validation

The mass model was tested using three cases: mass and sizing evaluation of a solid rocket (Vega [39]), a liquid serial staging rocket (Proton K [40]) and a liquid parallel staging rocket (Ariane 5 ECA [41]). For each case, the real design parameters were used, with the exception of the wall thickness, selected to be in the range of  $5 \text{ mm} \leq t_w \leq 10 \text{ mm}$ . Thus, the material used to calculate the stage and tank wall mass for the three cases was assumed to be an aluminum alloy, with density  $\rho = 2700 \text{ kg/m}^3$ . These assumption may produce errors in the dry mass prediction, but the material properties and wall thickness used in the three rockets are extremely difficult to find in literature.

The payload chosen for each rocket is the maximum mass capacity to LEO. Furthermore, instead of using the heuristic model in section 4.1.3, the real fairing mass was used.

#### 5.1.1 Solid Rocket: Vega

The Vega rocket was designed to launch small payloads to polar and LEO. The rocket is formed by three solid stages: the P80 first stage, the Zefiro 23 second stage, and the Zefiro 9 third stage. All solid stages use solid HTPB as propellant. The upper module is the AVUM liquid module. The module uses a RD-843 rocket engine with  $\text{N}_2\text{O}_4/\text{UDMH}$  as propellant. To estimate the mass and sizing of the Vega, the parameters in table 5.1 were used.

<b>Vega</b>	<b>Stage 1</b>	<b>Stage 2</b>	<b>Stage 3</b>	<b>Stage 4</b>
Stage type	solid	solid	solid	liquid
Diameter [m]	3	1.9	1.9	2.18
Total Mass [kg]	96263	26300	12000	968
Propellant Mass [kg]	87710	23814	10567	550
Thrust [N]	3040000	1196000	225000	2450
Specific Impulse [s]	280	289	295	316
Fuel Density [ $\text{kg m}^{-3}$ ]	1800	1800	1800	1008
Oxidizer Density [ $\text{kg m}^{-3}$ ]	–	–	–	1450
O/F Ratio	–	–	–	1.8
Number of Engines	1	1	1	1
Wall thickness [mm]	5	5	7	9
Payload Mass [kg]		1500		
Fairing Mass [kg]		490		

Table 5.1: Vega rocket characteristics [39].

		<b>Real [kg]</b>	<b>Model [kg]</b>	<b>Deviation</b>
Payload + Fairing	Total Mass	1990	1990	–
	Fuel Mass	550	549	-0.18%
Fourth Stage	Inert Mass	418	414	-0.96%
	Total Mass	2958	2953	-0.17%
	Fuel Mass	10567	10009	-5.28%
Third Stage	Inert Mass	1433	1211	-15.49%
	Total Mass	14958	14173	-5.25%
	Fuel Mass	23814	23125	-2.89%
Second Stage	Inert Mass	2486	2769	+11.38%
	Total Mass	41258	40067	-2.89%
	Fuel Mass	87710	86551	-1.32%
First Stage	Inert Mass	8553	9086	+6.23%
	Total Mass	137521	135704	-1.32%

Table 5.2: Vega mass error analysis.

The relative error between the real and predicted mass is presented in table 5.2. The maximum deviation is presented in the third stage inert mass, explained by the error in the calculated length in table 5.3. The inert mass could be corrected by increasing the stage wall thickness in the third stage. However, this option was not considered. The thickness is unknown and changing it would be a model adaptation to match the real rocket.

With the exception of the third stage, the model overestimates the inert mass and underestimates the propellant mass in all stages. Overall, the maximum relative error is below 15% and it is higher for the solid upper stages. The final rocket mass is close to the real mass, with an error of 1.32%.

	<b>Diameter [m]</b>	<b>Real length [m]</b>	<b>Predicted length [m]</b>	<b>Deviation</b>
Fourth Stage	2.18	2.04	1.84	-9.8%
Third Stage	1.9	4.12	3.23	-21.6%
Second Stage	1.9	8.39	7.47	-11.0%
First Stage	3	11.2	11.22	+0.2%

Table 5.3: Vega sizing error analysis.

The calculated dimensions are shown in table 5.3. With the exception of the first stage, all dimensions were smaller than those of the original Vega and the third stage presents an error larger than 20%, reflecting the inaccuracy of the sizing model to calculate the length of smaller stages. The error is correlated with the underestimation of both propellant and dry mass, being the last due to miscalculation of the solid motor mass and/or structural mass since the thickness is unknown. The error cause may also be due to disagreement between the specifications in the Vega user's manual [39] and in Avio's specifications [105] (Avio is Vega's prime contractor).

The stage mass and dimension are directly dependent on each other. The stage dimension is calculated regarding the required space to store the propellant, but the propellant mass is dependent on the structural mass and, therefore, is dependent on the stage dimensions. Hence, a small error in the calculation of any parameter may cause an error propagation.

Although the second stage and third stage inert mass present a error larger than 10%, it is still below the relative error given by other models [90], and the total rocket mass deviation is small. Hence, the model is suitable for preliminary designs of rockets and will be incorporated in the tool. However, the mass error may have a big impact in the rocket trajectory, comparatively with the real rocket, as it will affect the rocket acceleration. Additionally, as the propellant is consumed, the relative total mass error and the acceleration error increase. Therefore, to simulate the trajectory of a real rocket, it is recommendable to use the rocket real parameters instead of using the model.

### 5.1.2 Liquid Serial Staging Rocket: Proton K

The Proton K rocket is formed by three liquid stages. The first stage used six RD-253 engines and the second and third stage employed four RD-0210 and one RD-0212 engine, respectively. All stages used N2O4/UDMH as propellant. Table 5.4 shows the rocket parameters used to perform mass and sizing prediction, analyzed in table 5.5 and table 5.6 respectively.

<b>Proton K</b>	<b>Stage 1</b>	<b>Stage 2</b>	<b>Stage 3</b>
Stage type	liquid	liquid	liquid
Diameter [m]	5.4	4.1	4.4
Total Mass [kg]	450410	167863	50747
Propellant Mass [kg]	419410	156113	46562
Thrust [N]	10450000	2324970	608220
Specific Impulse [s]	316	327	327
Fuel Density [kg/m <sup>3</sup> ]	793	793	793
Oxidizer Density [kg/m <sup>3</sup> ]	1450	1450	1450
O/F Ratio	2.67	2.67	2.67
Number of Engines	6	4	1
Wall thickness [mm]	9	9	9
Payload Mass [kg]		19760	
Fairing Mass [kg]		2000	

Table 5.4: Proton K rocket characteristics [40].

		<b>Real [kg]</b>	<b>Model [kg]</b>	<b>Deviation</b>
Payload + Fairing	Total Mass	21760	21760	–
	Fuel Mass	46562	46460	-0.22%
Third Stage	Inert Mass	4185	4128	-1.36%
	Total Mass	72507	72348	-0.22%
	Fuel Mass	156113	155006	-0.71%
Second Stage	Inert Mass	11750	11311	-3.74%
	Total Mass	240370	238665	-0.72%
	Fuel Mass	419410	418501	-0.22%
First Stage	Inert Mass	31000	32117	+3.60%
	Total Mass	700780	689283	-0.22%

Table 5.5: Proton K mass error analysis.

The maximum deviation is presented in the second stage inert mass, with an error of 3.74%, explained by the underestimation of the propellant mass. The predicted propellant mass of each stage is slightly smaller than the real mass. Thus, the rocket predicted total mass is similar to the real mass, with an error less than 1%. The model calculates liquid stages mass more accurately than solid stages.

	<b>Diameter [m]</b>	<b>Real length [m]</b>	<b>Predicted length [m]</b>	<b>Deviation</b>
Third Stage	4.1	6.5	6.2	-4.6%
Second Stage	4.1	14	13.9	-0.7%
First Stage	5.4	21.2	20.9	-1.4%

Table 5.6: Proton K sizing error analysis.

Once again, the calculated dimensions in table 5.5 are smaller than the real length, with a maximum relative error below 5%. The sizing model tends to have higher error for top stages. However, the first stage has the second highest error, explained by the larger tanks diameter, comparatively to the other stages. The model sizing equations can be found in [90] and are suitable to calculate the stages dimensions.

The calculated mass and sizing maximum deviation is below 5%. Hence, the model is assumed to be suitable for preliminary design and will be incorporated in the optimization tool.

### 5.1.3 Parallel Staging Rocket: Ariane 5 ECA

Ariane 5 ECA rocket is an European heavy-lift launch vehicle, used to deliver payloads into geostationary transfer orbit or low Earth orbit. The rocket is formed by two core liquid stages and two solid boosters. The boosters use the engine P241 with HTPB as propellant. The first and second stage use one vulcain 2 and one HM7B engine, respectively, Both liquid stages use LH2 / LOX mixture as propellant.

The parameters used to estimate the mass and sizing are shown in table 5.7 and the mass and sizing calculations are expressed in table 5.8 and table 5.9, respectively.

Due to using equation (4.1) to estimate the inert mass of the solid booster, the booster thickness is not required. The boosters propellant and inert mass are calculated with reasonable accuracy. However, the first and second stage core inert mass present a relative error of approximately 10%, which are larger than

<b>Ariane 5 ECA</b>	<b>Booster (each)</b>	<b>Stage 1</b>	<b>Stage 2</b>
Stage type	solid	liquid	liquid
Diameter [m]	3.05	5.4	5.4
Total Mass [kg]	273000	184700	19440
Propellant Mass [kg]	240000	170000	14900
Thrust [N]	7080000	1390000	67000
Specific Impulse [s]	275	432	446
Fuel Density [kg/m <sup>3</sup> ]	1800	71	71
Oxidizer Density [kg/m <sup>3</sup> ]	–	1142	1142
O/F Ratio	–	5.3	5.3
Number of Engines/Boosters	2	1	1
Wall thickness [mm]	–	6	10
Payload Mass [kg]		20000	
Fairing Mass [kg]		2000	

Table 5.7: Ariane 5 ECA rocket characteristics [41].

		<b>Real [kg]</b>	<b>Model [kg]</b>	<b>Deviation</b>
Payload + Fairing	Total Mass	22000	22000	–
	Fuel Mass	14900	14627	-1.83%
Second Stage	Inert Mass	4540	4053	-10.73%
	Total Mass	41440	40680	-1.83%
First Stage	Fuel Mass	170000	172121	+1.25%
	Inert Mass	14700	16160	+9.93%
	Total Mass	226140	228961	+1.25%
Boosters	Fuel Mass	480000	483692	+0.77%
	Inert Mass	66000	65298	-1.06%
	Total Mass	772140	777951	+0.75%

Table 5.8: Ariane 5 ECA mass error analysis.

the results gathered in table 5.5. This error is solely due to the model accuracy, being independent of the boosters inclusion since the mass model starts calculating the propellant and inert mass from the top stage and downwards, calculating the boosters mass in the end. Once again, the error can be explained by the assumption of the material and thickness used, together with the miscalculation of the engine mass. The dry mass error amplifies the fuel mass prediction error and vice-versa, since the calculation of both is an iterative process.

	<b>Diameter [m]</b>	<b>Real length [m]</b>	<b>Predicted length [m]</b>	<b>Deviation</b>
Second Stage	5.4	4.7	5.7	+21.3%
First Stage	5.4	30.5	28.5	-6.6%
Boosters	3	31.6	30.3	-4.1%

Table 5.9: Ariane 5 ECA sizing error analysis.

The second stage predicted length is 20% larger than the real length, although the predicted inert mass is 10% smaller. The error is due to the model using cylindrical tanks with spherical ends to store the propellant. The real second stage uses cylindrical tanks with ellipsoidal ends and a common bulkhead [41], justifying the smaller length. The remaining errors are solely due to the model, and a more complex model could

provide better predictions.

Once again, the mass model calculates the total rocket mass with error less than 1%, despite of the error in the first and second inert mass core stage. Additionally, the error of the first and second stage inert mass is approximately 10%, using the equations in [90]. The model is considered suitable for preliminary design and can be used to optimize parallel staging rockets.

## 5.2 Testing the Trajectory Model

The trajectory model is tested by performing two simulations, with and without coast, to assess its importance in rocket optimization. Both simulations are performed for a Vega rocket, using the parameters in table 5.1. The thrust provided by each stage is equal for both simulations. The mission and the boundaries for the PSO optimization are shown in table 5.10 and 5.11, respectively.

		<b>PSO Boundary Range</b>	
	<b>Mission</b>	Coast time* [s]	250 - 500
Payload Mass [kg]	1500	Pitch Angle [rad]	1.53 - 1.57
Altitude [m]	700000	Adjoint Variables	-1.0 - 1.0
Velocity [m/s]	7037	Last Stage Duration [%]	0 - 100
Flight Path Angle [rad]	0.0	Coast Initialization* [%]	0 - 100

Table 5.10: Mission specification for trajectory validation.

\* - If rocket performs coast

Table 5.11: Trajectory parameters for trajectory validation.

The last stage duration parameter is given as a percentage of the time taken to deplete all the propellant, and the coast initialization parameter is given as percentage of the last stage flight duration (it provides the last stage thrusting time before initializing the coast), since the coast phase can only occur after the last staging. These parameters are only active if coast is used in the trajectory model.

The rocket mission is to deliver a payload of 1500 kg to a circular orbit at 700 km of altitude. The rocket is assumed to perform a vertical take-off from the Vega Launch Complex (SLV) in Kourou, French Guiana. Kourou is located at a latitude of  $5^\circ$ , where the eastward boost provided by the Earth's rotation is about 463 m/s. Hence, the rocket has to achieve a final velocity of 7037 m/s in a referential centered at Earth's center of mass and rotating with the Earth.

The PSO ideal population size and iteration number are highly dependent on the complexity of the problem. To test the trajectory convergence, the number of particles and iterations were set to 250 and 1000, respectively, as the PSO provides better results for larger populations and higher number of iterations [106]. However, these parameter values are only used to test the trajectory, as they are highly computational expensive to be used during the optimization process. The remaining parameters were tuned according to the PyGMO default values [101], with  $c_1 = c_2 = 2.05$ ,  $w = 0.7298$  and a particle maximum velocity of 0.5

(normalized using the bounds width). The optimization was performed using a Intel Xeon E312xx (Sandy Bridge), with 2000 MHz.

The resultant final rocket position and the PSO optimal solution are shown in table 5.12.

		Without Coast	With Coast
<b>Final Position</b>	Altitude [m]	700000	699993
	Velocity[m/s]	7037.00	7037.08
	Flight Path Angle [rad]	-1.79e-09	-1.10e-05
<b>PSO Optimal Solution</b>	Coast time [s]	–	442
	Pitch Angle[rad]	1.541	1.537
	Adjoint variable $\lambda_h$	-4.20e-03	-3.07e-03
	Adjoint variable $\lambda_V$	-5.96e-01	-4.94e-01
	Adjoint variable $\lambda_\gamma$	5.45e-02	-3.43e-01
	Last Stage Duration [%]	78.3	33.9
	Coast Initialization [%]	–	0.02

Table 5.12: Optimal trajectory parameters and final rocket state.

The trajectory optimization without coast phase took 835 seconds, while the optimization including coast took 1530 seconds. This is due to the introduction of two new parameters in the PSO, delaying convergence. Both simulations reached the desired orbit although the error in orbit insertion is slightly greater when introducing coast time. The use of a coast stage is beneficial for trajectory optimization, as it decreased the last stage duration, saving 224.2 kg of propellant mass. This reduction is advantageous to the preliminary design, allowing to reduce the total mass by withdrawing the unused propellant. The coast initialization low value informs that it's better for the rocket to perform coast before igniting the top stage. Due to the low thrust in the top stage, the rocket requires a large thrusting time for orbit injection.

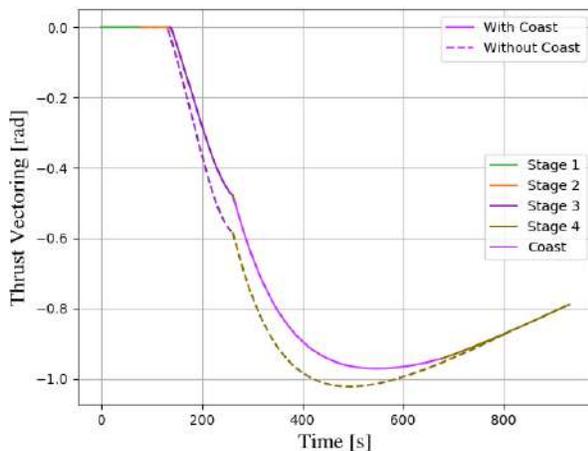


Figure 5.1: Vega thrust vectoring.

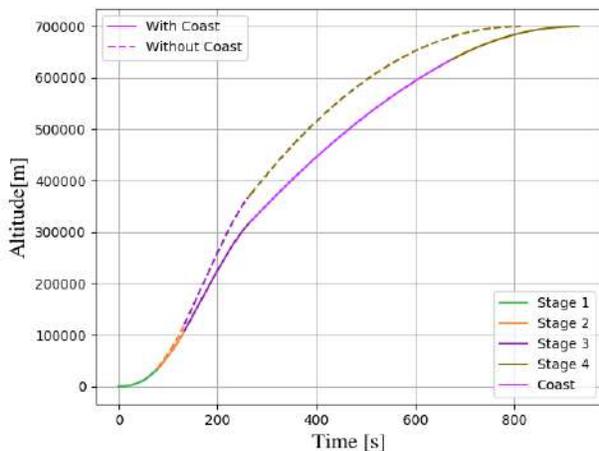


Figure 5.2: Vega altitude.

The thrust vectoring control for both simulations are graphically shown in figure 5.1 Both simulations present a similar graphical shape for vectoring control, even using different initial values for the adjoint variables, shown in table 5.12. The simulation without coast requires a greater thrust angle deflection, which will lead to an increased velocity loss due to vectoring. The optimal control starts at 142 seconds.

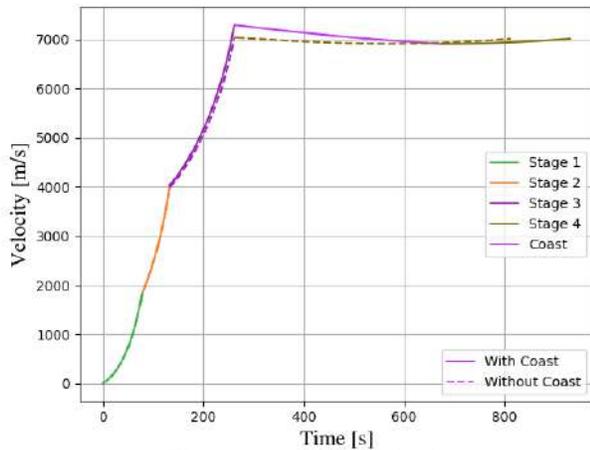


Figure 5.3: Vega velocity.

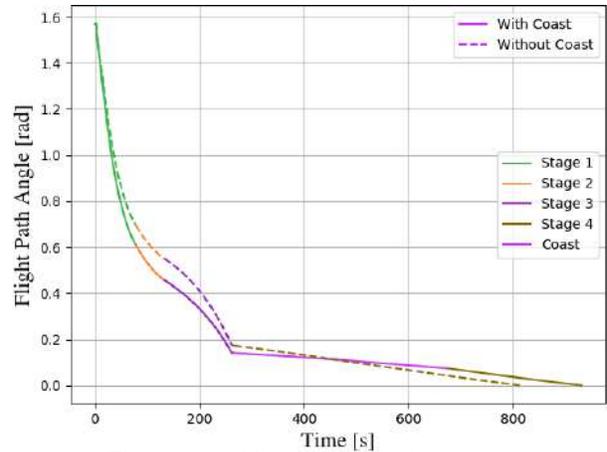


Figure 5.4: Vega flight path angle.

The altitude, velocity and flight path angle evolution with time are shown in figures 5.2, 5.3 and 5.4, respectively. Both simulations reached the pretended orbit with a difference of 125 seconds. At the end of the third stage thrusting, the simulation with coast presents an increase of 200 m/s, due to turning earlier, allowing the inclusion of the coast phase without losing too much energy to reach orbit.

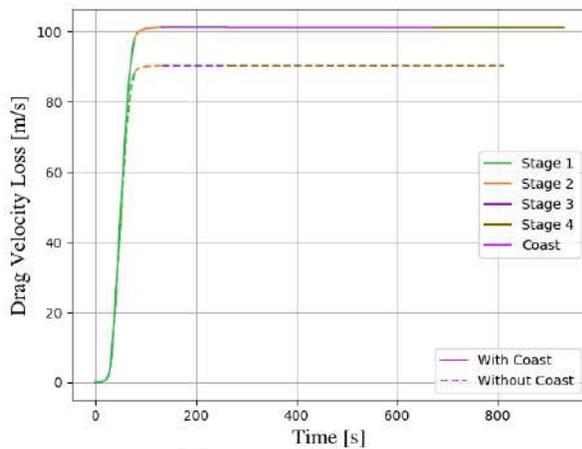


Figure 5.5: Velocity loss due to drag.

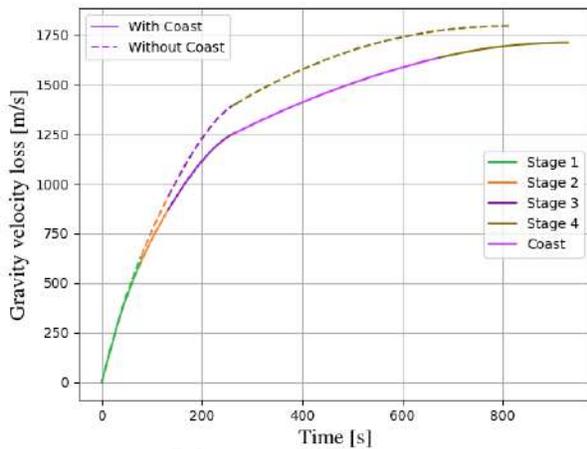


Figure 5.6: Velocity loss due to gravity.

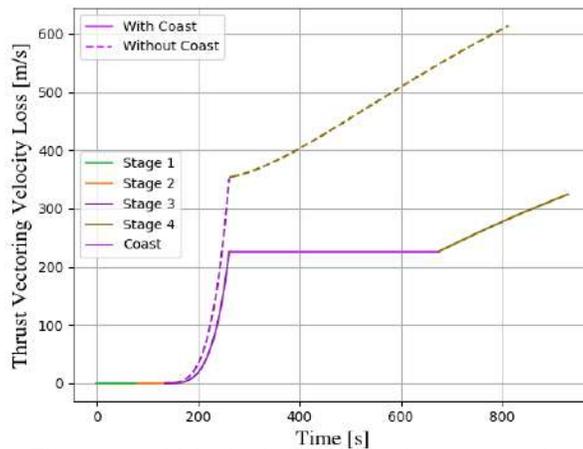


Figure 5.7: Velocity loss due to thrust vectoring.

The figures illustrate the velocity loss over time due to drag (figure 5.5), gravity (figure 5.6) and thrust vectoring (figure 5.7). The trajectory simulation with coast presents an higher velocity loss due to drag, explained by the lower value of pitch angle, meaning the rocket takes more time to leave the tangible atmosphere (the thrust is identical in both simulations). However, the loss due to gravity and thrust vectoring is largely decreased. The loss reduction due to gravity is also justified by a lower pitch value, making the rocket to turn more quickly to the horizontal position, observable in figure 5.6. The loss due to thrust vectoring is reduced with the inclusion of the coast phase. During coast phase, the rocket stops thrusting, preventing further velocity losses, and turns only using gravity, observable in figure 5.7.

As previously mentioned in section 2.1, the velocity losses are typically between 1.5 to 2 km/s, predominantly due to gravity, and it is verified for the simulation using coast. The trajectory model is able to successfully calculate an optimized trajectory, minimizing the propellant consumption while accurately reaching the required orbit. However, the thrust vectoring angle in figure 5.1 for both simulation exceeds  $45^\circ$ . Some works limit the thrust vectoring, such is the case of [12] where the angle never surpasses  $15^\circ$ . However, the vectoring control only starts when aerodynamic loads are negligible and, therefore, the thrust vectoring angle is not limited in this model.

The model shows promising results and is incorporated in the tool. Thus, a coast stage will be used to optimize the trajectory when designing a rocket since it shows improvements in the rocket performance during flight.

### 5.3 Genetic Algorithm Benchmark

The benchmark of the GA is accomplished by performing several single-objective test functions optimization, and comparing it with the GA in the DEAP (Distributed Evolutionary Algorithms in Python) library [107]. Both algorithms performed the optimization of the functions shown in table 5.13, using the dimensions and space search mentioned in table 5.14.

Function	Expression
F1 - Sphere	$f(x) = \sum_{i=1}^d x_i^2$
F2 - Rosenbrock	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F3 - Ackley	$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}) - \exp(\frac{1}{d} \sum_{i=1}^d \cos 2\pi x_i) + 20 + \exp(1)$
F4 - Schwefel	$f(x) = 418.9829n - \sum_{i=1}^d x_i \sin \sqrt{ x_i }$
F5 - Rastrigin	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$

Table 5.13: Optimization benchmark functions

Function	Optimal result	Dimension	Search Space
F1	$f(x^*) = 0$ , at $x^* = (0, \dots, 0)$	d=20	$-5.12 \leq x_i \leq 5.12$
F2	$f(x^*) = 0$ , at $x^* = (1, \dots, 1)$	d=2 and d=5	$-2.048 \leq x_i \leq 2.048$
F3	$f(x^*) = 0$ , at $x^* = (0, \dots, 0)$	d=2	$-30 \leq x_i \leq 30$
F4	$f(x^*) = 0$ , at $x^* = (420.9687, \dots, 420.9687)$	d=10	$-500 \leq x_i \leq 500$
F5	$f(x^*) = 0$ , at $x^* = (0, \dots, 0)$	d=10	$-5.12 \leq x_i \leq 5.12$

Table 5.14: Optimization benchmark function solutions and selected parameters.

The Sphere and Rosenbrock are unimodal functions, having only one optimal value with no local minima. The Sphere function is strongly convex, contrarily to the Rosenbrock function. The Rosenbrock global minimum is inside a long, narrow, parabolic shaped flat valley, making convergence harder when using free-gradient optimizers.

Several standard optimization algorithms struggle to leave local optima solutions. To test how the optimizers deal with the problem, the Ackley, Schwefel and Rastrigin multimodal functions are tested.

The results and comparison of the implemented GA and DEAP's GA are shown in table 5.15. For comparison, both algorithms used constant parameters. The crossover rate and the mutation rate chosen were  $p_c = 0.75$  and  $p_m = 0.2$ , as suggested by [108] and [109]. Due to using a constant step-size, the chosen value was  $\lambda = 0.01(max - min)$  since high values difficult the solution convergence.

Function	Parameters	Method	Average Minimum	Average Time [s]
F1	gen=2000	GA	1.975e-06	14.11
	pop=40	DEAP	8.901e-05	12.91
F2 (d=2)	gen=5000	GA	4.378e-06	19.08
	pop=40	DEAP	3.701e-07	19.93
F2 (d=5)	gen=5000	GA	3.231e-02	46.00
	pop=40	DEAP	8.905e-03	46.52
F3	gen=2000	GA	5.507e-04	8.55
	pop=40	DEAP	6.070e+00	10.28
F4	gen=2000	GA	1.602e-05	55.07
	pop=200	DEAP	6.412e-05	85.75
F5	gen=5000	GA	2.363e-05	67.11
	pop=100	DEAP	1.138e+01	91.28

Table 5.15: Comparison between implemented GA and DEAP's GA.

The analysis was performed by running the optimization 10 times and calculating the average minimum.

Both algorithms present similar minimas, with the exception of the Ackley and Rastrigin function optimization. DEAP's GA is unable to find the optimal solution due to the existence of several local minima. The run time of both GA is similar when a reduced population number is used. However, increasing the population number results in DEAP's GA substantially exceeding the run time of the implemented GA. Hence, between both optimizers, the implemented GA seems to be the best for rocket optimization.

As discussed in section 4.5.1, adaptive mutation rate and step-size are used to narrow the search space during optimization. The Ackley function with two dimensions is used to perform the parameter tuning. The average number of generations required to converge after running the optimization 10 times for each parameter couple are shown in table 5.16. A maximum of 500 generations and 100 individuals are used for optimization process, ending prematurely when an individual reaches the fit value of  $10^{-5}$ .

		DR incrementation for mutation rate									
		0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04	0.045	0.05
DR incrementation for step-size	0.055	129	111	117	120	175	72	171	–	–	–
	0.06	99	118	118	122	107	159	143	63	–	–
	0.065	98	94	107	133	117	160	140	144	–	–
	0.07	92	93	83	108	97	110	95	127	134	–
	0.075	94	59	104	81	47	120	104	143	127	–
	0.08	76	71	78	59	71	80	118	98	97	–
	0.085	71	88	81	95	79	103	99	108	105	107
	0.09	81	72	82	67	70	111	110	96	128	106
	0.095	61	66	80	65	66	93	75	71	91	125
	0.1	61	58	61	79	81	68	74	74	102	108

Table 5.16: Convergence influence study of the DR parameter.

Observing table 5.16, the required number of generations to find the optimal solution decreases with the increase of the Domain Reduction (DR) incrementation for the step-size, in opposite to the mutation rate. The minimal value of required generations was 47, using  $p_c = 0.5e^{-0.025 \times gen_k}$  and  $\lambda = 1.0(max - min)e^{-0.075 \times gen_k}$ . These expressions are used for the rocket design optimization.

The GA is tested latter by performing a optimization for a specific mission, using state-of-the-art existing rocket. The Falcon 9 rocket was chosen due to currently being the most technologically advanced rocket, in terms of material and propulsion [8]. The original Falcon 9 employs nine Merlin-1DV engines in the first stage and one Merlin-1D in the second stage, using LOX / RP-1 as propellant. The Merlin engines present a TWR of 179.8, being the most efficient modern engine.

	Mission
Altitude [m]	700000
Velocity [m/s]	7037
Flight Path Angle [rad]	0.0
Payload Mass [kg]	10000

Table 5.17: Mission specification for GA validation.

	Constraints
Dynamic Pressure [Pa]	< 55000
Acceleration [m/s <sup>2</sup> ]	< 5g <sub>0</sub>
Buckling	< Safety Load
Total Delta-V [m/s]	8500 < DV < 10000
Take-off TWR	1.2 < TWR < 2.0

Table 5.18: Constraints for GA validation.

The mission objective and the imposed constraints are shown in tables 5.17 and 5.18 respectively. For the buckling constraint, a safety factor of 1.5 was selected. The parameters chosen for optimization were the number of Merlin engines and the wall thickness of each stage, together with the propellant mass division between them. A list of the design parameters used can be seen in table 5.19, as well as the characteristics of the Merlin engines in table 5.20

	Stage 1	Stage 2
Stage type	liquid	liquid
Diameter [m]	7.4	4.1
DV [m/s]	3000-6000	3000-6000
Engine	Merlin-1D	Merlin-1DV
Number of Engines	5-8	1-2
Wall thickness [mm]	5-10	5-10
Fairing Mass [kg]	1500	

Table 5.19: Design parameters for GA validation.

	Merlin-1D	Merlin-1DV
Thrust [N]	919140	939400
Specific Impulse	311	348
Nozzle Area	0.9	3.0
Mass	470	470
Fuel Density	810	810
Oxidizer Density	1142	1142
O/F Ratio	2.61	2.61

Table 5.20: Merlin engine characteristics.

The optimization was performed testing multiple population dimensions to study how it affects the convergence for the optimal solution. The analysis is performed using a maximum number of 50 generations as shown in figure 5.8.

Population Dimension	Number of Evaluations
50	1500
100	2000
150	2400

Table 5.21: Individual evaluations.

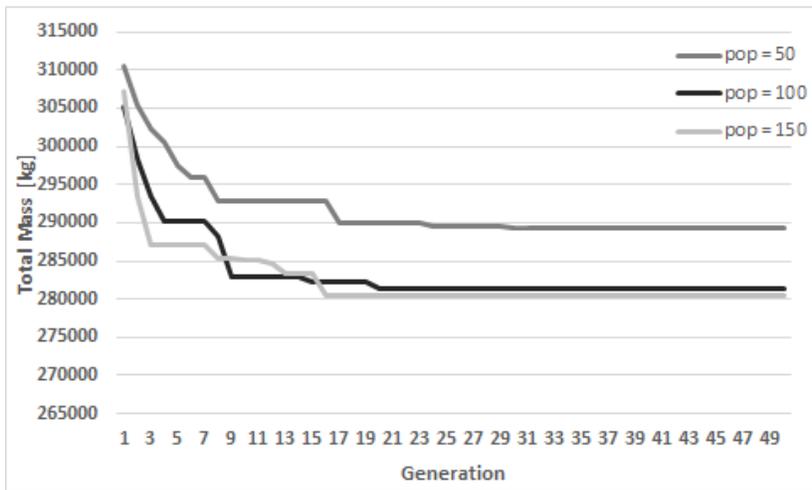


Figure 5.8: Rocket total mass along generations.

From the three dimensions tested, the population with 150 individuals presented the best solution. The three populations converged to a solution before reaching the maximum number of generations. The required amount of generations before converging reduces with the population increase. However, as shown in table 5.21, the number of individual evaluation increases, requiring higher computational power.

The optimal design and trajectory parameters are shown in table 5.22 and table 5.23, respectively. The rocket reached orbit using nearly all the propellant, indicating that the optimization algorithm reduced the excessive propellant mass. The design and trajectory parameters are within the chosen boundaries.

	Stage 1	Stage 2
DV [m/s]	3392	5700
Number of Engines	6	1
Wall thickness [mm]	6.74	5.07

Table 5.22: Optimal design parameters.

	Optimal Value
Coast Time [s]	677
Pitch Angle [rad]	1.539
Adjoint Variable $\lambda_h$	-9.84e-04
Adjoint Variable $\lambda_V$	-8.45e-01
Adjoint Variable $\lambda_\gamma$	6.52e-01
Last Stage Duration [%]	98.9
Coast initialization [%]	97.4

Table 5.23: Optimal trajectory parameters.

Analyzing the imposed constraints, the TWR at lift-off has the value of 1.8 and the rocket total  $\Delta V$  is equal to 9092 m/s, which is inside the selected range. Observing the dynamic pressure in figure 5.9, it has a maximum value of 53.5 kPa, below the imposed limit. The dynamic pressure peak is achieved 55 seconds after lift-off. Comparatively, the Falcon 9 rocket achieves maximum dynamic pressure 80 seconds after take-off, which is significantly higher than the optimized rocket. This divergence is due to the falcon 9 having an initial TWR of 1.28, lower than the optimized rocket.

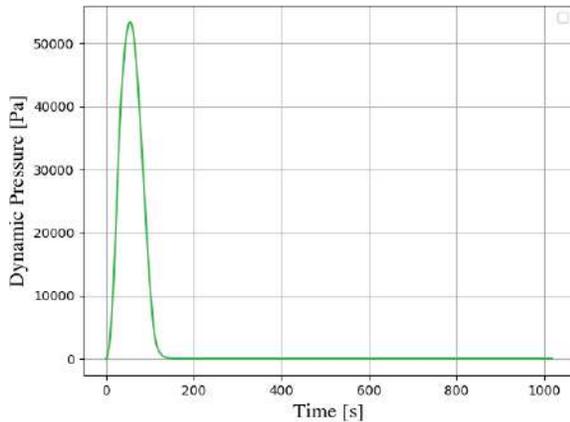


Figure 5.9: Dynamic pressure during flight

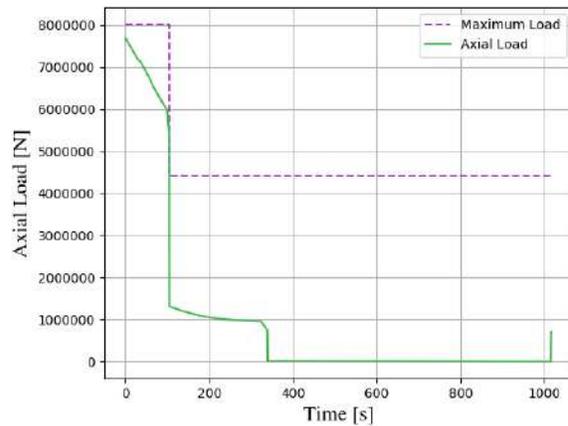


Figure 5.10: Axial load during flight.

The axial load also is below the safety load, observable in figure 5.10, using a safety factor of 1.5. The maximum axial load felt on the rocket is shortly after take-off, when the rocket has maximum weight and is producing a large amount of thrust to ascend.

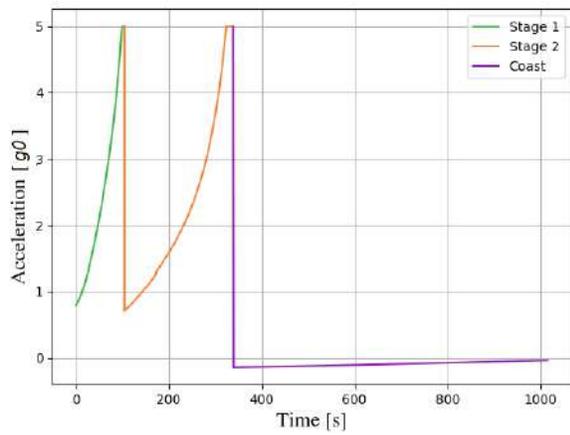


Figure 5.11: Rocket acceleration during flight.

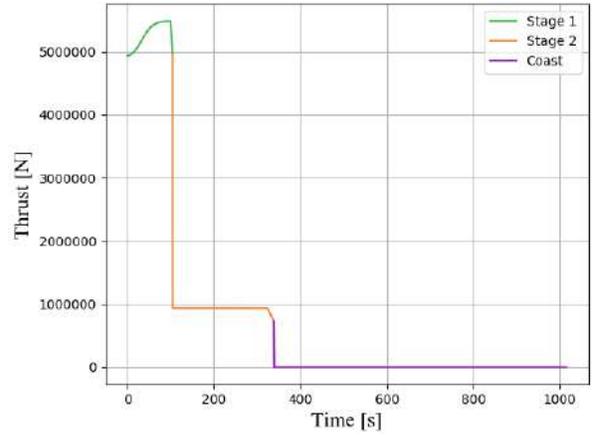


Figure 5.12: Rocket thrust during flight.

The acceleration felt by the rocket during flight is shown in figure 5.11. The rocket acceleration never surpasses the imposed constraint of  $5g_0$  by throttling down the engine at the end of each stage, observable in figure 5.12. Throttling the engine lowers the thrust provided by the stage, controlling the acceleration.

The GA optimizer is able to handle the imposed constraints, and therefore, is validated to perform the preliminary design of a small rocket, discussed in section 6.

## Chapter 6

# Preliminary Design of a Small Launch Vehicle

During the last decades, with the evolution of microelectronics, electronic devices have experienced enormous advancements in terms of performance and reliability. The reduction of production costs and sizing not only allowed the creation of smaller and lighter satellites, but also for academic organizations to start designing and building their own satellite.

Small satellites (or SmallSats) have a big advantage compared to big satellites: it is possible to apply a fly-learn-refly approach to the development and launch of Smallsats. This approach leads to shorter development cycles and teams, lower costs, and accelerated space technology innovation.

About one third of the satellites launched in the past 20 years were SmallSats [110]. Thus, the amount of Smallsats launched in the past 5 years is nearly equivalent to the amount launched in the previous 15 years, indicating a considerable rise in the use of SmallSats since the beginning of the millenium.

Satellites are still deployed into orbit by launch vehicles capable of carrying thousands kilograms of payload mass. Usually, the SmallSats fly as secondary payload (or piggy bank). It has the advantage of low price, but implies a defined orbit, eventually causing launch delays.

To address the increasing number of SmallSats in the market, new commercial Small- and Micro-LVs are currently under development. These launch vehicles are specifically used to carry small payloads to orbit. However, to be competitive with current launchers in the market, small launchers are required to deliver payload at competitive prices.

One example of a small-LV is the new state-of-the-art Electron developed by Rocket Lab [111], aiming to deliver 150 kg into a 500-km sun synchronous orbit. The Electron presents various similarities with the Falcon 9, having both the same number of stages filled with the same propellant. Thus, both have nine and one engine in the first and second stage, respectively. The Electron rocket has a total mass of 12,500 kg and is composed by two stages with a diameter of 1.2 meters and the core body is constructed using

a lightweight carbon composite material. Both stages use RP-1/LOX as propellant and Rutherford rocket engines to provide thrust. The Rutherford engines are the first electric-pump-fed engines, being powered by li-polymer batteries. The vast majority of the engine components are 3D printed, saving time and costs during the manufacturing process [111].

In the remark of testing how well the tool designs a rocket, a Small-LV optimization design is conducted, using an Electron's reference mission, allowing to compare the characteristics of the obtained rocket and the Electron.

## 6.1 Algorithm Setup

For the preliminary design, since there are no Small-LVs using boosters, the optimization will focus only on two-stage and three-stage rockets. To compare with the Electron rocket, the considered mission is shown in table 6.1. The chosen launch site is Rocket Lab Launch Complex 1 on Mahia Peninsula, New Zealand, which has a latitude of  $39.2615^\circ$  S. For sun synchronous missions, the Earth rotation has small influence on the rocket velocity to reach orbit, and therefore, will not be accounted.

	<b>Mission</b>
Payload Mass [kg]	150
Altitude [m]	500000
Velocity [m/s]	7612
Flight Path Angle [rad]	0.0

Table 6.1: Mission specification.

The trajectory optimization parameters are shown in table 6.2. These parameters are used for both two-stage and three-stage rockets. To prevent lack of propellant due to engine malfunction, the maximum value for the last stage duration is 95%, leaving 5% of propellant as reserve.

	<b>PSO Boundary range</b>
Coast time [s]	500 - 4000
Pitch Angle [rad]	1.55 - 1.57
Adjoint Variables	-1 - 1
Last Stage Duration [%]	70 - 95
Coast Initialization [%]	0 - 100

Table 6.2: Trajectory parameters for optimization.

The propulsive parameters are specified in table 6.3. For a better comparison between the optimized rockets and the Electron, the propellant used and the specific impulse are unchanged. Each stage will only have one propulsive engine, reducing the design space.

The exit nozzle diameter optimization is not included in the optimization process and needs to be assumed. It is highly dependent of the engine thrust, pressure and specific impulse. The first stages usually have a smaller diameter than the top stages, relatively to top stages, as the last need to be optimized for

vacuum operation. Although it may not be optimal, the first stage nozzle diameter is assumed to be 60% of the stage diameter, similar to Ariane 5 [41] and Vega [39], while for the top stages is assumed to be 90 %, being optimized for vacuum propulsion.

	<b>First Stage</b>	<b>Remaining Stages</b>
TWR	1.2 - 2.0	0.8-1.5
Isp [s]	303	333
Nozzle Diameter [m]	$0.6D_{stage}$	$0.9D_{stage}$
Fuel Density [kg/m <sup>3</sup> ]	810	810
Oxidizer Density [kg/m <sup>3</sup> ]	1142	1142
O/F Ratio	2.61	2.61

Table 6.3: Propulsive and propellant parameters.

The design parameters are shown in table 6.4, for the two-stage and three-stage rocket. To reduce the design search space, the diameter and wall thickness will be equal for all stages. The material chosen for both tank walls and for the rocket walls was the aluminum alloy, in contrast with the carbon composite used in the Electron's wall.

	<b>Two Stage Rocket</b>	<b>Three Stage Rocket</b>
Rocket type	liquid	liquid
Delta-V per Stage [m/s]	3000 - 6000	2000 - 4000
Diameter [m]	1.0 - 1.5	1.0 - 1.5
Wall Thickness [mm]	2 - 5	2 - 5
Engines per Stage	1	1

Table 6.4: Design Parameters.

Comparing to table 5.19, the wall thickness range imposed for the small launch vehicle optimization is largely inferior. This is due to the weight and thrust difference when comparing small and heavy launch vehicles. Smaller launchers have less mass and require less thrust to perform the mission. Hence, the wall thickness can be reduced, without compromising the rocket structure. Thus, to prevent unfeasible designs, the optimization is subjected to the constraints in table 6.5, similar to the constraints used in section 5.3. Once again, the safety load proposed is equal to the critical load, divided by a safety factor of 1.5.

<b>Constraints</b>	
Dynamic Pressure [Pa]	$< 55000$
Acceleration [m/s <sup>2</sup> ]	$< 5g_0$
Buckling	$< \text{Safety Load}$
Total Delta-V [m/s]	$8500 < DV < 10000$
Take-off TWR	$1.2 < TWR < 2.0$

Table 6.5: Imposed constraints.

Due to computational constraints, the optimization was performed with a population of 50 individuals along 35 generations although a larger population could provide better results, as shown in section 5.3. For each individual at each generation, a trajectory optimization using PSO is performed, with 100 particles

along 250 generations. The PSO optimization ends prematurely if it is able to find the optimal trajectory within 0.1% error for the insertion altitude and velocity, and a maximum deviation of 0.01 rad for the final flight path angle, to save optimization time. A better trajectory optimization is performed after the best rocket design is found.

GA Parameters	
Maximum Generation	35
Number of Individuals	50
Crossover Rate	0.75
Mutation Rate	$0.5e^{-0.025gen_k}$
Step-Size	$1.0e^{-0.075gen_k}$

Table 6.6: Genetic algorithm parameters.

PSO Parameters	
Maximum Generation	250
Number of Particles	100
Cognitive Parameter	2.05
Social Parameter	2.05
Inertia Weight	0.7298

Table 6.7: Particle swarm optimization parameters.

The parameters of both GA and PSO optimization algorithms are repeated in table 6.6 and table 6.7, respectively, for convenience. The step-size is normalized with the boundary width.

## 6.2 Results

The algorithm took 9641 seconds to find the optimal two-stage rocket design and 7370 seconds to find the optimal three-stage rocket design. The task was parallelized using a cluster with 13 Intel Xeon E312xx (Sandy Bridge) processors, 4 cores each. Each core has a frequency value of 2000 MHz.

The algorithm convergence is illustrated in figure 6.1. The algorithm was able to converge within the maximum number of generations. The best individuals have a larger mass variation during the first generations, which is gradually reduced throughout generations.

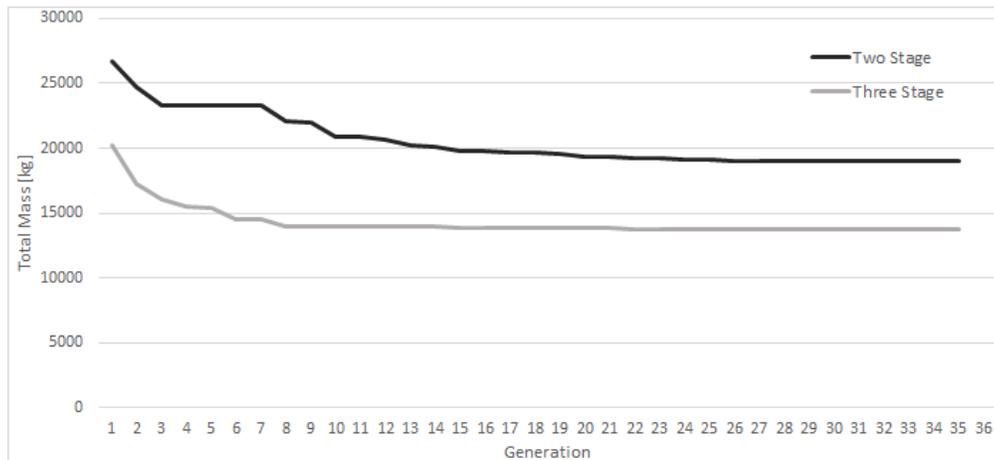


Figure 6.1: Rockets best total mass evolution.

The evolution of feasible and unfeasible two-stage and three-stage rockets are shown in figure 6.2 and

6.3, respectively. The three-stage optimization population starts with a very small number of feasible designs able to reach trajectory, and the two-stage optimization population has no feasible designs in the first generation. This issue can be corrected by increasing the GA population. Nevertheless, the number of feasible designs increase substantially during the first generations, while the rockets with unfeasible design decrease, remaining null after a few generation. The number of unfeasible trajectories does not follow a trend and could be attenuated by increasing the number of generations in the PSO algorithm. However, it would lead to a higher computational cost.

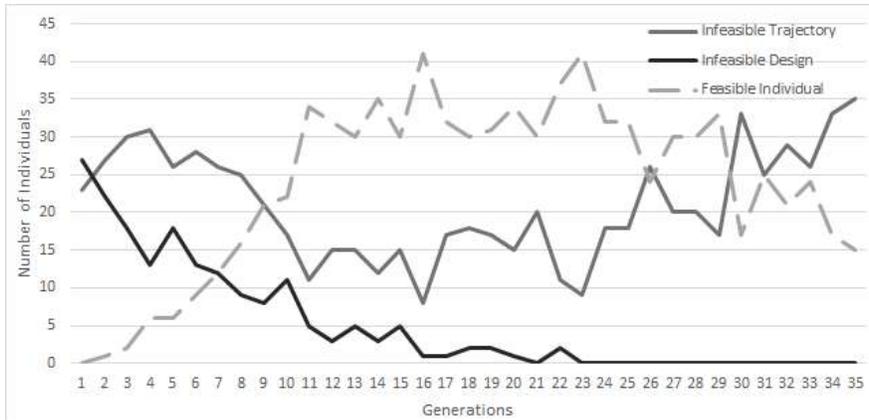


Figure 6.2: Analysis of feasible and unfeasible two-stage rockets.

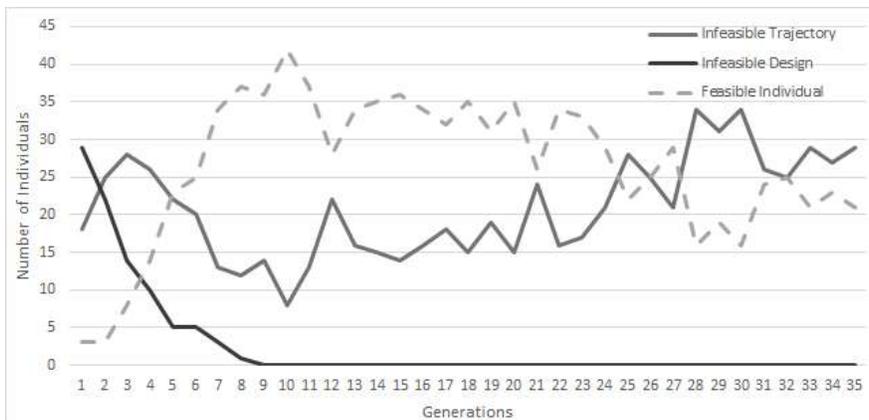


Figure 6.3: Analysis of feasible and unfeasible three-stage rockets.

The GA function calls is proportional to the number of individuals and generations. For both optimizations, the GA performed 1750 design evaluations. However, the trajectory optimization of computation cost is the most demanding step for the global optimization. The number of trajectory calls per generation are shown in figure 6.4. The number of trajectory function presents the same behaviour as the number of unfeasible trajectories in figure 6.2 and 6.3. When the trajectory is unfeasible, the PSO is not able to converge and needs to run until the maximum number or iterations is reached, leading to a substantial increase in computation time and function calls.

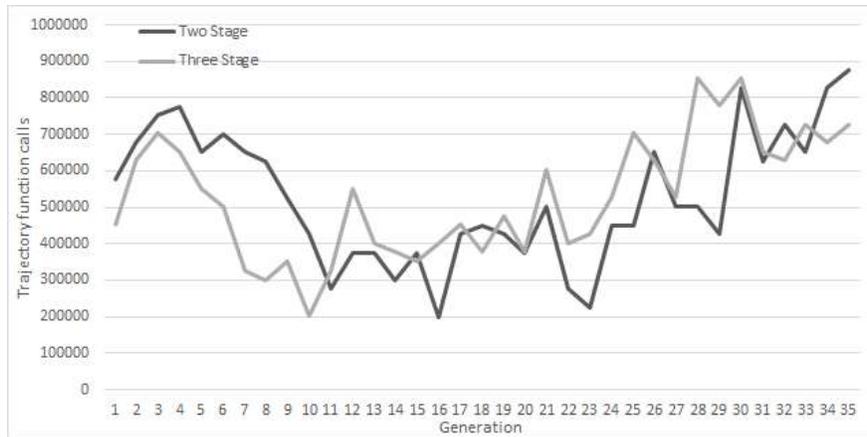


Figure 6.4: Number of trajectory calls per generation.

The optimal design parameters of the two-stage and three-stage rockets are shown in table 6.8 and table 6.9, respectively.

	First Stage	Second Stage
Delta-V [m/s]	4152	5334
TWR	1.68	1.07
Thickness [mm]	2.01	2.01
Diameter [m]	1.05	1.05
Propellant Mass [kg]	14271	2515
Inert Mass [kg]	1574	383
Fairing Mass [kg]		73
Total Mass [kg]		18971

Table 6.8: Two-stage rocket optimal design parameters.

	First Stage	Second Stage	Third Stage
Delta-V [m/s]	3123	3255	2861
TWR	1.80	1.02	0.82
Thickness [mm]	2.01	2.01	2.01
Diameter [m]	1.00	1.00	1.00
Propellant Mass [kg]	8938	2272	554
Inert Mass [kg]	1205	379	177
Fairing Mass [kg]		70	
Total Mass [kg]		13744	

Table 6.9: Three-stage rocket optimal design parameters.

Both rockets comply with the constraints specified in table 6.5. The three-stage rocket, as expected, is a better alternative to the two-stage rocket, drastically reducing the total mass by 5 tonnes. The wall thickness and diameter of both rockets tends for the minimum possible value, suggesting they could still be decreased if it does not interfere with the rocket structural integrity. They also present a first stage with a high thrust-to-weight ratio. The value is optimized as a compromise between the velocity loss due to drag and due to gravity. The rocket needs to leave the tangible atmosphere as fast as possible to prevent unnecessary

gravity losses by starting the optimal control, but high acceleration increase the drag loss.

	<b>Two Stage Optimal Value</b>	<b>Three Stage Optimal Value</b>
Coast Time [s]	2734	2977
Pitch Angle [rad]	1.566	1.560
Adjoint Variable $\lambda_h$	-9.95e-04	-9.97e-04
Adjoint Variable $\lambda_V$	-7.79e-01	-1.89e-01
Adjoint Variable $\lambda_\gamma$	-8.88e-01	-8.35e-01
Last Stage Duration [%]	95.0	95.0
Coast initialization [%]	98.8	98.3

Table 6.10: Rocket optimal trajectory parameters.

The optimal trajectory parameters are shown in table 6.10 and are within the specified boundaries. The optimal value for coast time is superior to 45 minutes in both cases, which is only possible due to the high TWR in the first stage and fast turning after fairing jettison, allowing the rocket to obtain sufficient velocity. The last rocket stage burns until it consumes 95% of the available propellant, leaving the remaining 5% as reserve. The coast phase is initialized near the last stage ending time for both rockets. Thus, the last stage provides an optimal impulse to reach circular orbit by using between 1% to 2% of the last stage burn time at the end of the flight.

The rocket altitude evolution with time is illustrated in figure 6.5. Both the rocket reached the required altitude but with a time difference of 410 seconds. Both rockets appear to have the same behaviour before the beginning of the coast time, as they tend to decrease the flight path angle as soon as possible to decrease the losses due to gravity. The rocket starts to ascent again only after starting the coast phase due to the term  $\frac{V^2}{Re+h}$  in equation (4.23), which accounts for the Earth curvature. This is only possible if the rocket acquires enough velocity to reach orbit.

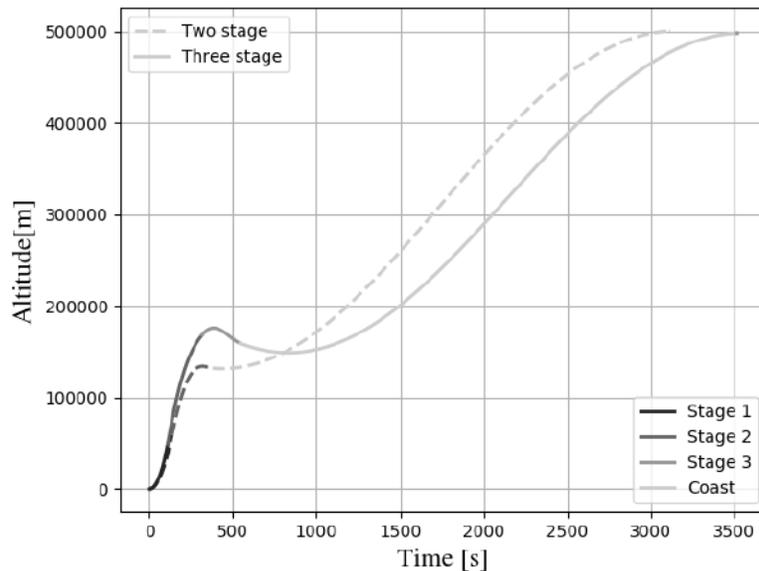


Figure 6.5: Rocket altitude evolution.

The rocket velocity history is illustrated in figure 6.6. Once again, both rockets are able to achieve the required velocity for the circular orbit. Before coast phase, both rockets achieve a minimum velocity of 7900 m/s to reach orbit. The rocket is able to gain sufficient velocity mainly due to turning as soon as possible to conserve energy, which can be verified by observing the flight path angle in figure 6.7. Both rockets have a similar flight path angle evolution, that is maintained slightly above zero during the entire coast phase to allow the rocket ascension.

The two-stage rocket has a faster increase in velocity due to performing staging later when compared to the three-stage rocket, which can be visualized in figure 6.15. Additionally, it can be verified that both rockets perform a last impulse at the end of the coasting phase, crucial to achieve circular orbit

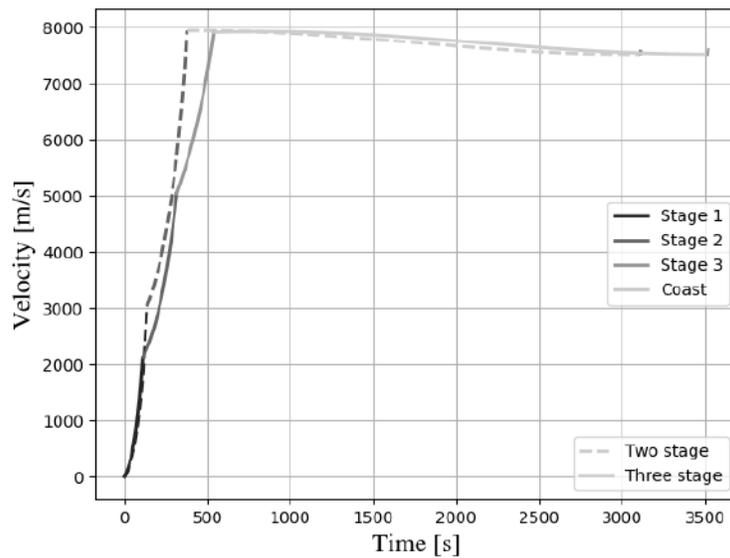


Figure 6.6: Rocket velocity evolution.

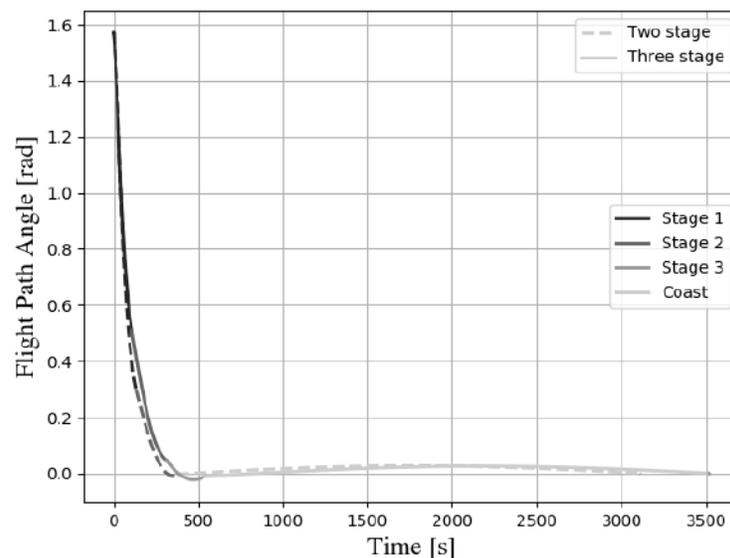


Figure 6.7: Rocket flight path angle evolution.

The thrust vectoring angle evolution is shown in figure 6.8. Contrarily to the Vega thrust vectoring obtained in figure 5.1, both rockets have a thrust vectoring angle below 0.2 rad and, therefore, present a more realistic thrust vectoring. Both controls only start after the fairing jettison, which happens at 188 seconds for the two-stage rocket and at 230 seconds for the three-stage rocket. Thus, the two-stage rocket has a smaller thrust deviation than the three-stage rocket resulting in a lower velocity loss, as shown in figure 6.9.

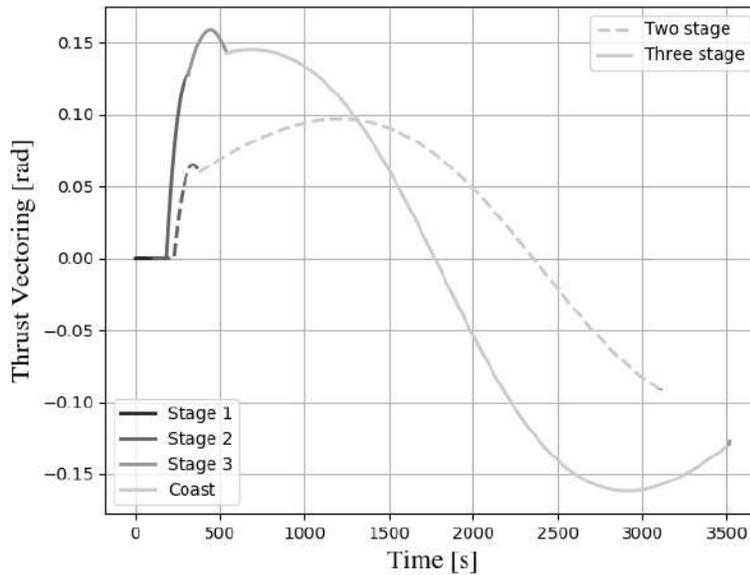


Figure 6.8: Rocket thrust vectoring evolution.

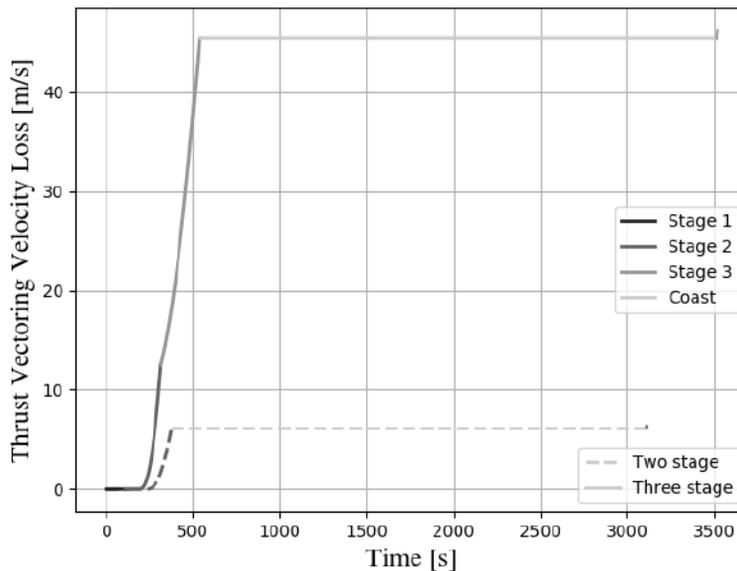


Figure 6.9: Rocket velocity loss due to thrust vectoring.

The remaining velocity losses due to drag and gravity are illustrated in figure 6.10 and figure 6.11, respectively. The velocity loss due to drag is identical for both configurations due to having a similar velocity during atmospheric flight, as seen in figure 6.6. However, relatively to the velocity loss due to gravity, the

three-stage rocket has a loss of 120 m/s larger than the two-stage rocket. This is mainly due to the three-stage rocket atmospheric flight being longer, as it can be observed in figure 6.5. Hence, the optimal control starts later and the rocket is unable to start turning before-hand, leading to a loss increase.

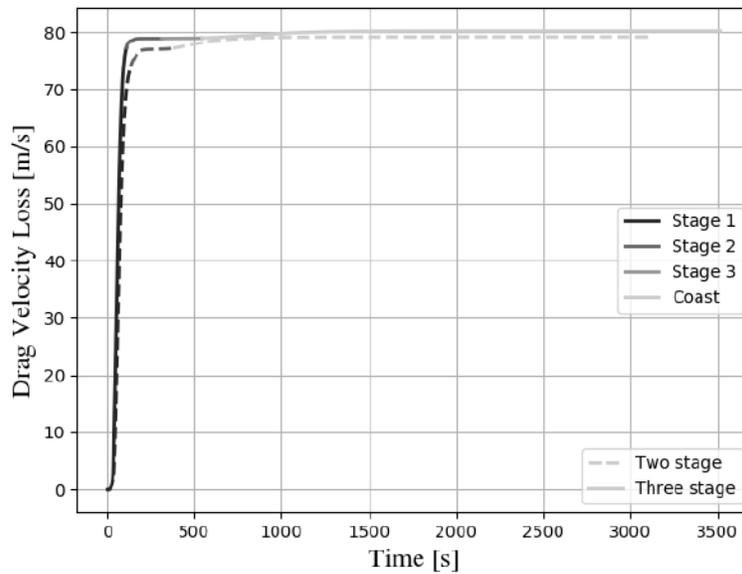


Figure 6.10: Rocket velocity loss due to thrust drag.

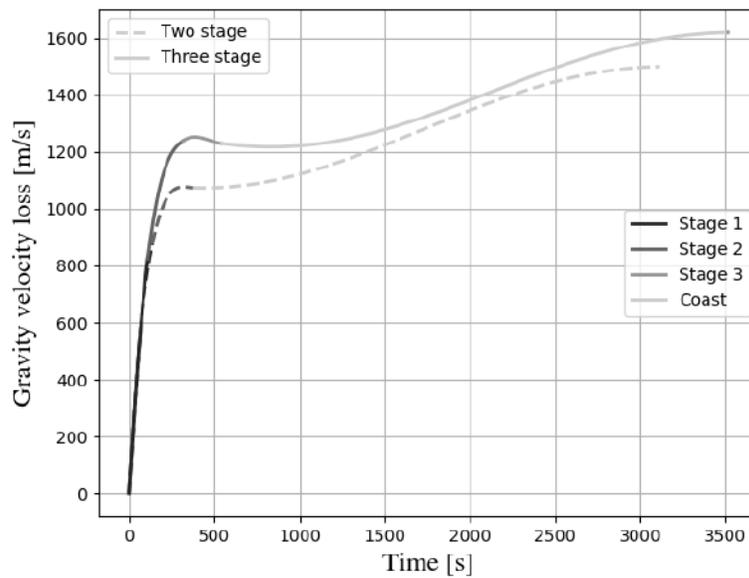


Figure 6.11: Rocket velocity loss due to gravity.

The constraints were successfully handled by the optimizer. The dynamic pressure in figure 6.12 is below the maximum allowable limit. both rockets present a similar dynamic pressure, once again explained by the similar velocities during atmospheric flight. The two-stage and three-stage rocket reach the dynamic pressure peak at 78 and 60 seconds, respectively.

Regarding acceleration, the algorithm constrains the two-stage rocket acceleration, keeping it below  $5g_0$ . To avoid the acceleration from reaching the maximum value, the TWR value could be reduced or more propellant could be allocated to the second stage. However, the algorithm rejects both hypothesis as it would lead to an increase of the rocket total mass.

The axial load for the two-stage rocket and three-stage rocket in figure 6.14 are below the safety load, suggesting the wall thickness could be reduced. Both rockets present identical safety load, as they have a similar wall thickness and diameter. However, the two-stage rocket has a larger axial load applied during the first stage thrusting comparatively to the three-stage rocket, due to having more mass and requiring higher thrust, as it can be observed in figure 6.15. The axial load peak for both rockets coincides with drag peak experienced in the atmosphere.

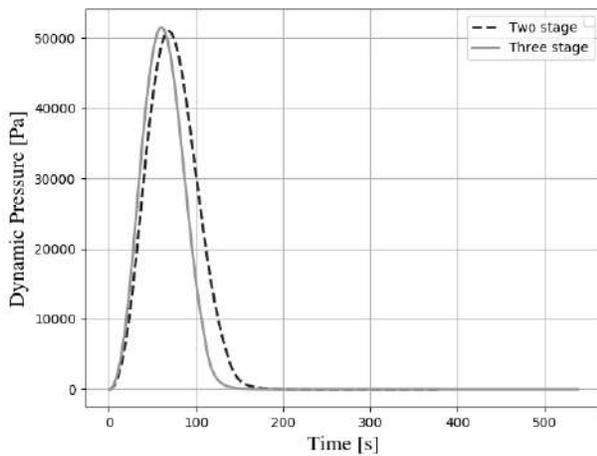


Figure 6.12: Dynamic pressure before coast phase.

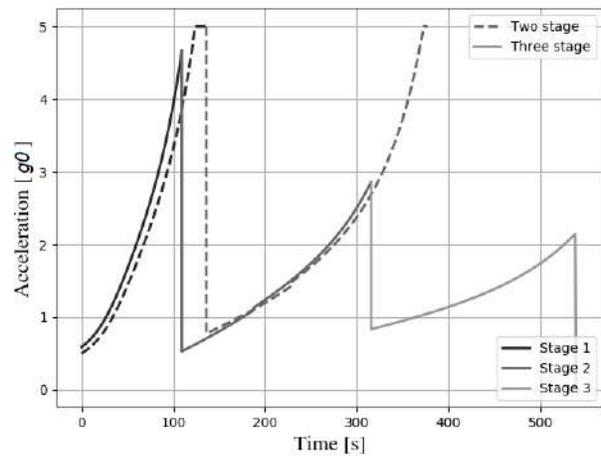


Figure 6.13: Rocket acceleration before coast phase.

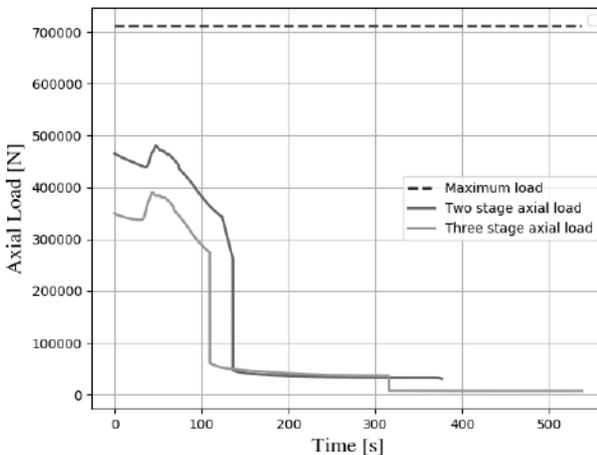


Figure 6.14: Rocket axial load before coast phase.

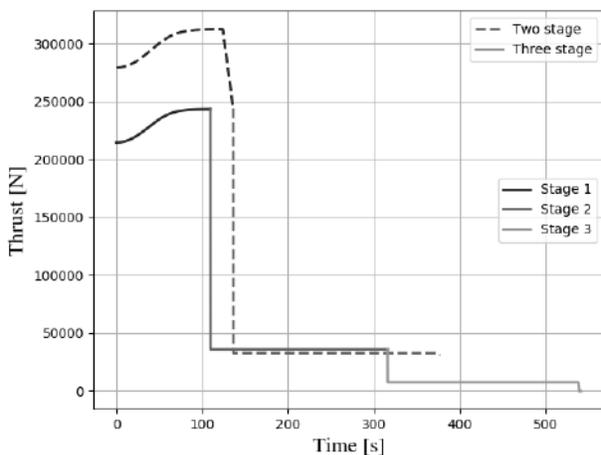


Figure 6.15: Rocket thrust before coast phase.

The rocket reached the specified orbit within the error margin. Observing the velocity losses in figure 6.10, figure 6.11 and figure 6.9, the value is between 1.5 to 1.8 km/s and, therefore, within the expected range.

	Two Stage Rocket	Three Stage Rocket	Electron Rocket
Number of Stages	2	3	2
Total Mass [kg]	18971	13744	12500
Diameter [m]	1.05	1.00	1.2
Length [m]	22.09	19.12	14.5
Number of Engines	1/1	1/1/1	9/1

Table 6.11: Comparison of design characteristics between optimized and Electron rocket.

		Stage 1	Stage 2	Stage 3
Two Stage Rocket	Thrust (vac) [kN]	312	33	—
	Isp [s]	303	333	—
	Propellant	RP-1/LOX	RP-1/LOX	—
Three Stage Rocket	Thrust (vac) [kN]	243	36	8
	Isp [s]	303	333	333
	Propellant	RP-1/LOX	RP-1/LOX	RP-1/LOX
Electron Rocket	Thrust (vac) [kN]	192	22	—
	Isp [s]	303	333	—
	Propellant	RP-1/LOX	RP-1/LOX	—

Table 6.12: Comparison of propulsive characteristics between optimized and Electron rocket.

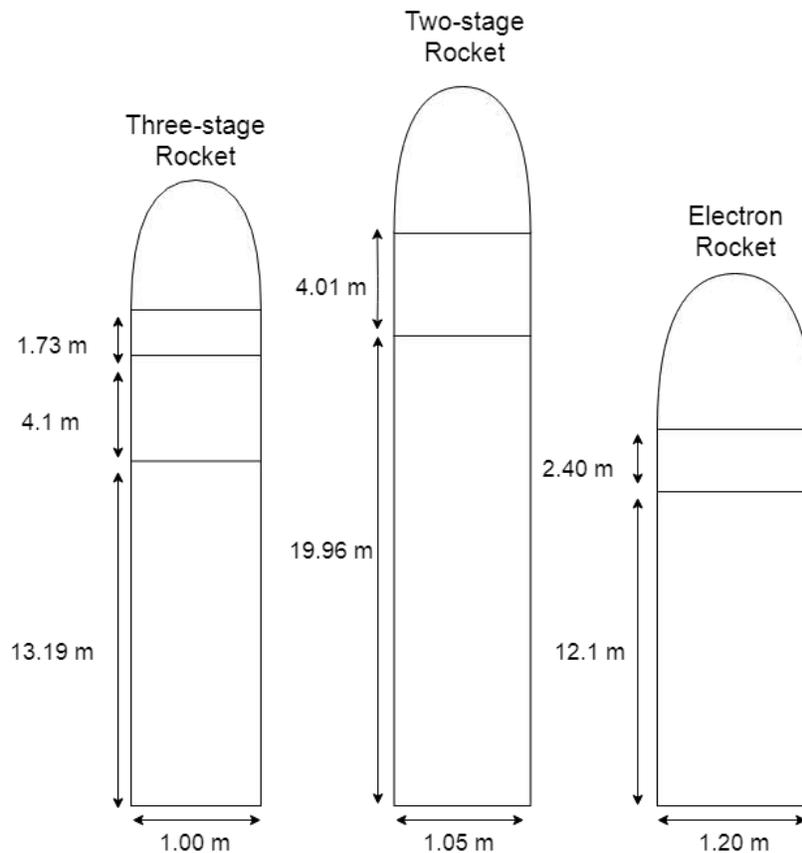


Figure 6.16: Optimized rockets and Electron rocket dimensions.

A comparison between the optimized rockets and the Electron is made in table 6.11 and table 6.12,

providing the design and propulsive characteristics, respectively. Thus, a simple rocket illustration in figure 6.16 allows to visualize the dimensions. Due to the lack of information regarding Electron's real trajectory, the trajectories are not compared.

The two-stage optimized rocket presents a 52% increase in total mass and length relatively to the Electron rocket, while the three-stage optimized rocket presents only a 10% increase in total mass and a 32% increase in length. The increase in length is not only due to requiring more space for the propellant mass, but also because of the smaller diameter. Thus, both optimized rockets have an higher TWR at lift-off than Electron, which has a TWR of 1.57 [111]. The optimized rockets have an higher acceleration during first stage thrusting, and consequently may experience higher aerodynamic loads.

The larger mass value is not only due to the simplifications made in the dry mass models, but also due to the structural and propulsive assumptions (i.e. fixed structural material, engine number, specific impulse and propellant). Current rocket technology is improving at a very fast pace, but the available data is very restricted. Available data and mass heuristics are based on older technology and may provide misleading mass values in rocket design.

The use of a gravity turn in the trajectory model also affects the total mass of the optimal vehicle. Gravity turn does not allow the rocket to have an angle of attack. However, very small angles of attack are admissible, as long as the loads do not compromise the rocket structure, providing a better trajectory and allowing to reduce the rocket mass.

Regardless of the simplifications made, the tool has proven to be able to successfully optimize rocket design and trajectories using a coupled approach and computational parallelization. Thus, the separation of the different disciplines into modules allows to replace them without changing the tool core, allowing the use of higher fidelity models in the future.



# Chapter 7

## Conclusions

### 7.1 Summary and Achievements

In this work a tool capable to perform a rocket preliminary design using a coupled multi-disciplinary optimization approach was developed. Within this framework, a trajectory and staging optimization code were developed separately.

A trajectory model was successfully developed, incorporating both drag and atmospheric models. It is able to find an optimal rocket Pontryagin's Minimum Principle to calculate the optimal control and the PyGMO PSO algorithm to find the optimal trajectory parameters.

The staging optimizer was also able to successfully perform design optimization using the developed GA algorithm to minimize the rocket total mass. It allowed to perform parallel optimization and was able to converge before the generation limit while successfully handling the imposed design constraints. The GA algorithm has also shown promising results for other applications, when compared to DEAP's GA.

The lack of accurate mass and sizing models for solid stages, boosters and fairing led to the creation of new models. The models were tested and allowed to reduce the estimation errors when compared to literature. Oppositely, the liquid stages mass and sizing models were taken from literature.

The tool was tested by performing four steps: mass and sizing model validation, trajectory optimization validation, GA benchmark and parameter tuning, and finally, constraint handling verification by performing a simple design optimization.

The mass and sizing model was tested using Vega, Proton K and Ariane 5 ECA rockets as reference, with the values presented in the user's manual. The results indicate a higher error when calculating the inert mass and length of the top stages, relatively to the bottom stages. Additionally, the liquid stages present more accurate results than the solid stages. The total mass for the three cases presented an error below 2%, and were considered reasonable for the preliminary design. The thickness value used in the simulations were arbitrary, within imposed boundaries, due to the difficulty of finding the data.

Posteriorly, the trajectory optimization was tested using a Vega rocket and a reference mission in the user's manual. The influence of coast during flight was analyzed and its inclusion allowed to save over 40% of propellant of the last stage. Both tested trajectories reached mission orbit

The developed GA was benchmarked and compared with DEAP's GA. The developed GA was comparatively better to the DEAP's GA when the function presented several local minima, and presented similar results for the remaining functions. The step-size and mutation rate were tuned, enhancing convergence speed. Thus, it was verified that the population using 150 individuals converged faster and obtained the rocket with less mass and all the imposed constraints were respected.

The tool is finally tested by performing a small launch vehicle conceptual design. Two optimizations were performed, regarding a two-stage and a three-stage rocket, to carry a 150 kg payload to a sun synchronous orbit at 500 km of altitude. Both designed rockets are able to perform the mission, and the three-stage rocket presents less mass than the two-stage, as expected. Comparatively to the Electron, the three-stage optimized rocket has 10% more mass. The mass and sizing errors are due to the assumptions made, inaccuracy of the mass and sizing models and the use of gravity turn in trajectory.

Nevertheless, the tool is able to perform conceptual rocket design and trajectory optimization, parallelizing the task using a master-slave architecture. The models used by the tool can be replaced independently from the other models to improve the tool in the future.

## 7.2 Future Developments

Although the optimization tool was successfully developed, the simplistic models used may provide results with significant errors. A clear path to future work is the creation of better heuristic mass and sizing models. The models used in the tool are rather simplistic and can be improved by increasing the number of parameters (e.g. propellant tank pressure and combustion temperature) and/or by using more recent data.

The trajectory model can also be improved with the inclusion of throttling as a control variable. The tool only optimizes the thrust vectoring and the inclusion of throttling can lead to a propellant saving increase. Thus, a better trajectory than the gravity turn can be used, allowing the rocket to have a slight angle of attack during ascent.

To simulate real-world rocket trajectories, a trajectory model with six degrees of freedom and inclusion of wind needs to be developed and integrated in the tool. Thus, to better modelling the drag force, software like missile DATCOM can be integrated directly into the tool.

Convergence speed and results can also be enhanced with better parameter tuning of the genetic algorithm and the particle swarm optimization. This can be done by performing numerous rocket and trajectory optimization, although it requires high computational power.

# Bibliography

- [1] M. Cerf, "Versatile launcher preliminary design," *Journal of Spacecraft and Rockets*, vol. 56, pp. 517–525, Mar. 2019.
- [2] "First flight of 163 foot Saturn space vehicle (SA-1) from Cape Canaveral.," 1961. <https://www.floridamemory.com/items/show/8764> (visited: 06-05-2019).
- [3] R. S. Ryan and J. S. Townsend, "Fundamentals and issues in launch vehicle design," *Journal of Spacecraft and Rockets*, vol. 34, no. 2, pp. 192–198, 1997.
- [4] J. C. Blair, R. S. Ryan, L. A. Schutzenhofer, and W. R. Humphries, *Launch Vehicle Design Process: Characterization, Technical Integration, and Lessons Learned*. National Aeronautics and Space Administration (NASA), 2001.
- [5] M.-u.-D. Qazi and H. Linshu, "Nearly-orthogonal sampling and neural network metamodel driven conceptual design of multistage space launch vehicle," *Comput. Aided Des.*, vol. 38, pp. 595–607, June 2006.
- [6] J. Martins and A. Lambe, "Multidisciplinary design optimization: A survey of architectures," *AIAA Journal*, vol. 51, pp. 2049–2075, Sept. 2013.
- [7] L. F. Rowell and J. J. Korte, *Launch vehicle design and optimization methods and priority for the advanced engineering environment*. National Aeronautics and Space Administration (NASA), 2003.
- [8] Space Exploration Technologies , *Falcon User's Guide*, Jan. 2019.
- [9] N. Duranté, A. Dufour, V. Pain, G. Baudrillard, and M. Schoenauer, *Multi-disciplinary Analysis and Optimisation Approach for the Design of Expendable Launchers*. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Aug. 2004.
- [10] A. F. Rafique, L. He, A. Kamran, and Q. Zeeshan, "Hyper heuristic approach for design and optimization of satellite launch vehicle," *Chinese Journal of Aeronautics*, vol. 24, pp. 150 – 163, Apr. 2011.
- [11] T. Tsuchiya and T. Mori, "Optimal conceptual design of two-stage reusable rocket vehicles including trajectory optimization," *Journal of Spacecraft and Rockets*, vol. 41, pp. 770–778, Sept. 2004.

- [12] M. Balesdent, *Multidisciplinary Design Optimization of Launch Vehicles*. PhD thesis, Ecole Centrale de Nantes (ECN), Nov. 2011.
- [13] R. D. Braun and I. M. Kroo, "Development and application of the collaborative optimization architecture in a multidisciplinary design environment," *Multidisciplinary Design Optimization: State of the Art*, vol. 80, pp. 98–116, Aug. 1996.
- [14] J. Sobieszczanski-Sobieski, J. Agte, and J. Robert Sandusky, *Bi-level integrated system synthesis (BLISS)*. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Feb. 1998.
- [15] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 303–317, June 2005.
- [16] W. Yao, X. Chen, W. Luo, M. Van tooren, and J. Guo, "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles," *Progress in Aerospace Sciences*, vol. 47, pp. 450–479, Aug. 2011.
- [17] F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, "Special section on multidisciplinary design optimization: Metamodeling in multidisciplinary design optimization: How far have we really come?," *AIAA Journal*, vol. 52, pp. 670–690, Apr. 2014.
- [18] M. Qazi and H. Linshu, "Nearly-orthogonal sampling and neural network metamodel driven conceptual design of multistage space launch vehicle," *Computer-Aided Design*, vol. 38, pp. 595–607, June 2006.
- [19] M. Qazi, H. Linshu, and P. Mateen, "Hammersley sampling and support-vector-regression-driven launch vehicle design," *Journal of Spacecraft and Rockets*, vol. 44, pp. 1094–1106, Sept. 2007.
- [20] T. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Comparison of response surface and kriging models for multidisciplinary design optimization," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Sept. 1998.
- [21] A. Arias-Montano, C. A. Coello Coello, and E. Mezura-Montes, "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE Transactions on Evolutionary Computation*, vol. 16, pp. 662–694, Oct. 2012.
- [22] D. J. Bayley, R. J. Hartfield, J. E. Burkhalter, and R. M. Jenkins, "Design optimization of a space launch vehicle using a genetic algorithm," *Journal of Spacecraft and Rockets*, vol. 45, July 2008.
- [23] T. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA Journal*, vol. 39, pp. 2233–2241, Dec. 2001.

- [24] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 193–207, Mar. 1998.
- [25] I. M. Ross and F. Fahroo, "Pseudospectral knotting methods for solving nonsmooth optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 27, pp. 397–405, May 2004.
- [26] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, pp. 338–342, July 1987.
- [27] M. Pontani, "Particle swarm optimization of ascent trajectories of multistage launch vehicles," *Acta Astronautica*, vol. 94, p. 852–864, Feb. 2014.
- [28] R. Bilstein, U. N. Aeronautics, S. Administration, N. Aeronautics, S. A. Staff, U. S. N. Aeronautics, and S. A. H. Office, *Stages to Saturn: A Technological History of the Apollo/Saturn Launch Vehicles*. NASA SP, National Aeronautics and Space Administration, NASA History Office, 1996.
- [29] U. S. C. B. Office, *Alternatives for future United States space launch capabilities*. A CBO study, Congress of the U.S., Congressional Budget Office, 2006.
- [30] A. Tewari, *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB® and Simulink®*. Modeling and Simulation in Science, Engineering and Technology, Birkhäuser Boston, 2007.
- [31] M. Turner, *Rocket and Spacecraft Propulsion: Principles, Practice and New Developments*. Springer Praxis Books, Springer Berlin Heidelberg, 2008.
- [32] A. Darrin and B. O'Leary, *Handbook of Space Engineering, Archaeology, and Heritage*. Advances in Engineering Series, CRC Press, 2009.
- [33] N. Sarigul-Klijn and M. Sarigul-Klijn, "Air-launching earth to orbit: Effects of launch conditions and vehicle aerodynamics," *Journal of Spacecraft and Rockets*, vol. 42, pp. 569–575, May 2005.
- [34] P. Sforza, *Theory of Aerospace Propulsion*. Aerospace Engineering, Elsevier Science, 2011.
- [35] E. Civek, *Multistage Launch Vehicle Design with Thrust Profile and Trajectory Optimization*. PhD thesis, Middle East Technical University, Sept. 2014.
- [36] E. Kyle, "Delta II Data Sheet." <http://www.spacelaunchreport.com/delta2.html>, Oct 2018. Accessed: 14-03-2019.
- [37] E. Kyle, "Atlas V Data Sheet." <https://www.spacelaunchreport.com/atlas5.html>, Oct 2018. Accessed: 14-03-2019.
- [38] G. Sutton and O. Biblarz, *Rocket Propulsion Elements*. Knovel Library, Wiley, 2016.

- [39] Arianespace, *Vega C User's Manual*, May 2018.
- [40] International Launch Services, *Proton Launch System Mission Planner's Guide*, 2009.
- [41] Arianespace, *Ariane V User's Manual*, Oct. 2016.
- [42] P. F. Gath and A. J. Calise, "Optimization of launch vehicle ascent trajectories with path constraints and coast arcs," *Journal of Guidance, Control, and Dynamics*, vol. 24, pp. 296–304, Mar. 2001.
- [43] S. Byung-Chan, Y.-K. Park, W.-R. Roh, and G.-R. Cho, "Attitude controller design and test of korea space launch vehicle-i upper stage," *International Journal of Aeronautical and Space Sciences*, vol. 11, pp. 303–312, Dec. 2010.
- [44] P. Lu and B. Pan, "Highly constrained optimal launch ascent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 33, pp. 404–414, Mar. 2010.
- [45] M. Griffin and J. French, *Space Vehicle Design*. AIAA education series, American Institute of Aeronautics & Astronautics, 2004.
- [46] L. Dr. M. A. Sharaf, "Computational algorithm for gravity turn maneuver," *Global Journal of Science Frontier Research*, vol. 12, no. 13-F, 2012.
- [47] P. Lu, B. Griffin, G. Dukeman, and F. R. Chavez, "Rapid optimal multi-burn ascent planning and guidance," *Journal of Guidance Control and Dynamics*, vol. 31, pp. 1656–1664, Nov. 2008.
- [48] K. S. Bernstein, "Structural design requirements and factors of safety for spaceflight hardware," tech. rep., National Aeronautics and Space Administration (NASA), Feb. 2011.
- [49] B. Bhat, *Aerospace Materials and Applications*. Progress in astronautics and aeronautics, American Institute of Aeronautics and Astronautics, Incorporated, 2018.
- [50] J. D. Kemp and R. L. Clark, "Noise reduction in a launch vehicle fairing using actively tuned loudspeakers," *The Journal of the Acoustical Society of America*, vol. 113, no. 4, pp. 1986–1994, 2003.
- [51] United Launch Alliance, LLC, "Delta II 732x-10 expanded view." [https://www.ulalaunch.com/images/default-source/default-album/deltaiischema1642606f2258d4fc4adc0c96211e5dbd5.jpg?sfvrsn=4e4beec7\\_2](https://www.ulalaunch.com/images/default-source/default-album/deltaiischema1642606f2258d4fc4adc0c96211e5dbd5.jpg?sfvrsn=4e4beec7_2), 2007. [Online; accessed 15-May-2019].
- [52] W. Tjardus Koiter, *The Stability of Elastic Equilibrium*. PhD thesis, Technische Hooge School, Feb. 1970.
- [53] P. Dépincé, B. Guédas, and J. Picard, *Multidisciplinary and multiobjective optimization: Comparison of several methods*. May 2007.

- [54] G. Fasano and J. D. Pintr, *Space Engineering: Modeling and Optimization with Case Studies*. Springer Publishing Company, Incorporated, 1st ed., 2017.
- [55] S. Kodiyalam, "Evaluation of methods for multidisciplinary design optimization (mdo)," 10 1998.
- [56] R. Perez, H. H. T. Liu, and K. Behdinan, "Evaluation of multidisciplinary optimization approaches for aircraft conceptual design," *Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, vol. 4, Aug. 2004.
- [57] R. Michael Lewis, G. R. Shubin, E. J. Cramer, J. Dennis, P. D. Frank, R. Michael, L. Gregory, and R. Shubin, "Problem formulation for multidisciplinary optimization," *SIAM Journal on Optimization*, vol. 4, Feb. 1997.
- [58] R. Balling and J. Sobieszczanski-Sobieski, "Optimization of coupled systems: A critical overview of approaches," *AIAA Journal*, vol. 34, Jan. 1995.
- [59] R. Sellar, S. Batill, and J. Renaud, *Response surface based, concurrent subspace optimization for multidisciplinary system design*. 34th Aerospace Sciences Meeting and Exhibit, Jan. 1996.
- [60] H. Kim, D. Rideout, P. Papalambros, and J. Stein, "Analytical target cascading in automotive vehicle design," *Journal of Mechanical Design*, vol. 125, Oct. 2003.
- [61] J. Sobieszczanski-Sobieski, A. Morris, and M. Van tooren, *Multidisciplinary Design Optimization Supported by Knowledge Based Engineering*. John Wiley & Sons, 2017.
- [62] D. Zingg, M. Nemeč, and T. Pulliam, "A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization," *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 17, Mar. 2008.
- [63] J. Martins, J. Alonso, and J. J. Reuther, "A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design: Special issue on multidisciplinary design optimization (guest editor: Natalia alexandrov)," *Optimization and Engineering*, vol. 6, Mar. 2005.
- [64] R. Bulirsch, E. Nerz, H. J. Pesch, and O. von Stryk, *Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider*, pp. 273–288. *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*, Basel: Birkhäuser Basel, 1993.
- [65] L. Rios and N. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, Nov. 2009.
- [66] D. M. Olsson and L. S. Nelson, "The nelder-meade simplex procedure for function minimization," *Technometrics*, vol. 17, no. 1, pp. 45–51, 1975.

- [67] J. Gablonsky and C. Kelley, "A locally-biased form of the direct algorithm," *Journal of Global Optimization*, vol. 21, pp. 27–37, Jan. 2001.
- [68] O. Kramer, D. E. Ciaurri, and S. Koziel, *Derivative-Free Optimization*, pp. 61–83. Computational Optimization, Methods and Algorithms, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [69] S. Kirkpatrick, C. D. Jr. Gelatt, and M. P. Jr. Vecchi, "Optimization by simulated annealing," *Science (New York, N.Y.)*, vol. 220, pp. 671–80, June 1983.
- [70] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Oct 1995.
- [71] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182 – 197, May 2002.
- [72] E. K. Burke, J. P. Newall, and R. F. Weare, "Initialization strategies and diversity in evolutionary timetabling," *Evol. Comput.*, vol. 6, pp. 81–103, Mar. 1998.
- [73] O. Yeniay, "Penalty function methods for constrained optimization with genetic algorithms," *Mathematical and Computational Applications*, vol. 10, pp. 45–56, Apr. 2005.
- [74] T. Vinkó and D. Izzo, "Global optimisation heuristics and test problems for preliminary spacecraft trajectory design," *GOHTPPSTD European Space Agency, the Advanced Concepts Team.*, 2008.
- [75] R. Sivaraj and T. Ravichandran, "A review of selection methods in genetic algorithm," *International Journal of Engineering Science and Technology*, vol. 3, Mar. 2011.
- [76] R. Chelouah and P. Siarry, "A continuous genetic algorithm designed for the global optimization of multimodal functions," *Journal of Heuristics*, vol. 6, pp. 191–213, June 2000.
- [77] P. L. Schoonover, W. Crossley, and S. D. Heister, "Application of a genetic algorithm to the optimization of hybrid rockets," *Journal of Spacecraft and Rockets*, vol. 37, pp. 622–629, 2000.
- [78] Y. Shi and E. RC, "A modified particle swarm optimizer," *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, vol. 6, pp. 69 – 73, June 1998.
- [79] J. Bansal, P. Singh, M. Saraswat, A. Verma, S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," *Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011*, pp. 633–640, Oct. 2011.
- [80] B. A. Conway, *The Problem of Spacecraft Trajectory Optimization*, p. 1–15. Cambridge Aerospace Series, Cambridge University Press, 2010.

- [81] L. Ma, K. Wang, Z. Shao, Z. Song, and L. T. Biegler, "Direct trajectory optimization framework for vertical takeoff and vertical landing reusable rockets: case study of two-stage rockets," *Engineering Optimization*, vol. 51, no. 4, pp. 627–645, 2019.
- [82] D. Kraft, *On Converting Optimal Control Problems into Nonlinear Programming Problems*, vol. F15 of *Computational Mathematical Programming*, pp. 261–280. Jan. 1985.
- [83] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, pp. 849–904, Jan. 2017.
- [84] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, pp. 357–373, Jan. 1992.
- [85] L. Pontryagin, *Mathematical Theory of Optimal Processes*. Classics of Soviet Mathematics, Taylor & Francis, 1987.
- [86] R. Humble, *Space Propulsion Analysis and Design*. McGraw-Hill Companies, Incorporated, 1995.
- [87] F. Castellini, *Multidisciplinary design optimization for expendable launch vehicles*, PhD thesis. PhD thesis, Politecnico di Milano, Nov. 2012.
- [88] D. L. Akin, "Mass estimating relationships - principles of space systems designs," *University Lecture*, 2005. University of Maryland.
- [89] B. Zandbergen, "Simple mass and size estimation relationships of pump fed rocket engines for launch vehicle conceptual design," 6th European Conference for Aeronautics and Space Sciences (EU-CASS), June 2015.
- [90] C. Frank, O. Pinon, C. Tyl, and D. Mavris, "New design framework for performance, weight, and life-cycle cost estimation of rocket engines," 6th European Conference for Aeronautics and Space Sciences (EUCASS), June 2015.
- [91] S. Greenland, S. Senn, K. J. Rothman, J. B. Carlin, C. Poole, S. Goodman, and D. Altman, "Statistical tests, p values, confidence intervals, and power: A guide to misinterpretations," *European Journal of Epidemiology*, vol. 31, May 2016.
- [92] R. R. Rohrschneider, *Development of a Mass Estimating Relationship Database for Launch Vehicle Conceptual Design*. PhD thesis, Georgia Institute of Technology, Apr. 2002.
- [93] H. Curtis, *Orbital Mechanics: For Engineering Students*. Aerospace Engineering, Elsevier Science, 2015.
- [94] M. Reilly and N. R. L. W. DC., *Equations of Powered Rocket Ascent and Orbit Trajectory*. Defense Technical Information Center, May 1979.

- [95] A. Tewari, *Advanced Control of Aircraft, Spacecraft and Rockets*. Aerospace Series, Wiley, 2011.
- [96] E. Betts and R. Frederick, "A historical systems study of liquid rocket engine throttling capabilities," *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, July 2010.
- [97] U. S. C. on Extension to the Standard Atmosphere, *U.S. standard atmosphere, 1976*. National Oceanic and Atmospheric Administration : for sale by the Supt. of Docs., U.S. Govt. Print. Off., 1976.
- [98] P. A. Ritter, "Optimization and design for heavy lift launch vehicles," Master's thesis, University of Tennessee, 2012.
- [99] D. Kliche, C. Mundt, and E. H. Hirschel, "The hypersonic mach number independence principle in the case of viscous flow," *Shock Waves*, vol. 21, pp. 307–314, Aug. 2011.
- [100] P. Chambre and S. Schaaf, *Flow of Rarefied Gases*. Princeton Legacy Library, Princeton University Press, 2017.
- [101] D. I. Francesco Biscani and M. Märtens, "Esa/pagmo2: Pagmo 2.10." <https://doi.org/10.5281/zenodo.2529931>, Jan. 2019.
- [102] B. G. M. Husslage, G. Rennen, E. R. van Dam, and D. den Hertog, "Space-filling latin hypercube designs for computer experiments," *Optimization and Engineering*, vol. 12, pp. 611–630, Feb. 2006.
- [103] W. Haeussermann, "Saturn launch vehicle's navigation guidance, and control system," *Automatica*, vol. 7, pp. 537–556, Sept. 1971.
- [104] J. Meseguer, I. Pérez-Grande, and A. Sanz-Andrés, *Spacecraft Thermal Control*. Woodhead Publishing in mechanical engineering, Elsevier Science, 2012.
- [105] Avio, "Vega stage characteristics." <http://www.avio.com/en/vega/vega/>. Accessed: 15-05-2019.
- [106] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, pp. 1945–1950, July 1999.
- [107] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, July 2012.
- [108] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 124–141, July 1999.
- [109] R. Haupt and S. Haupt, *Practical Genetic Algorithms*. Wiley InterScience electronic collection, Wiley, 2004.

[110] T. Wekerle, J. Bezerra Pessoa Filho, L. Loures, and L. G. Trabasso, "Status and trends of smallsats and their launch vehicles — an up-to-date review," *Journal of Aerospace Technology and Management*, vol. 9, pp. 269–286, Aug. 2017.

[111] Rocket Lab , *Payload User's Guide*, Apr. 2019.



# Appendices



# Appendix A

## Regression Data

Several tables containing the data used for the regressions made in section 4.1 are shown. Table A.1 is a compilation of solid engines characteristic ordered by dry mass.

Number	Motor	Dry Mass [kg]	Propellant Mass [kg]	Thrust [kN]
1	ASAS 21-85V	96	655	62.3
2	Star 31	108	1286	82.3
3	Orion 38	121	771	32.2
4	ASAS 21-120	130	924	99.2
5	Star 48 B	131	2010	68.9
6	ASAS 32-58 V	146	1041	106.3
7	Orion 32	200	1941	128.1
8	ASAS 28-185	331	2800	231.8
9	Orion 50	344	3025	114.6
10	Orion 50 XL	395	3924	160.6
11	ORBUS 21	667	9707	199.8
12	Orion 50S XL	1150	15023	626.3
13	Orion 50S XLT	1157	15023	614.9
14	Orion 50S XLG	1179	15023	588
15	Castor 30	1224	12837	258.9
16	Orion 50S	1243	12162	465.1
17	Orion 50ST	1249	12157	454.4
18	GEM 40 VN	1327	11775	478.8
19	GEM 46 VN	2275	16865	601.4
20	Castor 120	4072	49005	1685.9

Table A.1: Solid motor characteristics for regression [90]

Table A.2 gives the stage characteristics of three solid rockets: Vega, Minotaur IV and Minotaur I. Thus, several solid boosters are also presented.

<b>Stage/Booster</b>	<b>Total Mass [kg]</b>	<b>Propellant Mass [kg]</b>	<b>Thrust [kN]</b>	<b>Diameter [m]</b>	<b>Length [m]</b>
Vega stage 1	8533	87710	3000	3	11.7
Vega stage 2	2486	23814	1120	1.9	8.39
Vega stage 3	1433	10567	317	1.9	4.12
Minotaur IV stage 1	3600	45400	2224	2.34	8.4
Minotaur IV stage 2	3200	24500	1223	2.34	7.9
Minotaur IV Stage 3	650	7080	289	2.34	2.44
Minotaur I stage 1	2292	20785	792	1.67	7.49
Minotaur I stage 2	795	6237	267.7	1.33	4.12
Minotaur I stage 3	391	3915	170	1.28	3.07
Ariane V Booster	31000	237000	6470	3.05	31.6
PSLV Booster	30200	138000	4860	2.8	20.34
H-IIA Booster	11360	65040	2260	2.5	15.1
Atlas V Booster	5470	40957	1688	1.58	17
Delta IV Booster	3952	29698	1236	1.52	13.2
Delta II Booster	1102	11766	643	1.03	11.05

Table A.2: Solid stages and boosters characteristics for regression

Finally, the data concerning the rocket fairing are expressed in table A.3.

<b>Number</b>	<b>Rocket Fairing</b>	<b>Mass [kg]</b>	<b>Maximum Payload [kg]</b>	<b>Length [m]</b>	<b>Diameter [m]</b>
1	Angara A7	4500	35000	25	5.5
2	Antares 200	970	7000	9.87	3.94
3	Ariane 5 ECA	2500	21000	17	5.4
4	Atlas V 421	2127	13600	12	4.2
5	Atlas V 551	3524	18500	20.7	5.4
6	Delta II	880	4850	8.49	2.9
7	Electron	50	225	2.5	1.2
8	Vega	540	1500	7.88	2.6
9	Falcon 9	1750	13150	13.1	5.2
10	Minotaur C	300	1458	2.34	1.33

Table A.3: Fairing characteristics for regression