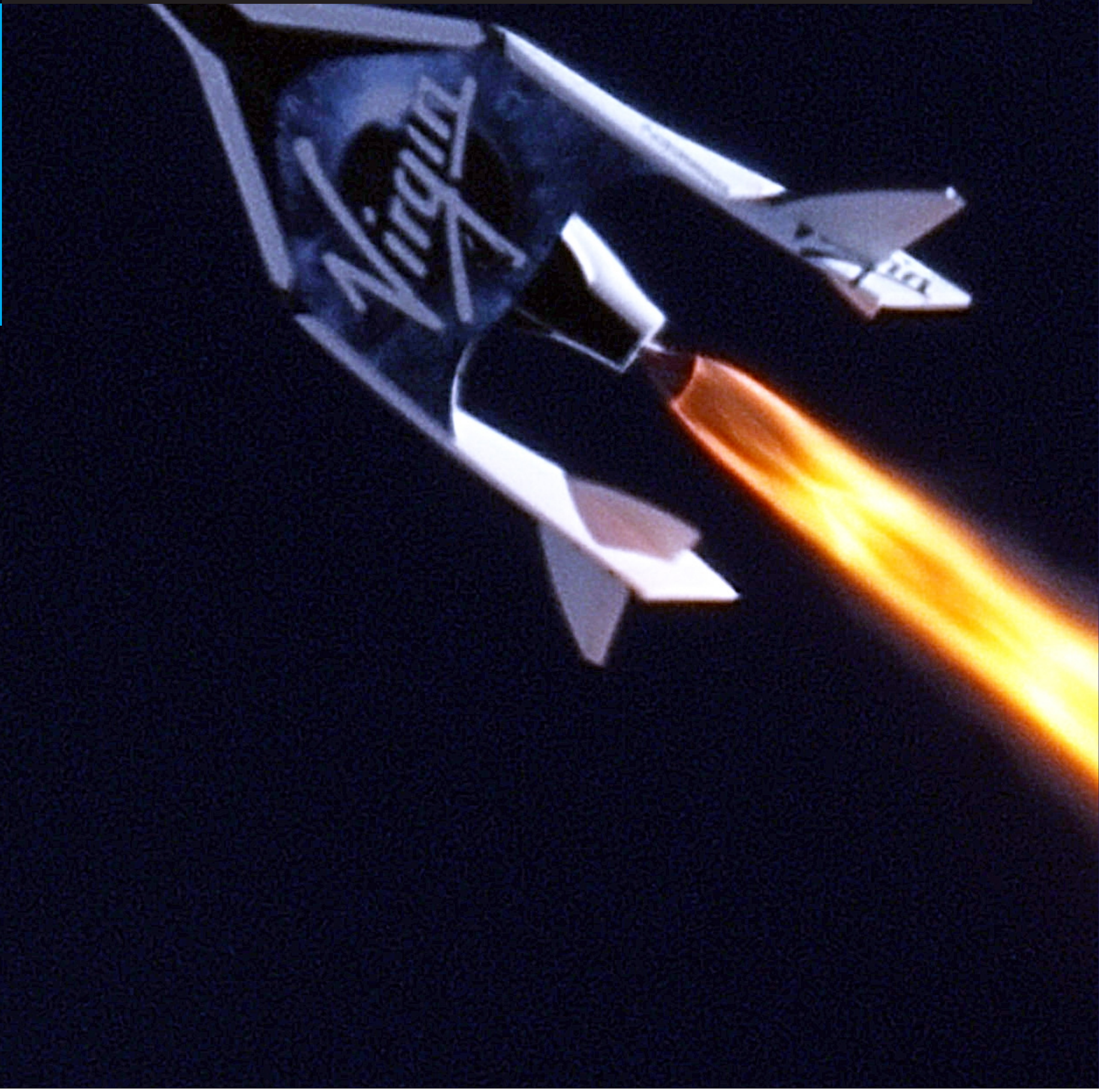
# Design Optimization of Ground and Air-Launched Hybrid Rockets

## Development of an Optimisation Tool for Multi-Technology Rocket Launch Vehicle Design

Francisco Miranda

Technische Universiteit Delft



TU Delft Delft University of Technology

Challenge the future

# Design Optimization of Ground and Air-Launched Hybrid Rockets

## Development of an Optimisation Tool for Multi-Technology Rocket Launch Vehicle Design

by

**Francisco Miranda**

in partial fulfilment of the requirements for the degree of

**Master of Science**
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Tuesday February 27, 2015 at 10:00 AM.

| | | |
|---|---|---|
| Course Code: | AE5810 | |
| Supervisor: | Prof. dr. ir. E.K.A. Gill | TU Delft |
| Thesis committee: | Ir. B. T. C. Zandbergen, | TU Delft |
| | D. Dirkx, | TU Delft |
| | M.C. Naeije, | TU Delft |
| | A. van Kleef | NLR |

TU Delft Delft University of Technology

# Abstract

Access to space is now more important than ever. For the last few years, smaller and smaller satellites are acquiring the capacity to do what was previously only possible with large platforms. Currently, most small countries can build a small satellite, but most cannot afford to launch one by themselves.

Air launching rocket vehicles has been touted as a possible solution to reduce the cost of both the launch vehicle and its operation, especially for the launch of small payloads. Most research in the field looks into solid rockets, here, however, hybrids are investigated. Hybrid rocket vehicles work by burning solid fuel with liquid oxidizer. Unlike solid rockets, hybrids can be re-ignited, are safer, and have a potentially higher orbital injection accuracy and specific impulse.

To study the feasibility and performance of such a solution, a Multidisciplinary Design Optimization tool was used. The tool is an improvement upon a previous software package that only supported solid rocket technology. It addresses trajectory computation, estimation of vehicle dimensions and propulsion performance and launch simulation. The tool was developed in $C^{++}$, and is now capable of handling multi-stage solid-only rockets, hybrid-only rockets and mixed configurations. It consists of performance and dimension estimators for individual rocket stages, these are integrated into a launch vehicle whose launch is simulated and analysed.

The tool has been used to optimize the configuration of a rocket vehicle designed to attain a circular orbit (eccentricity $< 0.01$) at 780 km altitude. By iterating on the design using an optimization process known as Particle Swarm Optimization, it attempts to reduce the take-off weight of its solutions. The tests allow one to compare hybrid propellant rockets to solid ones in both air and ground launched scenarios. Based on the assumptions made, it was found that air-launch reduces GTOW by about 70 % when compared to an equivalent ground launch. It is also found that, in general, hybrid rockets have a GTOW similar to, or higher than, their solid counterparts. The research also tackles whether mixed configurations (with both solid and hybrid stages) allow one technology to strengthen the other. Results are also shown on the launch performance of existing engines, to both validate the models and offer insight to manufacturers into how their vehicles will perform once launched.

# Preface

Humanity achieved access to space nearly 60 years ago. Yet, accessing space for commercial, personal or other motives, is no more democratic today than it was then. Getting into orbit today, like in 1957 - when Sputnik was launched - is as hard as the amount of mass you want to get there. Unlike in 1957, today very small satellites and spacecraft are capable of doing very large amounts of science. An easy, cheap access to space could revolutionize a number of industries, get us to learn more about the planet and ourselves, and is thus a noble goal. The market for dedicated launch vehicles for small satellites is just now emerging. Now more than ever there is the possibility for any company or government to access and benefit from space. Making cheap reliable launch solutions is then a current necessity. This work explores the use of Hybrid Rocket technology and Air-launching to achieve these goals. Combining a reliable, safe and potential out-performing technology like Hybrid propulsion with the benefits of an air-launch system holds promise in getting small payloads to orbit. This thesis finds that the technology has the ability to compete and maybe even surpass the alternative Solid and Liquid technologies. Aided by an air-launch, the launching of a rocket vehicle from an aircraft, the benefits are even greater. Parallel to this research this project has another goal: to develop a comprehensive launch vehicle design optimization tool. This is, hopefully, the start of a project that can be continued by other students to add and improve on this software. Making the capabilities that I and those before me developed, more and more powerful allowing for better and more exciting modelling to take place.

*It takes sixty-five thousand errors before you are qualified to make a rocket.*

- Wernher von Braun

*To infinity and beyond!*

- Buzz Lightyear

## Acknowledgements

This project was the culmination of my education in Delft. This MSc and now this project were one of the most important steps in my life. Delft has been a great experience for me, and a part of my life I will never forget. The project in particular was the phase of my time here where I learned the most and the most satisfying one. I would like to thank Barry Zandbergen and Dominic Dirkx who supervised me throughout this project and had the patience to sit with me and answer all my questions any time I needed. Also to Nammo and DARE for the data and insight provided that made this research possible. All my friends and colleagues at the Spaceflight Department: Anna, Eneko, Guillaume, Martijn, Pablo and Tiemen who contributed with their valuable inputs in all stages of this project and with their support. Especially during the thesis' months, I had the support and friendship of my Dutch friends and housemates: Fokke, Jasper, Lucas, Maarten, Thijs and Yorick, and more recently Douwe and Rolf, who supported me and made sure I enjoyed Holland and had fun more often than I should. To them I'd like to say "dank je wel". Finally I would like to thank my parents and my brother for their unconditional support and for making it possible for me to come to The Netherlands and do all of this.

*Francisco Miranda*
*Delft, January 2015*

# Contents

# List of Symbols and Acronyms

## Roman Symbols

| | |
|---|---|
| $a$ | Semi-major axis (m) |
| $A_e$ | Nozzle Exit Area (m$^2$) |
| $A_f$ | Fuel filled cross-sectional area (m$^2$) |
| $a_o$ | Regression rate coefficient (-) |
| $A_p$ | Port area (m$^2$) |
| $A_t$ | Nozzle throat area (m$^2$) |
| $A_t$ | Throat Area (m$^2$) |
| $a_{thrust}$ | Acceleration induced by the thrust force (m$^{-2}$) |
| $c^*$ | Characteristic velocity (m s$^{-1}$) |
| $C_F$ | Thrust Coefficient (-) |
| $D_o$ | Outer diameter of the fuel grain (m) |
| $D_t$ | Nozzle throat diameter (m) |
| $D_{fill}$ | Diameter of cylindrical or spherical form of the fluid filling the oxidizer tank (m) |
| $D_{fuel}$ | Diameter of the Fuel Tank (m) |
| $DF$ | Design Factor (-) |
| $e$ | Eccentricity (-) |
| $e_{ratio}$ | Expansion ratio (-) |
| $f$ | Fitness function value (-) |
| $f_{level}$ | Maximum fitness function value for a particular type of solution (-) |
| $FF$ | Fill Factor (-) |
| $FoS$ | Factor of Safety (-) |
| $g_0$ | Gravitational acceleration at sea level (m s$^{-2}$) |
| $G_{ox}$ | Mass flow rate over the port area (kg s$^{-1}$ m$^{-2}$) |
| $I_{sp}$ | Specific Impulse (s) |
| $L$ | Fuel grain length (m) |
| $L/D$ | Length over Diameter Ratio (-) |
| $L_{fuel\ tank}$ | Length of the Fuel Tank (m) |
| $m$ | Mass flow rate (kg s$^{-1}$) |
| $m_f$ | Fuel mass flow rate (kg s$^{-1}$) |

| | |
|---|---|
| $M_\tau$ | Total spacecraft mass at an instant $\tau$ (kg) |
| $M_f$ | Fuel Mass (kg) |
| $M_{ox}$ | Oxidizer Mass (kg) |
| $m_{ox}$ | Oxidizer mass flow rate (kg s$^{-1}$) |
| $M_{propellants}$ | Combined propellant mass - fuel & oxidizer - (kg) |
| $MEOP$ | Maximum Expected Operating Pressure (Pa) |
| $p_a$ | Atmospheric Pressure (Pa) |
| $p_c$ | Chamber Pressure (Pa) |
| $p_e$ | Exit Pressure (Pa) |
| $p_r$ | Port radius of the fuel grain (m) |
| $P_\delta$ | Turbo-pump Power Density (MW m$^{-3}$)[1] |
| $P_{axial}$ | Axial Pressure (Pa) |
| $P_{eq}$ | Equivalent Pressure (Pa) |
| $p_{ox}$ | Oxidizer Pressure (Pa) |
| $q$ | Fitness function score multiplier (-) |
| $R$ | Gas Constant (J K$^{-1}$ mol$^{-1}$) |
| $r$ | Regression rate (mm s$^{-1}$)[1] |
| $R_t$ | Cross-sectional radius of the throat (m) |
| $R_u$ | Longitudinal radius of the throat (m) |
| $S_b$ | Burning surface (m$^2$) |
| $t$ | Thickness (m) |
| $T_c$ | Chamber Temperature (K) |
| $T_{real}$ | Rocket thrust after accounting for the atmospheric pressure (N) |
| $V_{ox}$ | Required Oxidizer Volume (m$^3$) |
| $W_T$ | Web Thickness (m) |

**Greek Symbols**

| | |
|---|---|
| $\beta$ | Nozzle Half Angle (degree) |
| $\Delta p_{loss}$ | Fraction of pressure drop in oxidizer feed lines (-) |
| $\Delta V$ | Change in Velocity (m s$^{-1}$) |
| $\Gamma$ | Vandenkerckhove Parameter (-) |
| $\gamma$ | Ratio of Specific Heats (-) |
| $\rho_f$ | Fuel Density (kg m$^{-3}$) |
| $\rho_{ox}$ | Oxidizer Density (kg m$^{-3}$) |

[1] The unit is used instead of the S.I. base unit because it is used as the independent variable in an empirical law.

| | |
|---|---|
| $\sigma$ | Stress Case (Pa) |
| $\sigma_{allowable}$ | Allowable Stress (Pa) |
| $\theta$ | Nozzle Angle (degree) |
| $\xi$ | Combustion Quality (-) |

**Acronyms**

| | |
|---|---|
| ALOSS | Affordable Launch Opportunities for Small Satellites |
| CEA | Chemical Equilibrium with Applications |
| DARE | Delft Aerospace Rocket Engineering |
| ESA | European Space Agency |
| FIPSO | Fully Informed Particle Swarm Optimization |
| GTOW | Ground Take-Off Weight[2] |
| HDPE | High Density Polyethylene |
| HRM | Hybrid Rocket Motor |
| HTPB | Hydroxy-terminated Polybutadiene |
| INPE | Instituto Nacional de Pesquisas Espaciais - Brazilian National Institute for Space Research |
| LOX | Liquid Oxygen |
| MDO | Multidisciplinary Optimization |
| NIST | National Institute of Standards and Technology |
| NLR | Nationaal Lucht- en Ruimtevaartlaboratorium - Dutch Aerospace Laboratory |
| OAT | One At a Time approach |
| PaGMO | Parallel Global Multiobjective Optimizer |
| PMMA | Polymethyl Methacrylate |
| PSO | Particle Swarm Optimization |
| RPA | Rocket Propellant Analysis |
| SRB | Solid Rocket Booster |
| SRM | Solid Rocket Motor |
| Tudat | **TU D**elft **A**strodynamics **T**oolbox |
| V&V | Verification and Validation |
| VEB | Vehicle Equipment Bay |

[2] The acronym GTOW is used freely throughout this work as it is the standard in most literature, however, at times it indicates the Take-Off Weight of an air-launched vehicle in which case the adjective "Ground" is not appropriate. Even though this is acknowledged, GTOW is the one used in all instances in this report

# PHYSICAL CONSTANTS

| Name | Value | Name | Value |
|---|---|---|---|
| Earth Radius (Equator) | 6378137.0 (m) [1] | Ratio of Specific Heats Air | 1.401 (-) [2] |
| Sidereal Day | 86164.09 (s) [1] | Molar Weight Air | 28.9644 (kg kmol$^{-1}$) [2] |
| Molar Gas Constant | 8314.32 (J K$^{-1}$ mol$^{-1}$) [2] | Earth Gravitational Parameter | 3.9859383624e14 (m$^{3}$ s$^{-2}$) [1] |

# 1

# Introduction

This introductory chapter presents a short summary of the research (Section 1.1), the motivation behind this work (Section 1.2), clarifies the main research questions (Section 1.3) and provides a formulation for the main goals of the project in Section 1.4.

The following chapters of this report develop in detail the different topics: Chapter 2 discusses the previous work conducted at the TU Delft and and how this project collaborated with existing research. Chapter 3 presents the Air-Launch technology, Chapter 4 outlines the modelling of Hybrid Rockets as considered in this work. It discusses both the theoretical models and empirical relations used in the research. Chapter 5 focuses on Multidisciplinary Optimization (MDO) to explore the possible launch solutions, with a brief discussion on the available optimization methods and the optimization process itself. This implementation and the characteristics of the software are discussed in Chapter 6. Chapter 7 discusses the methodology followed for the modelling. As any engineering tool and model, the software and models used here underwent Verification and Validation (V&V), here described in Chapter 8. Chapters 9 shows the results, Chapter 10 conducts a sensitivity analysis on these and 11 shows the conclusions and future recommendations drawn from them respectively.

The research and its experiments are done using a software developed purposely for this research. The development of the software was the part that required the most effort and work hours and its implementation is worth discussing, especially if it comes to be used by future students.

## 1.1. The Research

The study presented here follows and expands from previous work done at the TU Delft (Department of Space Systems Engineering) initiated by faculty staff, on the topic of air-launch modelling. A research that fits into a more global effort on reducing the cost of access to space, particularly for small satellites. This work, initiated and supervised by faculty staff, started by Vandamme [3] and later continued by van Kesteren [4], investigated the cost and performance benefits of air-launching rocket propelled vehicles as opposed to traditional ground launches. The intention for this project is to expand the investigation by Vandamme and van Kesteren onto the field of Hybrid Rockets. The goal is to provide insight into whether there are benefits over traditional Solid and Liquid technologies and if these benefits are potentially enhanced in an air-launch scenario. The present research was continued through the literature study done in preparation for this thesis from which this report draws heavily. Whereas the Literature Study provided for a comprehensive review on the state of the art concerning the research goals, this report contains the work developed upon that research.

## 1.2. Motivation

In recent times there has been a growing demand for placing smaller satellites (nano and micro-satellites) in space. Between 2006 and 2009, more than half of all the satellites under 500 kg weighed less than 50 kg [5]. This means that satellites under 50 kg are now a dominant market sector. The decrease in size and mass of modern computers and instruments allows small vehicles to perform tasks that used to require much larger platforms before. This has been exacerbated with the standardization of the buses, especially the CubeSat phenomenon [5]. However, satellite launches have an inherent base cost that still makes it prohibitive for small-scale ventures to place their equipments in space. This is changing in recent years with a myriad of solutions being developed every day with the goal of reducing the cost per kilogram to orbit. The NLR developed a standard called ALOSS that aims to establish solutions for this need, this is an important part of the motivation for the work and is described in detail in Section 2.1. One of the proposed solutions is to use **air-launches**.

> An **Air-launch** is a deployment method in which one launches a rocket propelled vehicle from aboard a flying aircraft, as opposed to a traditional launch where the rocket would start its journey from the ground.

The reason why this solution is most suited for small payloads is that the size of the payload is directly limited by the carrier aircraft and thus only those that can be carried aboard would benefit from it. Previous work done at the Department of Space Systems Engineering at the TU Delft researched into the performance and advantages of this solution. This work ties in with general research interest of the faculty and more specific the space department to investigate advanced/novel techniques for access to space. To investigate the benefits of air-launch, Vandamme [3] and van Kesteren [4] developed a Multidisciplinary Optimization Tool that aims to yield the optimum launch vehicle configuration for a particular mission target. The optimum is defined as the cheapest launch solution one can build in order to attain a particular orbit for a particular payload mass. This tool allowed for a comparison between ground and air-launch solutions to be made in the sense that it showed for which cases each particular solution fits best. One of the main limitations, however, is that the tool only considers Solid Rocket Technology for its simulations. This Thesis aims to explore what possible advantages would there be in using Hybrid Rocket Technology instead.

> **Hybrid Rockets** are a rocket technology that produces thrust by starting the combustion process through the injection of Liquid Oxidizer into a Solid Fuel grain.

This differs from both Solid and Liquid Rockets. Solid burn through a solid propellant grain and Liquids burn the Fuel and Oxidizer both in the liquid state. This part of the study describes what advantages might exist in the use of both Hybrid Rockets and Air-launching and why it is worth considering as a solution. The continuation of the software started by Vandamme and van Kesteren is also part of the motivation for developing this work, creating the foundations of what may become a practical software tool to be used not only at the TU Delft but by companies and other users in academia that might be interested.

## 1.3. Research Question

This project has as a main objective to answer the following research question:

> How does the life-cycle cost of Hybrid Rockets perform when compared to competing technologies in both Air and Ground Launched scenarios?

This question has been addressed quite scarcely so far. Studies on Hybrid Rocket technology are mostly of an empirical nature and focus on attempting to build and develop a single rocket engine. The advantages of hybrid rocket technology itself have been explored in a theoretical way in other papers, see: 4.2. In terms of feasibility and vehicle design I found the paper in [6] to be the one closest to my work having some similar goals. My question furthers the research into the topic of overall vehicle design. By making part of this project the development of a software application with support for different rocket technologies, a more expansive investigation becomes possible making this research all the more unique. The software component is, in fact, central to conducting the research. Therefore, Section 1.4 is divided into two sections. The first focuses on defining the Research Question, the central topic that this research will aim to tackle. The second section defines the detailed goals, elaborating both on the path to addressing the question itself but also defining individual goals for the tools whose development will allow for the research question to be answered. The tool's development is also a primary goal of the thesis project.

## 1.4. Goals

Modelling a hybrid rocket is a task that can be taken to varying levels of complexity, and simulation is a "never-ending story", the precision level and the diversity of simulations was chosen to make the most out of the time available. The modelling is representative of the main challenges related to Hybrid Rocket technology. Naturally, the main goal is to answer the research question which is done, as explained, through the upgrading and expansion of the existing software tools. When the project is completed the goal is to have a software tool that can achieve three main goals. If the software is prepared to accomplish these goals, then the research question can be answered.

1. Simulation of full Hybrid Rocket Launches, air and ground-launched, to the same standards[1] that the current software does for Solid Rockets.
2. To produce the optimum[2] Hybrid Launch Vehicle for a particular target mission.
3. To produce not only the optimum Hybrid Launch but also allow for multistage vehicles to include stages of both types and get the optimum launch for mixed (hybrid and solid stages) configurations as well.

The comparisons Hybrid-Solid will be done in a Ground Take Off Weight (GTOW) perspective (since there is no cost data available for the hybrid case). In [4] shows that there is a relation of proportionality between GTOW and Cost. From this one can draw some comparisons but not 1:1 for every case. This is addressed in the Results chapter. However, since the tool is intrinsically coded to yield the *optimized launch* it has to be re-coded to disregard the cost when doing the comparisons. The Literature Study that preceded this thesis, [7], showed that the software was nowhere near the plug-and-play state that was assumed. In the initial assumption the work would focus on extensive testing of different possibilities and modelling refinement, this was shown not to be the case and thus the necessity for work on the software arose. To achieve the aforementioned goals, the following actions will be undertaken to improve the tool:

1. Extension of the Propulsion Module (addressed in Chapter 7 and 4):
   (a) To support at least 1 hybrid fuel-oxidizer combination
   (b) To support a properly motivated grain regression law that can be validated against existing experimental data .
   (c) To simulate the combustion process of a hybrid rocket and use the yielded thrust curve for the launch simulation.
2. Extension of the Geometry (Launch Vehicle Dimensioning) Module (addressed in Chapter 4):
   (a) To support the design and dimensioning of a Solid Fuel and a Liquid Oxidizer tanks.
   (b) To support both the choice of pumping feed systems and pressurized solutions (both self pressurized and pressurant-requiring).
   (c) To dimension the nozzle and fairing components in a similar approach to that of van Kesteren [4].
3. Extension of the Mass Module (addressed in Chapter 7 and 4):
   (a) To estimate the mass of individual components from simple models that are in conformity with their correspondent geometry models. Particularly:
      i. To support the estimation of the masses for Solid Fuel and a Liquid Oxidizer tanks
      ii. To support the estimation of the masses for pump-fed (through historical data, see [8]) or pressurization systems.
4. Improved software architecture (addressed in Chapter 6):

[1] The same standards here mean the same precision and modelling accuracy of the software developed by van Kesteren and Vandamme. It is about taking the same approach but expanding its scope greatly

[2] Preivously, in [4], for Solid Rockets in [4] the optimum was defined as the launch with the lowest cost, here, due to the lack of historical cost-estimation data, optimum is defined as the launch vehicle configuration with the lowest GTOW

(a) Improved User Experience and Usability of the tool, in particular removing the need to re-compile the program every time a new case is to be tested

(b) Upgrade the current tool's back-end and the libraries it depends on.

(c) Facilitate further upgrading of the tool so that the work can be continued by other students in the coming years

The development of this product is essential for answering the research questions. It will enable a comprehensive study on the advantages of air and ground launched hybrid rockets which is the main goal of the thesis project. However this is not considered to be a fully working Hybrid Rocket launch optimizer. My recommendation is that others take advantage of this work in the same way I did with the work of the people who preceded me. One possible improvement on the tool is the implementation of Liquid Rocket Models for instance.

**Publishing goals:** Finally I set as personal goal to create work that is worth of publishing either in an journal or at a conference which is one of the best ways to confirm the value of this project.

# 2 Collaborations and Previous Work

During the literature study that preceded this thesis, [7], papers relevant for the topic were surveyed. There were two main sides to this task. The first was understanding hybrid rocket modelling and the practical implications of the technology. The second was to acquire some experimental data for verification and validation but also for understanding the assumptions that were necessary to make in order to get a running model. This project counted on experimental data from a few sources and a brief outline of those is given in this chapter. DARE (Delft Aerospace Rocket Engineering) and Nammo (A Space and Defence company) were the two primary sources. However, the collaborations the project was involved with extend further than just data. The NLR (Dutch Aerospace Laboratory) has significant interest in the topic due to a framework of theirs: ALOSS. Contacts with the Brazil's INPE were also established to a similar purpose. Finally the work that had already been done at the TU Delft (Technische Universiteit Delft) was also an important part of the project and its scope and influence is also described in this chapter.

## 2.1. NLR and ALOSS

ALOSS - Affordable Launch Opportunities for Small Satellites is a program by the NLR, the Dutch Aerospace research institute, that aims to tackle the problem of the high cost associated with launching even the smallest satellites. The goal is to develop an independent launch solution that enables the smaller satellites to be delivered independently without having to piggy-back on the launch of larger satellite missions (which are usually the ones that can afford large launchers). The proposed solution is described in a technical feasibility study found in [9]. It considers air-launch platforms as a way to get 10-30 kg payloads to orbit. Specifically the Lynx Spaceplane and the F-16 Fighter with a payload of 10 kg. The launcher for the F-16 concept is illustrated in Figure 2.1 and its characteristics can be found in the **Appendix section of this study (see: Tables B.1 and B.2)**. It is an important point of reference for this study since it is exactly the type of concepts that the project's research is on. Detailed dimensional characteristics of the rocket are confidential. Van Kesteren [4] estimated a length of

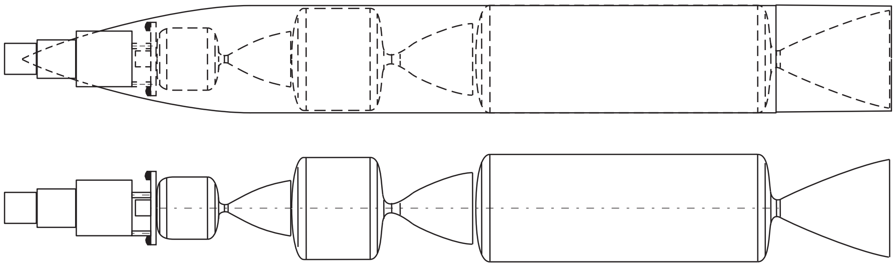


Figure 2.1: Top and side views of the 3-stage launcher for the ALOSS F-16 Air-launch concept - from [9]

| Parameter | Value |
|---|---|
| GTOW | 1855.3 (kg) |
| Payload | 10 (kg) |
| Length | 5.5 (m) |
| Diameter | 0.66 (m) |

Table 2.1: ALOSS Vehicle characteristics.

5.5 m and a diameter of 0.66 m for ALOSS. The estimation is necessary due to the full extent of the data not being publicly available. According to [9] the lift-off mass is 1855.3 kg and the payload ratio approximately 0.0054. The main characteristics for the ALOSS Rocket are summarized in Table 2.1.

## 2.2. Previous work at the Department

The people that preceded me made this research possible. The contributions from the previous students consist not only of research but also some of the software tools that I expanded upon. The work of Vandamme and van Kesteren is of particular importance and is discussed here. The literature study and Thesis by van Kesteren (see [10] and [4]) constitute an important source of information for this research. Vandamme started the work in 2012 with his thesis on "Assisted-Launch Performance Analysis Using Trajectory and Vehicle Optimization" where he developed software that allowed for performance analysis of assited-launch scenarios. This was all, in part, made possible due to the contributions of Frank Engelen that developed the original ascent simulator [11]. With his thesis, van Kesteren, continued and elaborated on this work, greatly expanding the tool in order to get a software that can compare ground launches to air-launches for Solid Rocket case. The comparisons now work on a variety of dimensions which even include the cost. The final product of his work is actually an MDO (Multidisciplinary Optimization) tool that can calculate the optimum launch solution for a particular target mission.

**Assisted-Launch Performance Analysis Using Trajectory and Vehicle Optimization** The thesis by Vandamme focused on Assisted-Launch performance analysis. Here the main focus was to answer the question: "How does a launch-assist affect the performance and the design of a launch vehicle?" [3]. He investigates the fact that most launch-assist platforms in use or under experimentation nowadays are re-purposed vehicles that are in no way optimized for the task at hand. A software was developed that can simulate various launches, including launch vehicle characteristics, different trajectories and different initial conditions [3]. The study considers only single stage launches in order to give insight on the assist required to make a particular launch competitive. The tool yields an optimized set of launch vehicle parameters along with the characteristics of the assist and the optimal trajectory [3].

**Air Launch versus Ground Launch: a Multidisciplinary Design Optimization Study of Expendable Launch Vehicles on Cost and Performance** This thesis uses part of the C$^{++}$ code from Vandamme and completes it by building a full MDO tool upon it. For the last version of this software, van Kesteren used a highly modular architecture wherein the different factors that characterize the mission are defined and then integrated into the optimizer. The tool is designed to answer the following research question:

> What is the performance gain in terms of cost, gross take-off weight and the amount of $\Delta V$ required to orbit for an optimized expendable air launched vehicle in comparison to an optimized expendable ground launched vehicle for different launch parameters (release altitude, velocity and flight path angle), payload classes and the presence of a wing? [4]

The physical and operational modules considered for this optimization are the following:

1. **Propulsion** - Estimates the amount of required fuel, the length of the stage, estimates mass flow and thrust.

2. **Launch Vehicle Properties** - Estimates overall mass and geometry based on the requirements given by the thrust module (e.g. dimensions the case for the required grain length, estimates nozzle mass from thrust levels, etc.)

3. **Aerodynamics** - Estimates aerodynamic parameters such as drag, heat flux and dynamic pressure using an atmospheric model and the rocket's current position.

4. **Trajectory** - Returns the required Pitch Angle value for every instant based on a list of 11 points provided by the user or the optimizer.

5. **Cost** - Estimates Cost based on the estimates for the Propulsion and Vehicle Properties modules.

6. **Environment** - Establishes the Gravity model for the earth, connects the launch vehicle to its surroundings (atmosphere and gravity) and propagates the trajectory.

The main focus was the development of an optimization architecture that researched the performance gains when comparing ground and air-launches for the same objectives. The remaining discussion of the work mainly concerns the viability of particular missions under size or budget constraints like the ALOSS (Affordable Launch Opportunities for Small Satellites, see: Subsection 2.1) framework by NLR. This tool only supports Solid Rocket motors.

## 2.3. Nammo

In [12] the company Nammo AS (An aerospace and defence company based in Norway) describes its involvement in hybrid propulsion while including some valuable test data that are very relevant to use for this project. The end-goal is to reproduce some of the tests in a Simulated Flight scenario. This scenario is a test case where the properties and characteristics of the Hybrid rockets developed by NAMMO are to be used in the tool. The test serves to observe and compare the outputs of the tool with the results that are available, both **flight-wise** and **model-wise**. Hybrid rocket data is scarce and this cooperation is fundamental for the success of this project. The tests described in [12] used Hydrogen Peroxide at 87.5% concentration as the oxidizer and HTPB/C as the Solid Fuel. HTPB/C is a proprietary propellant mixture by Nammo AS, described as HTPB "with strengthening additives added for improving the polymers mechanical properties [12]". The test data available includes the following:

1. Oxidizer to Fuel ratio analysis

2. Oxidizer Mass Flow

3. Regression Rate

4. Chamber Pressure vs. Time

5. Chamber Pressure vs. Time (Thrust modulation with throttling)

The data was obtained with a static test set-up, and the paper does not include Launch Vehicle data which is crucial to the simulations that are to be run. The available data is instead used to establish a tentative launch vehicle based on the engine specifications provided. This is in fact the role of the tool, the estimation of the launch vehicle parameters. Even though there is no data for validation *for this particular case* it remains an important baseline, and one on which the information available is enough.

**Nammo AS Meeting** TU Delft established a contact with Nammo in which the company provided data on their engine and showed their interest in the future results of this thesis. The new information provided included a wide range of values including the inert mass fraction of the motor, reportedly near 0.25. Values like thrust, $I_{sp}$, mass and dimensions of the individual components and other relevant engine data were made available to us. Another information that is highly relevant was their motivation behind their choice of Fuel and Oxidizer. Nammo stated that the Hydrogen Peroxide-HTPB/C choice comes from both the combustion stability it provides, the energy density and long years of experience behind it.

## 2.4. DARE and the Stratos II Rocket

DARE - Delft Aerospace Rocket Engineering, is a student rocket society at the TU Delft. With the Stratos project and the Stratos II Rocket, the team has conducted research and developed a hybrid rocket. The team was kind enough to provide me with a great deal of information and data. This collaboration was fruitful for both sides as I shared my findings and conclusions that came from the use of their data. The engine is so far the most developed component. It is called the "Aurora" engine. Test data is extensive, and the full specifications (geometry and mass wise) have been provided to me by the team.

**"Development of a Hybrid Rocket Motor for the Stratos II Rocket" - [13]** The first of the three papers, describes the development of a small hybrid engine designed to reach 1 km altitude. An extensive systems engineering approach is covered along with the descriptions of the subsystems that compose the rocket. Information includes dimensions, mass and rocket performance parameters. This information can be found in the Appendix to this report. Test results including experimental curves are also included.

**"Sorbitol-Based Hybrid Fuel Studies with Nitrous Oxide for the Stratos II Sounding Rocket" [14]** This particular paper discusses the motivations for their choice of Oxidizer and Fuel, Nitrous Oxide - Sorbitol/Paraffin/Aluminium, and includes thrust and specific impulse data, as well as theoretical curves. A discussion on why the increase in the regression rates achieved by the addition of paraffin is very important is also present and is a part of the motivations for the Propellants section of this work.

**"Test Campaign on a 10 kN class sorbitol-based hybrid rocket motor for the Stratos II Sounding Rocket" - [15]** Describes the test campaign for the more powerful DHX-200 Aurora Motor. Designed for 10 kN Thrust and 200 kN s total impulse. It verifies the performance increase given by the additions of paraffin and aluminium. As per their previous study, it concludes that the reduction in combustion instability and improvements in regression rate were determinant in getting the required performance. The test consists of static firings of the motor. The motor will power the Stratos II rocket that is supposed to deliver 15 kg of scientific payload to 50 km. Data includes test Thrust profiles and several performance parameters.

**Other Data** The team has also provided me with detailed CAD Drawings, the raw test data and some related information that proved useful when simulating their rocket.

**Collaboration** With all the data provided on the Aurora Engine, one can simulate the optimized launch vehicle that uses this engine. The results allowed me to validate my models to great extent and to start a discussion on the current design's abilities.

## 2.5. Other Collaborations and Relevant Papers

### 2.5.1. Konkuk University - Korea

The paper "Preliminary Design of the Hybrid Air-launching Rocket for Nanosat" [16] is of particular interest because it tackles the concept design of an air-launched hybrid rocket. The conceptual approach is described in detail. The characteristics of the proposed rocket being 5.89 m, and the payload mass 3.56 kg. A minimization of the total take-off weight is also the main goal as in [4]. Their final design data is given including structural and propellant masses for each of the stages. The geometry considered for the propulsion system is presented as two connected tanks (separate from the outer shell) and the chosen launch vehicle is a winged one, the wing profile also described in the paper. This case is particularly important because it allows for a comparison of the results of my tool with theirs for the air-launch, winged vehicle case. Which so far remains my only and most important validation option for this configuration.

### 2.5.2. INPE - Brazil

The Instituto Nacional de Pesquisas Espaciais of Brazil (National Institute for Space Research) has also conducted some development analysis on the matter. Their papers "Study of Paraffin/H2O2 Hybrid Rockets for Launching Nanosats" [17] and "Preliminary Analysis of Hybrid Rockets for Launching Nanosats into LEO" [18] are quite relevant for this research. Both papers describe their approach to the modelling of the launch vehicle keeping in mind the performance of the propellants that they studied. The whole system is described for all three stages of the rocket. Geometrical and Mass models for the Fuel Chamber, Nozzle, Pressurizing System, Oxidizer Tank and Stage Case are given, based on simple stress equations. Their system uses a simple Helium tank for pressurizing the oxidizer (Hydrogen Peroxide) and use catalytic ignition to start the combustion with the Paraffin grain. For this fuel-oxidizer pair, they provide as well Specific Impulse curves as a function of the O/F ratio. For each of the stages propellant mass fraction, as well as overall dimensions and mass for the individual components are provided. In their paper "Development of a test bench for hybrid thrusters" - [19] they described their test set-up for the hybrid rocket and include experimental data such as thrust and specific impulse curves.

**Contact with INPE** I have successfully established e-mail contact with INPE. They showed interest in my research, and in having me there if this becomes possible. They informed me that their research has progressed in fuel grain regression analysis and that therefore they would be interested in the Launch Vehicle Optimization part of my work. As of the writing of this thesis no further contact was initiated on either part. This is left for future work due to time constraints.

# 3 Air-launch

Whereas the first two Chapters introduced the work and placed it in today's research context, this chapter concerns the introduction of the first core component of this thesis: air-launching rocket vehicles. This chapter is a brief description of the Air-launch technology and its theoretical model that is reproduced by the software developed for this research. The theoretical modelling for the Air-launch part of this project is highly based on the works of van Kesteren [4], and Vandamme [3]. The expansion on this part was limited and a lot more effort was put into researching and modelling hybrid rockets. Air-launch is, nonetheless, a critical part of this project and is therefore presented here. Following a brief historical overview of Air-launching, the principles and advantages are then described. The final section describes the model used here.

## 3.1. Why Air-launching?

Air-launch brings several advantages over traditional ground launches such as: the ability to fly over or around weather that wouldn't usually allow for a launch to take place. Even though flight is also sensitive to weather, it is much more flexible than a ground launched rocket. The ability to achieve **any launch azimuth, operate free of scheduling constraints and having reduced safety requirements** are also advantageous aspects of the technology. These aspects derive from the intrinsic nature of flight, which is far more mastered and optimized for cost and operations than space-flight. Moreover, Air-launches can take place without a dedicated launch facility, being able to be **launched from almost anywhere** - provided that the carrier aircraft can take-off and land. However, the most significant advantage is that it allows for significant propellant savings by flying over the denser part of the atmosphere (thereby reducing drag losses) and having the velocity of the carrier aircraft add to the $\Delta V$ of the rocket launch vehicle [4, 20]. This reduces the $\Delta V$ (change in velocity [m/s]) that Air-launched rockets require since the carrier aircraft already provides a considerable fraction of it [4]. An Air-launch takes place as high as the carrier aircraft permits it, usually at over 10 - 15 km. This is in stark contrast with traditional ground-launches. When a ground-launched rocket reaches the equivalent of the starting altitude for an Air-launch it has already spent most of its initial fuel [4]. Even if only subsonic aircraft are considered an Air-launch can add between 350-900 $\mathrm{m\,s^{-1}}$ to the $\Delta V$ of the Rocket Launch Vehicle, the addition of a wing to the launcher can save an additional 100-200 $\mathrm{m\,s^{-1}}$ [21]. Because the drag losses are more significant for smaller vehicles, this is the class that has the most to gain from an Air-launch [22]. Another performance gain to be considered is that rocket nozzles are designed for a particular pressure-altitude in order to ensure perfect expansion. Since the rocket will have already flown past by the most significant part of the atmosphere, the nozzle optimization is done for a smaller range of altitudes and is thus more efficient than a design made to sustain a full atmospheric crossing [23]. The main cost advantage comes from the fact that, in an Air-launch system, the first half of the task, the airlift of the rocket vehicle, is done by a completely reusable vehicle. This is in contrast to ground launches where the first stages, and all the remaining ones for that matter, are completely disposable. In essence, Air-launch brings ease of use combined with a reduction in launcher size and mass. Air-launching a rocket is demonstrated by van Kesteren to reduce the GTOW of the launcher of a 10kg payload from approximately 3,000 kg to less than 1,000 kg [4]. For this reason, investigating the

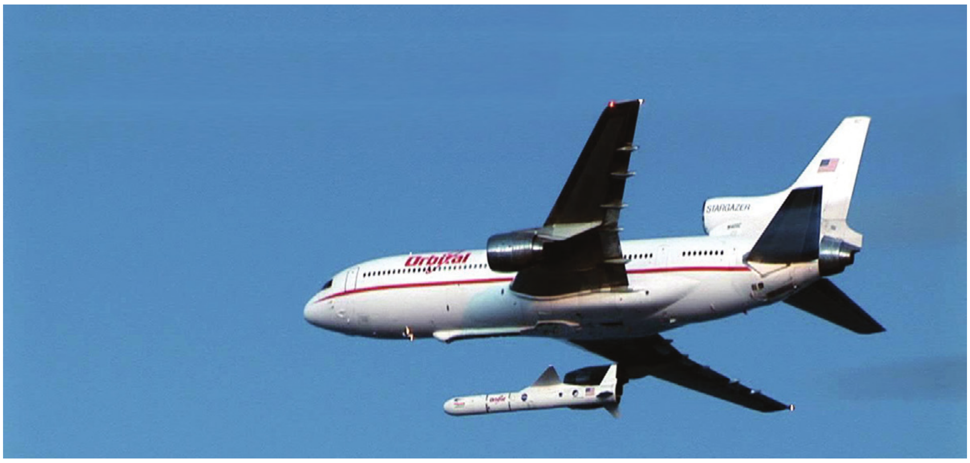


Figure 3.1: A Pegasus XL Launcher being Air-launched by the Stargazer Aircraft at the moment of release [30].

effect of Air-launch for Hybrid Rockets is paramount.

## 3.2. Historical Perspective

One of the earliest examples of Air-launched rockets is the American "NOTSNIK", a Launch Vehicle carried aboard a Douglas F4D-1 Skyray with a 1 kg payload that was launched from 12500 m at 725 km/h and at a 50° initial flight path angle. This effort was part of the cold war space race and was cancelled in the late 1960's [24]. Other attempts were made along the years with the British HOTOL [20], the USSR's "MAKS" [25], the United States' NASP [26] and the European Sanger II [27] being the most noteworthy. In recent times there has been a resurge of interest in Air-launches. Commercial ventures like Virgin Galactic's SpaceShipTwo - hybrid motor-powered spaceplane launched from a "mothership" known as WhiteKnightTwo which has already undergone successful powered flights, albeit suborbital ones [28] and Stratolaunch's Stratolaunch System consisting of a carrier aircraft developed by Scaled Composites that will carry a multi-stage booster manufactured by SpaceX [28]. The most well known Air-launch system is Orbital Sciences' Pegasus XL rocket - a 3 stage solid booster with wings attached to the first stage. A Pegasus XL launch costs around 12$-15$ USD million with the carrier aircraft being the only reusable component in the launch [20]. At the time of this writing the Pegasus XL has completed 42 flights with 3 Failures and 2 Partial Failures in its record; it is still an operational launch system [29]. Figure 3.1 shows the Pegasus launcher during the moment of release. The Pegasus case is the one referenced here because it is the one that is the most developed and with the best established record. Comparisons for cost and performance of solid-based rocket launch solutions, refer to [10] and [4] where an in-depth investigation into the concepts was conducted.

## 3.3. Modelling Air-Launch

The program's Launch Simulation code is a combination of three separate models: Aerodynamic, Gravitational and the Vehicle itself (these are addressed in further detail in Chapter 6). Modelling Air-launch is not unlike modelling any launch, the main difference is in the starting conditions. Whereas ground-launches start with standing still (in the local reference frame), Air-launches have an initial velocity and a position that is higher than a conventional launch. However, for the modelling in the ascent simulator, all that is relevant is that a correct State Vector (vector representation of the rocket's position and velocity) is given. This means that all the information that one must supply to simulate an Air-launch is the following:

1. Latitude and Longitude
2. Starting Altitude
3. Aircraft's velocity
4. Initial (the aircraft's) Flight Path Angle

The simulator is, in essence, a position *propagator* whose workhorse is an *integrator*. It integrates the state vector, calculating the position in successive instants, starting in the initial point in the initial instant (epoch). For a ground launch the state-vector describes a point on the Earth's surface, with no local velocity. The only difference in the Air-launch case is that **the starting condition is at a prescribed altitude with a prescribed starting local velocity**. After this indication, the gravity and atmospheric models kick in automatically influencing the motion of the rocket. Upon ignition, the thrust force then guides the rocket into orbit. Separation concerns, such as collision with the carrier aircraft, are not taken into account here. The case that the simulator considers the Thrust to begin immediately at the start of the simulation. In a more detailed model, some time for the separation to occur safely should be accounted for. This is left as a future recommendation, as the limited time for this thesis was focused on the Hybrid Rocket Modelling and on upgrading the Software tool used. A study on the impact of the separation process on performance of air-launches is an interesting topic that, for the same reasons, is left for future study.

# 4 Hybrid Rocket Modelling

Hybrid rocket modelling is one of the pillars of this research. Having a proper working model is key to make an in-depth comparison between the performances of Hybrid and Solid rocket technology. Gathering from the insight given by the Literature Study that preceded this Thesis, I state and explain my choices for the models that are to be implemented in my tool in this Chapter of the report. There are other models of hybrid rockets with particular emphasis for the one in [6] that takes a similar approach to this research, and the experimental work on real models in [12, 14, 18]. This work stands out for the multidisciplinary approach applied to to the overall hybrid rocket vehicle design. All the models pertaining to propulsion come from [23], the reader for the course of Thermal Rocket Propulsion at the TU Delft. The mass estimating relationships for all that is common with Solid Rocket Engines is taken from [4]. Some dimensioning relations come from [31], the reader for the course Aerospace Design & Systems Engineering Elements also at the TU Delft.

## 4.1. Why Hybrid Rockets?

Hybrid rockets have many advantages over Solid and Liquid rockets. Hybrid rockets are thought to be inherently safer, since the fuel is inert meaning it can be transported and handled safely. Moreover, the system as a whole is non-explosive since an intimate mixture of fuel and oxidizer is not possible [32]. The added safety should be the focus of future study, as it is increasingly an important factor especially for human spaceflight. Unlike SRM (Solid Rocket Motors), HRMs (Hybrid Rocket Motors) can be throttled for thrust by changing the mass flow of the oxidizer that is injected into the fuel chamber, this feature also enables complete thrust termination - something highly desirable in the event of an abort procedure, which SRMs do not permit[32]. The combination that makes the technology possible, i.e. solid fuel and liquid oxidizer, also causes the HRM to have a high propellant versatility as many different combinations of fuels and oxidizers with different characteristics are possible [32]. Other advantages such as a lower cost per launch and the fact that the propellants can be made environmentally friendly (and non-toxic) [33] are also highly appealing. Hybrid rockets also have a theoretically higher $I_{sp}$ (specific impulse [s]) than Solid Rockets [34] and a higher Fuel Density ($\rho_f$ [kg m$^{-3}$]) than Liquid Rockets [35]. This study focuses on the feasibility of Hybrid Rockets as small-satellite launching technology. The advantages listed here are certainly part of the appeal but are not the focus of this study. Therefore, it is my **recommendation to study them in the future**, especially the implications of throttling, and a review on the $I_{sp}$ values of a wide range of fuels. However, hybrids also have some known disadvantages. These include: A low fuel grain regression rate, a lower combustion efficiency has been reported in some studies [33] and a higher inert mass when compared to solid rockets is expected due to of feed system components. The first disadvantage has been overcome using multiple ports in the grain - increases the burning surface - and by implementing paraffin based fuels, which affects the way the combustion occurs increasing the regression rate directly. Due to the lack of extensive data on these topics, there is no universal correlation for quantifying these characteristics. An extensive survey of all developed Hybrid Rockets would be invaluable as future work to improve the usage and development of the technology. Hybrid Rockets will probably not eliminate the competing technologies but there is a place for it: if the same or comparable performance to current technology can be achieved then Hybrid Rockets have the edge

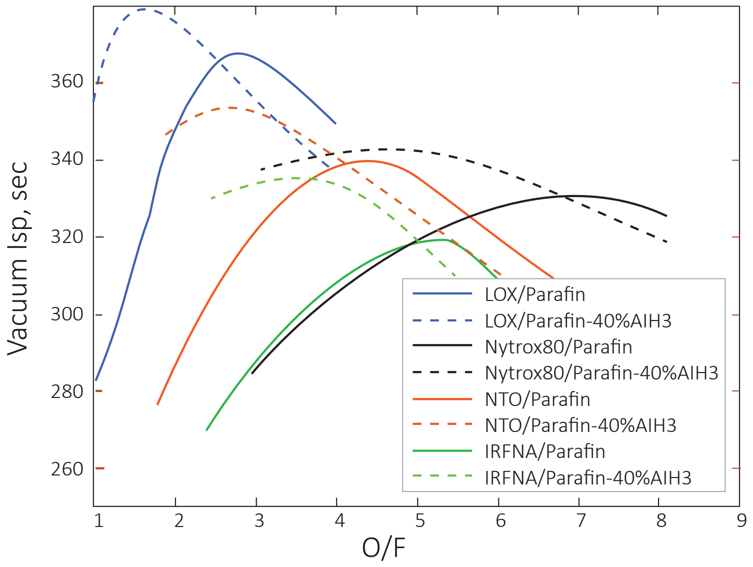


Figure 4.1: Theoretical $I_{sp}$ for various combinations of Hybrid Rocket Propellants (considered expansion ratio for this plot is 70) - from [34].

on safety, making them particularly suited for human spaceflight applications. Another benefit, for instance, is the fact that the propellants are mostly non toxic and have low explosion risk. The inherent safe characteristics make it possible for small organizations with less specialized facilities to be able to fabricate them as well and thus allowing for a faster adoption and maturing of the technology in the future.

> The question then remains: are Hybrid Rockets close to achieving the same performance (measured in payload-to-orbit ratio) than the more mature solid rocket technology?

To ensure that it is so and to give extra motivation for the work, a simple Excel model was developed using simple Tsiolkovsky Rocket Equation. I used a simple case study from [18] from which I used the propellant mass ratio values for the Solid Rockets listed (around 0.9) and the one for the Hybrid Rocket being developed in the paper (closer to 0.8). At the listed $I_{sp}$ : $279$s (specific impulse, a common measure of performance for rocket engines, see 4.3) the payload ratio is *9 times* lower than for a Solid Rocket with an $I_{sp}$ : $290$s. The $I_{sp}$ has to reach a level of approximately $368$s for the ratios to be matched. Since higher $I_{sp}$ values, at least theoretical ones, have already been reported (around 380s, see Figure 4.1 and [34]) and since the inert mass ratios can also be improved, through the inevitable maturation of the technology, something that for Solids is not likely to happen due to their already high degree of maturation, it is likely that Hybrid Rockets are a valuable and interesting opportunity for spaceflight. This is expected since the values for the mass ratios and performance are already in the neighbourhood of being beneficial over the solid technology. Natural technological maturing brings, in the world of rocket building, more efficient structures with lower inert mass derived from the improvement of the manufacturing procedures and elimination of performance losses. There is potential for the performances of the competing technologies to be surpassed in the future. The results and steps for the simple model's calculations can be found in the Appendix in Figures A.1 and A.2.

## 4.2. History of Hybrid Rockets

Hybrid Rocket Motors (HRM) have been tested since the 1920s. Hybrid Rocket motors are rocket motors that burn Solid Fuel with Liquid Oxidizer. Despite its many advantages over the competing technologies (Liquid and Solid rocket motors) all commercial ventures based on it have failed so far. However a revival of interest in hybrid rockets has been happening since the mid-90s, this is attributed to the improvements in the regression rates of the fuel grain that have since been achieved [33]. The reason why regression rates are paramount is that they are a direct measure of the total mass flow and thus thrust. A low

Figure 4.2: Virgin Galactic SpaceShipTwo during a powered flight [28]

regression rate implies that a much larger burn surface would be required to attain the same mass flow, thus reducing the potential benefits of using hybrid rocket technology. One of the earliest successful attempts with a HRM powered flight was conducted by the Pacific Rocket Society during the 1940's with a successful flight in 1951 [32]. The U.S. Air Force used the technology in its drone program for 15 years for the Firebolt target drone, HRMs were chosen for its wide throttling ratio [32]. In 1981-1985 Starstruck developed the Dolphin sounding rocket, which was then continued in the following decade with the H-1800 a $LO_2$/HTPB motor [35]. In 1998, the use of paraffin was reported at Stanford University as a source of performance improvement [36]. The use of paraffin was the main breakthrough that allowed for a substantial increase in regression rates. Three times as much according to [33] and 7 compared to classic HTPB Hybrid according to [37]. As a consequence, this has been credited with the revival of interest in HRM's [33]. Other developments were made by Lockheed in 2000s and finally Scaled Composites and SpaceDev developed the $N_2O$/HTPB hybrid rocket for the sub-orbital vehicle SpaceShipOne whose successor is today operating for Virgin Galactic as part of its suborbital flight program [28, 35]. Table 4.1 contains a summary of some existing Hybrid Rockets with their respective properties. In the table one can find general reference data for some real-life rocket engines. This helps us get a better grasp on the orders of magnitude involved for each of the values. Particularly the values for the inert mass fraction of the engines. The inert mass fraction, as discussed multiple times along this work, is one of the main concerns when looking at the disadvantages of hybrid rocket solutions. It is accepted that in general there will be a higher inert mass fraction, when compared to solid rockets, due to the presence of feed system components and all the accompanying parts. Whereas solid rocket engines have inert mass fractions on the order of less than 0.1 (see the database in [10]), meaning a propellant mass fraction higher than *0.9*. In Table 4.1, this is represented in the row "Mass Fraction" corresponding to the propellant mass fraction. There we see only one value greater than 0.9 which is the ESA 3kN engine. This optimistic value is because this case is purely a theoretical design. The other cases for which data is available have, on average, 0.82 as their propellant mass fractions.

## 4.3. Modelling a Hybrid Rocket

To carry out the analysis that is intended with the global tool, it is necessary to first develop the theoretical basis for the particulars of hybrid rocket modelling. Since the goal is to transpose these models to software, each part of the modelling is approached as its own module, analogous to the software's structure. The modules in the work of van Kesteren [4] serve as the basis for this organization. Each module tackles its own physical aspect of the simulation. The primary goal is to model accurately the propulsion for the Hybrid case, addressed in 4.3.1, and the geometry of hybrid rocket systems, addressed in 4.3.2. In order for propulsion simulation to be possible, new thermodynamic and chemical data on the propellants was also included in the program, theory on propellants is addressed in Propellants. The trajectory module required only small improvements, since these were only on a technical level, they are addressed in Chapter 6, which covers the software development. The same applies to changes done to the environment models.

| **Motor designation** | **H-30** | **H-500** | **H-1500** | **H-1800** | **3kN** | **HY-157** | **0.1 kN** | **Chiron** | **0.5 kN** |
|---|---|---|---|---|---|---|---|---|---|
| Company | AMROC | AMROC | AMROC | AMROC | ESA | DLR | TU-Delft/TNO | USAFA | UoS |
| Average vacuum thrust [kN] | 14.1 | 400.3 / 315 (sl) | 758.4 | 1143.1 | 3 | 5224 (sl) | 0.088 (sl) | 4.5 | 0.5 |
| Total vacuum impulse [kNs] | | | | 82292.1 | | | | | |
| Effective vacuum specific impulse [s] | 287.3 | | 295.2 | 278 | 295 | 298.2 (sl) ?? | 184.6 (sl) | | 310 |
| Total mass [kg] | 641 | | 31200 | 36970 | 6994 | 188000 | | >15 kg | |
| Propellant mass [kg] | 497 | | 24890 | 31980 | 6710 | 157200 | 0.794 | | |
| Mass fraction [-] | >0.78 | | 0,8 | 0.87 | 0.96 | 0.84 | | | |
| Burn time [s] | 100 | 70 | 95 | 72 | 7000 | | 16.4 | 15 | 18.5 |
| Diameter [m] | 0.61 | 1.29 | 1.83 | 1.85 | 1 | 5.2 | | 0.21 | 0.1 |
| Stage length | | 15.55 | | 19.5 | | | | 3.62 | |
| Combustor length [m] | 0.71 | | 8.79 | | >2.6 | 23.5 | | | 0.36 |
| Length of casing (w/o nozzle) [m] | | | | 9.80 | | | | | |
| Stage length to diameter ratio [-] | | 12.05 | | 10.52 | | | | 17.24 | |
| Propellant | HTPB-N2O | HTPB-LOX | HTPB-LOX | HTPB-LOX | PE-NTO | PE-LOX | PE-GOX | HTPB-LOX | PE-HTP |
| Mass mixture ratio [-] | | | | 1.7 | 3.6 | | 4.83 | | |
| Grain configuration | | 11-spoke | 15-spoke (triangular port shape) | 15-port + cylindrical port | Single port (cylindrical) | 12 port (cylindrical) | Single port (cylindrical) | 3 port (pie-shaped) | Single port (cylindrical) |
| Liquid feeding | Vapor pressure | He pressurant | Pump | He Pressurant | Pressurant | Pump | Pressurant | He pressurant | N2 pressurant |
| Tank material | | Steel / carbon fibre | Aluminum | | Titanium | Aluminum | Steel | 0.003 mm Aluminium | |
| Casing material | Graphite-epoxy | Graphite-epoxy | Graphite-epoxy | Graphite-epoxy | Kevlar | Steel | | 302 steel | 316 steel |
| Thermal insulation | | | Kevlar | | EPDM | | | | |
| Nozzle material | Carbon-phenolic | Silicon-phenolic / glass-phenolic | Silicon-phenolic / glass-phenolic | Silica-phenolic | Carbon-carbon | | | Graphite / silicon-carbide | |
| MEOP [bar] | | | | 31.5 | | | | | |
| Average pressure [bar] | 31 | | 34.5 | 24.9 | 10 | 48 | 9.1 | 18 | 18.1 |
| Expansion ratio | 75.1 | 40 | 17.7 | 9 | 215.8 | | | 3.34 | 3.34 |

Table 4.1: Examples of developed Hybrid Rockets with their respective properties. Adapted from [38], data from [1, 39, 40]. The value accompanied by "??" indicates that the source data might be unreliable and that the value may correspond to the vacuum specific impulse and not the sea-level one.

### 4.3.1. Propulsion

In this subsection the matter of propulsion simulation is addressed. It describes the models chosen for this project and motivates the choices as well. The main topics covered are Ideal and Hybrid Rocket Theory, the choice and influence of the propellants and their properties as well as combustion effects particular to Hybrid Rockets (a variable O/F ratio). In the end of this subsection all the necessary elements to design a propulsion module will have been covered.

#### Ideal Rocket Theory

In order to model propulsion the principles of ideal rocket theory are the starting point. Equations 4.1 to 4.5 are all from [23] unless stated otherwise.

$$T = \dot{m} \cdot I_{sp} \cdot g_0 \tag{4.1}$$

Equation 4.1 defines Thrust and applies to any type of rocket technology be it Solid, Liquid or Hybrid. $T$ designates thrust, $m$ the mass flow rate, $I_{sp}$ the specific impulse and $g_0$ the gravitational acceleration at sea-level for Earth. The main difference will focus on $\dot{m}$ which in the Hybrid case is a combination of the contribution of the Oxidizer flow from its tank with the vaporized fuel and the combustion products generated in the chamber. Thrust, in rocket engines, is about enhancing what would otherwise just be a simple momentum change. While Equation 4.1 describes thrust in general, Equation 4.2 shows how the rocket propulsion affects this. In Equation 4.2, the mass flow rate is multiplied by $C_F$, the thrust coefficient, and $c^*$, the characteristic velocity. The Thrust Coefficient (Equation 4.5) accounts for the contribution of gas expansion in the nozzle, and therefore, the pressure effects. The characteristic velocity reflects the energy level of the propellants being used, it is only a property of the propellants, [23], and independent of the pressure ratio.

$$T = \dot{m} \cdot C_F \cdot c^* \tag{4.2}$$

In order to calculate the terms in Equation 4.2 we analyse each of them. The mass flow, $m$, is a design variable that is dependent on some of the characteristics chosen for the feed system, this is addressed in Hybrid Rocket Propulsion. For now it is left as an unknown. The terms $A_e$, $p_e$ and $p_a$ are not dependent variables. The first two are design choices related to the nozzle and the last one is simply an environmental propertie. We then calculate the remaining unknowns in the equation - $C_F$ and $c^*$.

$$c^* = \frac{\sqrt{R \cdot T_c}}{\Gamma} \tag{4.3}$$

Here $\Gamma$, the Vandenkerckhove constant, is defined in Equation 4.4, where $\gamma$ is the ratio of specific heats, a thermodynamic property of the propellants. The chamber temperature is represented by $T_c$.

$$\Gamma = \sqrt{\gamma \cdot \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}} \tag{4.4}$$

Due to the fact that the combustion is not perfect, an efficiency factor needs to be multiplied by the characteristic velocity ($c^*$) [41]. Finally we need to obtain the Thrust Coefficient $C_F$.

$$C_F = \Gamma \cdot \sqrt{\frac{2\gamma}{\gamma - 1} \cdot \left(1 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right)} + \left(\frac{p_e}{p_c} - \frac{p_a}{p_c}\right) \cdot \frac{A_e}{A_t} \tag{4.5}$$

In Equation 4.5, $p_c$ is the chamber pressure, $A_e$ the nozzle exit area, $A_t$ the throat area, $p_e$ the nozzle exit pressure and $p_a$ the atmospheric pressure. The remaining rocket dimensions are design choices and are discussed ahead in the relevant sections (see: 4.3.2 and 4.3.3). The mass flow $\dot{m}$ is a complex subject for the case of hybrid rockets, it is analysed in depth in Hybrid Rocket Propulsion. It was suggested by van Kesteren in [4] that keeping to maximum thrust is the most beneficial solution for solid rocket propulsion, this would mean keeping the mass flow at maximum as well. However, it has also been suggested that throttling the thrust, by controlling the variation of the oxidizer mass flow, can bring significant performance advantages improvements especially for flight phases in the lower atmosphere

[42]. The reason for this is that thrust variations (even merely cut-off and restart) can reduce the speed and thus the drag losses in a way similar to the benefits of coasting phases for staged rockets. However, this study will focus greatly on the performance of the Air-launched case, where the benefits of throtlling will be less evident since the lower atmosphere (where drag losses are the highest) is actually below the launch point. The investigation of this point is not central for the thesis but made possible by the developed tool, due to time constraints it is my recommendation that this is investigated further in the future.

### Hybrid Rocket Propulsion

Up until this point, the equations presented apply to all forms of rocket propulsion. From here the hybrid-specific characteristics will be developed. What makes hybrid rockets unique is their combination of Liquid and Solid technologies. As such, they have a very characteristic physical constitution with a liquid oxidizer tank and a solid fuel chamber. A schematic based on [23] is found in Figure 4.3. Here the configuration depicted is a tank of pressurised gas. This is one of three common alternatives, the other two being a pump system and a self-pressurized oxidizer tank. For simplification, in this section and those that follow the different mass flows are referred using $m$ and an appropriate subscript instead of $\dot{m}$. As explained in Subsection 4.3.1 the main driver behind the propulsion is the mass flow rate ($m$) with the pressure difference ($p_e - p_a$) accounting for the rest. It is then necessary to analyse how the mass flow rate behaves. The mass flow originates from a combination of the oxidizer flowing into the fuel grain and the fact that the grain itself is burning away. Hybrid Rocket grains usually burn radially, according to the shape of their port. The port is the orifice in the grain through which the oxidizer flows. There can be multiple ports in a grain. A defining parameter of any grain is therefore the port area, $A_p$, which is a measure of the space the oxidizer has available to travel through. There are two main contributions to the overall mass flow, beginning with the fuel mass flow rate ($m_f$) which is dependent on the regression rate ($r$) which is described in Equation 4.6 from [23].

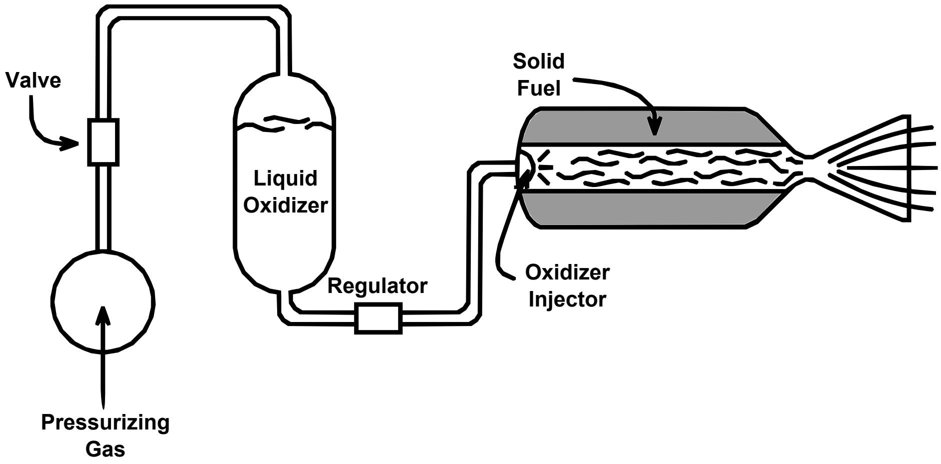


Figure 4.3: A schematic of a generic Hybrid Rocket propulsion system - Adapted from [23]

$$m_f = \rho_f \cdot r \cdot S_b \tag{4.6}$$

Here $S_b$ designates the burning surface area, $r$ the regression rate and $\rho_f$ the fuel density. No dependence on the position along the grain is included here, since from the investigation in the literature study that preceded this thesis, [7], we find that a simple power law for regression rate modelling is standard. It is worth noting that most references point out that using $G_{ox}$, instead of $G$, loses validity for O/F ratios close to, or smaller than, 1.0 [32]. Equation 4.7 is then established to create a dependence between the fuel regression rate and the oxidizer mass flow ($m_{ox}$) [23].

$$\bar{r} = a_o \cdot {G_{ox}}^k \tag{4.7}$$

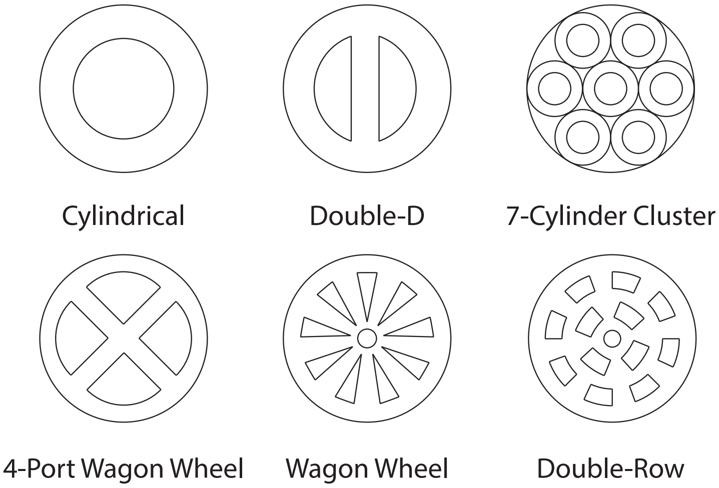


Figure 4.4: Examples of grain geometries of solid fuel grains for hybrid rocket engines - Adapted from [23]

In Equation 4.7, $\bar{r}$ designates the averaged regression rate over the length.

$$G_{ox} = \frac{m_{ox}}{A_p} \tag{4.8}$$

The parameters $a_0$ and $k$ are an empirical coefficient and power, respectively, that model the regression of a particular fuel-oxidizer pair. $G_{ox}$ is defined as the ratio between the oxidizer mass flow rate $m_{ox}$ and the port area $A_p$ [23]. Returning to Equation 4.6 and substituting Equation 4.7 we find Equation 4.9.

$$m_f = \rho_f \cdot a_o \cdot {G_{ox}}^k \cdot S_b \tag{4.9}$$

$G_{ox}$ is dependent on the port area ($A_p$) and on oxidizer mass flow ($m_{ox}$) as seen in Equation 4.8. The port area is dependent on the grain configuration. Historically, geometries with multiple ports per grain have been used to get hybrid performance to keep up with solid technology in terms of regression rate [33, 43, 44]. The port area, $A_p$, can be chosen through the casting of fuel grains with different port geometries - some examples of grain geometries can be found in Figure 4.4. The evolution of the port area, $A_p$, along time is therefore an important aspect as it modifies the mass flow significantly. In fact, the evolution of the grain along time affects Equation 4.9 twofold: through $G_{ox}$ and $S_b$. For this project, due to time constraints, only cylindrical grains are analysed. In cylindrical grains, $A_p$ and $S_b$ can be re-written in function of a single variable: the port radius, $p_r$. Equations 4.10a, 4.10b describe this, and Equation 4.11 describes the evolution, as a function of time $\tau$, of a cylindrical grain, with a uniform, along the length ($L$) burn.

$$A_p = \pi \cdot {p_r}^2 \tag{4.10a}$$

$$S_b = 2 \cdot \pi \cdot p_r \cdot L \tag{4.10b}$$

$$p_r(\tau) = p_r(0) + r \cdot \tau \tag{4.11}$$

The oxidizer mass flow is left as an independent variable, to be inferred from the design parameters of each rocket. Different approaches could have been taken and this is ultimately a choice of how to implement these models. This implementations and the motivations behind it is discussed in Chapter 7. In this model, the only parameters that remain undefined are $a_0$ and the exponent $k$. These are experimental parameters adjusted individually to each fuel-oxidizer pair. In [32] theory-based regression rate laws are analysed with complex parameters included that delve into the finer details of hybrid combustion simulation. An approach of this type is not the point of this work and priority will be given to experimentally proven empirical laws as those in [23] meaning that these will come from literature (see 4.3.1). In Equation 4.12 is then established an expanded view on Thrust including Hybrid-only relations for the definitions of the mass flows, it derives from Equation 4.1 and Equations 4.8 and 4.9.

$$T = \left( m_{ox} + \rho_f \cdot a_0 \cdot \left( \frac{m_{ox}}{A_p} \right)^k \cdot S \right) \cdot C_F \cdot c^* \tag{4.12}$$

Even though not explicitly represented in Equation 4.12, the chamber pressure $p_c$ is one of the crucial design variables for the modelling of any rocket. In the particular case of Hybrids, the design $p_c$ has more implications than merely the thrust it yields. It will also determine whether or not certain feed system components, such as pumps and pressurizing gas tanks, are needed. This has significant impact in the total GTOW (Ground/Air Take-Off Weight) which means it has to be included in the optimization through the mass and geometry modules. This is further developed in subsection 4.3.2.

### Propellants

The choice of which fuel-oxidizer combination to use is an important one as it has implications on both the grain regression law to be chosen and on the thermodynamic properties that need to be computed before hand. The Literature Review showed that the biggest problem with classic Hybrid Rockets propellants was the low regression rate [33]. In the paper by DARE, [14], a student rocketry society at the TU Delft, on the Stratos II fuel studies, the problem with the low regression rate is described as follows:

> Early investigations found that the low regression rates of traditional hybrid fuels such as high-density polyethylene (HDPE), poly(methyl methacrylate) (PMMA) and hydroxyl-terminated polybutadiene (HTPB) drive combustion chambers towards long and slender designs or multi-port configurations. Slender combustion chambers become less and less practical with increasing thrust level due to their unfavorably low volumetric efficiency in a rocket. Multi-port designs are practical, however they result in high sliver fractions of residual fuel as these motors need to be shut down before all fuel is burnt [14].

Classical Hybrids traditionally used HTPB for the fuel, the oxidizer that was used in these cases varied from Liquid Oxygen, Gaseous Oxygen or more commonly Hydrogen Peroxide, this is documented in more recent experiments: HTPB-$N_2O$ in [42, 45] and HTPB-$H_2O_2$ in [12, 46]. HTPB is credited with ease of availability, long-time usage (experience) and ease of manufacturing [46]. Increased safety was cited as one of the reasons for which it was chosen for its commercial implementation in the Virgin Galactic Spaceline [45]. The combination HTPB-LOX was used in [16] citing it as a popular combination in classic hybrid engines. However, there is consensus that the Paraffin Wax and PMMA (Plexiglas) fuels provide higher regression rates [14, 33, 46–49] with [33] even citing that the regression rate increases up to 5 times when compared to HTPB. Paraffin Wax was used in combination with $H_2O_2$ in [17] which praises $H_2O_2$ (Hydrogen Peroxide) as the oxidizer choice due to the fact that is has been used for a long time and that it is reliable (consistent with the information NAMMO provided). Paraffin wax is also used in [47, 49] but in conjunction with Nitrous Oxide. An example was also found for Liquid Oxygen (LOX) and Paraffin Wax which shows very high performance [48]. It seems that the paraffin yields much higher regression rates which in theory would present with a much more efficient solution, especially when combined with LOX. However, paraffin-based fuels have the disadvantage of a lower mass density, poor structural properties and high ideal oxidizer-to-fuel ratio [14].

**Propellant Combination Choice** For the tool to undergo Verification and Validation experimental data provides the ideal benchmark to do so. As such, due to cooperation that provide the required data, the chosen propellants for the initial implementations are those by NAMMO and DARE. Nammo, in their proprietary fuel HTPB/C makes use of additives including paraffin and aluminium that improve the regression rate [12] in the same way as discussed in Propellants. DARE's Sorbitol-based solid fuel also makes use of the same benefits of additives, but its choice mainly fell on the ease-of-use and handling safety aspects.

1. NAMMO: 87.5% Hydrogen Peroxide with HTPB-C
2. DARE: Nitrous Oxide with Sorbitol

**Thermodynamic Properties and Propellant Simulation** The approach to determine the thermodynamic properties of the combustions is to pre-compute an extensive table with Pressure, Specific Heat Ratios and Combustion Temperatures in the appropriate ranges - similar to the approach taken in [4, 41]. This is done using CEA (Chemical Equilibrium with Applications, a tool by NASA) and RPA (Rocket Propellant Analysis) are two important data sources for this project, both are further described in 6.3.3. Both are freely available and accredited software used to compute this sort of information. As

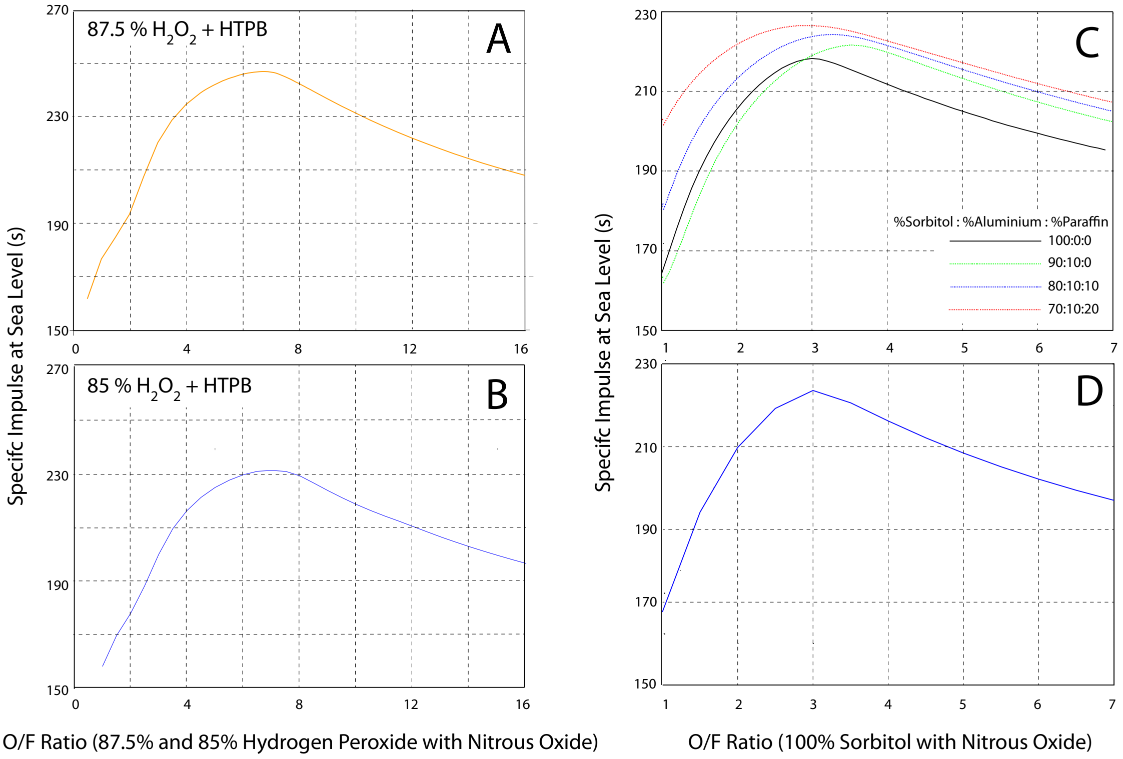


Figure 4.5: **(Left: A & B)** The $I_{sp}$ vs O/F Ratio Plots adapted from the NAMMO paper [12] (top) and the one computed through RPA (bottom). RPA computed only the $I_{sp}$ Curve. **(Right: C & D)** The $I_{sp}$ vs O/F Ratio Plots adapted from the DARE paper [14] (top) and the one computed through RPA (bottom). RPA computed the 100% Sorbitol (100:0:0) Curve. **All data presented is theoretical**.

part of this study simulations were run to try and achieve the same theoretical results by both NAMMO and DARE in their papers ([12, 14]) and to demonstrate RPA as software. van Kesteren used CEA (by NASA) but RPA has a wider range of propellant options and a vastly improved user experience, RPA's data is based on CEA [50] - thus keeping the reliability of the first. Figure 4.5 shows the $I_{sp}$ as a function of the O/F ratio for both the papers and those computed here through RPA by me. All data shown is theoretical, its purpose is to illustrate that the reproduction of the thermodynamic properties of the combustions could be faithfully reproduced (to the same quality as the authors of the experiments) by the tools used in this research. A 30 bar configuration with optimum expansion at sea level was used as per the paper. The curve as computed by RPA for the NAMMO case shows slightly lower values than the one computed by NAMMO itself. RPA only has $H_2O_2$ at 85% available and NAMMO used 87.5% which probably accounts for the difference.

**Regression Rate Law** The paper that most extensively covers regression rate analysis is "Fuel Regression Rate Characterization Using a Laboratory Scale Nitrous Oxide Hybrid Propulsion System" [51]. It disregards longitudinal variations of the regression consistent with the approach in Equation 4.6. The same paper also provides **a systematic way of determining the law from experimental data including error analysis on the process**. The paper tackles the problem in two ways, through space and time averaging and through an averaged $c^*$ method. It also states the values for $a$ and $k$ for the Nitrous Oxide - Sorbitol combination, which, being DARE's case, is highly relevant even though the study used pure Sorbitol and DARE ultimately tests also other variants that include Paraffin and Aluminium. INPE, for instance, provides explicitly $a_0$ as being 0.0344 and $k$ as 0.9593 for their fuel-oxidizer pair. In [51] it is also unfortunately mentioned that the empirical regression rate law presented for Sorbitol-Nitrous Oxide (DARE's combination) is quite low. The Stanford paper, [51], **does not specify how low the correlation it is** but states that the curve is "purely notional as more data points are needed to refine the curve". For this case the values are for $a_0$ 0.286 and for $k$ 0.310 (for $\dot{r}$ in mm/s$^{-1}$ and $G_{ox}$ in

$\mathrm{g\,cm^2\,s^{-1}}$). Unfortunately I couldn't find an independent source for a regression rate law for the NAMMO case, however, NAMMO provides data on the rate itself as a single value (average across length and time) which makes it possible to test the case anyway. The influence of this on the model is reflected on the estimation of the amount of fuel required and the sizing of the accompanying structures - namely the case. The precise influence and its effects can be inferred from the implementation discussion in Chapter 7. The most relevant information from [51] is shown in Table 4.2. Here HDPE stands for high density polyethylene and PMMA for polymethyl methacrylate. .

| Fuel | a | k |
|---|---|---|
| HDPE | 0.104 | 0.352 |
| PMMA | 0.111 | 0.377 |
| HTPB | 0.198 | 0.325 |
| Sorbitol | 0.286 | 0.310 |

Table 4.2: Regression rate law coefficients for the burning of the fuels in the Fuel column with Nitrous Oxide. Vales assume regression rate $r$ to be in $\mathrm{mm\,s^{-1}}$ and $G_{ox}$ to be in $\mathrm{g\,cm^{-2}\,s^{-1}}$. Adapter from [51].

### Variable Oxidizer-to-Fuel Ratio Effect

The **O/F**, or Oxidizer-to-Fuel ratio, indicates the proportion between the propellants used in a particular combustion. Altering the ratio changes the thermodynamic properties of the combustion. Having an optimal O/F ratio yields the greatest amount of energy possible for a particular oxidizer-fuel pair. In the case of Hybrid Rockets, as the oxidizer is inserted and the fuel grain is burned through, **radially** (not axially), the burn surface area, $S$, changes. With a changing $S$, systems that maintaining $m_{ox}$ constant will have $m_f$ changing over time (see Equation 4.9). A changing $m_f$ and a constant $m_{ox}$ cause the O/F ratio to *shift*. This effect is know as the O/F shift. The uneven burning caused by boundary layer and turbulence effects also adds to this shift in the O/F ratio [23], i.e. it is not constant neither along the grain nor along time [23]. To compensate for the uneven burning the approach will be to add a small percentage of "extra fuel" set by the user in a case by case basis. In [41] it is shown the dependence on the O/F ratio is quite significant for the Adiabatic Flame Temperature of a burn similar to those considered in this project (Figure 4.6). The chamber temperature plays a fundamental role in the calculation of the thrust. This influence occurs through $c^*$ and $C_F$ which depend directly on $T_c$ and $\gamma$ respectively (Equations 4.3 and 4.5). The variation with the thermodynamic properties for $C_F$ is much weaker than for $c^*$. The dependence of $C_F$ on the thermodynamic properties is only through the specific heats ratio that varies only a few percent and therefore does not have a very significant impact. This effect is included in the simulations through the instantaneous (in every step) recalculation of the ratio $\left(\phi = \frac{m_{ox}}{m_{fuel}}\right)$ which in turn comes from the reassessment of its individual parameters. The implementation of this effect onto the tool is discussed in Section 7.2.2. The company NAMMO mentioned that they have achieved throttling ratios of 40%; this could be used to avoid the shift's effect. However in [34] it is mentioned that solving the shift is a negligible performance gain compared to other effects like Nozzle Erosion or Combustion Efficiency. Another solution to minimize the effect of the O/F shift is to experiment with different grain configurations, particularly those with a constant burning surface. The O/F shift is also described in [52] where it is praised as a form of limiting the maximum acceleration (since it decreases with the O/F shift that occurs naturally) for payloads that are sensitive to it.

## 4.3.2. Geometry

This section deals with the dimensions and placement of the individual components that make up the vehicle. Following the architecture for the tool from van Kesteren, the geometry model only addresses volume and placement concerns.

### Components to consider

For this project the following structural components are considered:

1. Oxidizer Tank

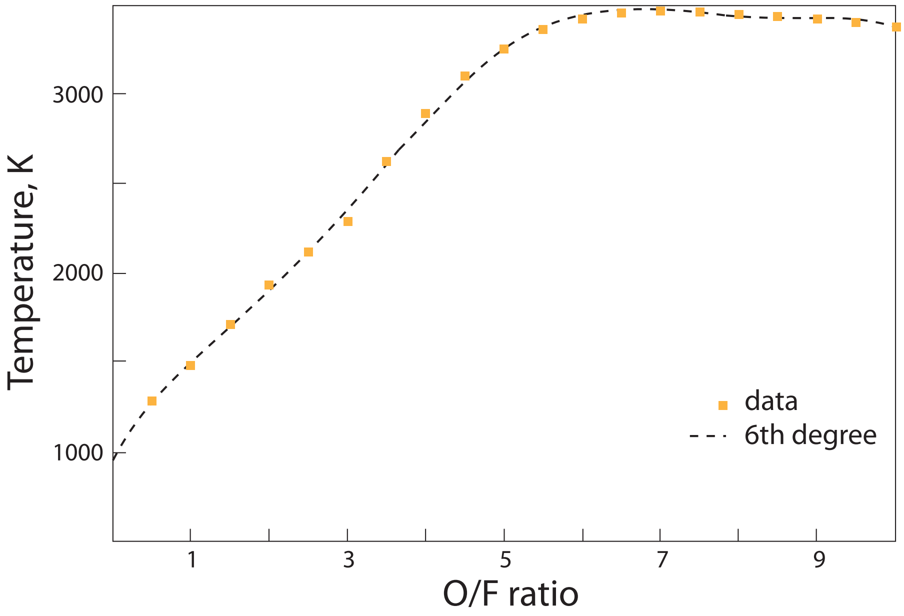


Figure 4.6: Adiabatic Flame Temperature vs. O/F Ratio for the $N_2O$-HTPB O-F pair [41].

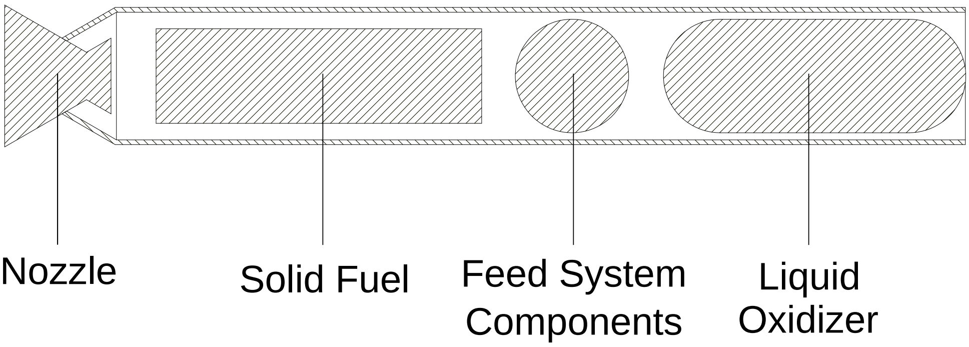


Figure 4.7: Schematic of the main components of a Hybrid Rocket Stage. In this example schematic, the shell is detached from the individual components, this is not the most efficient solution in terms of mass and is not the one ultimately used in the model. This schematic is presented as an illustration of the approach to the considered components.

2. Feed System Components

3. Solid Fuel Tank

4. Nozzle

5. Rocket Frame (Interstages, outer shells and similar structural elements)

The reason for subdividing the rocket into these specific components comes from the work done in the literature study [7], where it is concluded that a similar structure was used for the solid case by van Kesteren in [4] and well motivated in his work. The hybrid rocket model considered in this tool will model each individual stage (component-wise) as depicted in Figure 4.7. Requesting 3-stage Hybrid Rocket from the tool, for example, yields the design depicted in Figure 4.8. Here $L$ designates the length of a particular element with the appropriate subscript. Subscripts **conv** and **div** designate the convergent and divergent parts of the nozzle respectively. The subscripts **O& F** are the distance between the tip of the oxidizer tank and the end of the fuel grain. Lower case $d$ is the distance between each stage, this is different from the length of the interstage, $L_{interstage}$. The details of these lengths are analysed in the equations that follow in this section. VEB designates the Vehicle Equipment Bay.

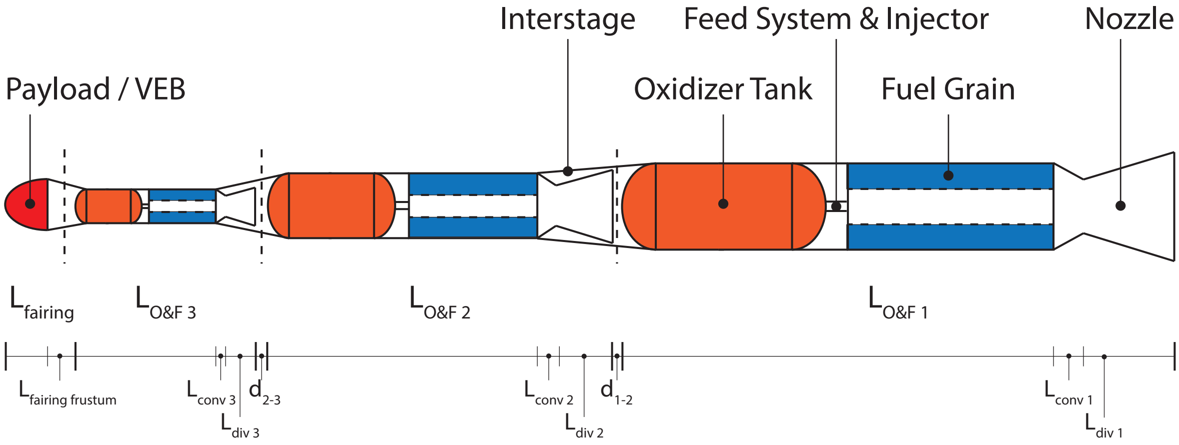


Figure 4.8: Hybrid Rocket Model used by the simulation software when a 3-stage non-winged vehicle is requested.

### Computing the total length

The overall length of the launcher is defined in Equation 4.13.

$$L_{launcher} = L_{fairing} + \sum_{i=1}^{n_{stages}} \left(L_{case_i} + L_{conv_i} + L_{div_i}\right) \tag{4.13}$$

All of the parameters in Equation 4.13 are represented in Figure 4.8 with the exception of the $L_{interstages}$ which is represented through the white unlabelled spaces in the guide in that same figure. These are to account for extra structural length required by the mounting of the stages, the nozzle integration and parts such as aft-skirts. The values for this follow the model by van Kesteren in [4].

### The Rocket Case / Rocket Structure

There are two options when designing a rocket's outer structure.

1. A case is used as an outer shell that envelops the entire rocket, designed to withstand the external forces.
2. Each component is designed to withstand the external forces. Structural elements are merely used between components and at the inter-stages to ensure surface smoothness across the vehicle.

For both choices the structure is assumed to be a hollow cylinder for which length and thickness must be determined. The thickness depends on its material and the loadings it is subjected to. Since this is the main factor in determining the mass of the case, the thickness is addressed in the Mass subsection (see Subsection 4.3.3). If **Option 1** is chosen, the total length is the result of Equation 4.14.

$$L_{case} = L_{oxi\ tank} + L_{fuel\ tank} + L_{feed\ system} + L_{inter-stages} \tag{4.14}$$

In case the walls correspond to the tanks' walls, **Option 2**, the length of the required case corresponds to that of the feed system components and the inter-stages, as per Equation 4.15.

$$L_{case} = L_{feed\ system} + L_{inter-stages} \tag{4.15}$$

The **second option** is usually the more modern and **weight-saving solution** and thus the optimal one. This option is the one depicted in Figure 4.8.

### Fairing

The fairing keeps most of the specifications based on the same motivations as in [4]. It will remain an aerodynamic ogive shaped nose cone. As also mentioned in [4] there are no small weight class payload (under 10 kg) standards. Therefore the same dimensioning will be used: $0.43 \times 0.28 \times 0.19$ m and the payload has to fit in the fairing as described. The same reasoning will be followed for the higher payload classes with the 2000 kg range using the dimensions for the VEGA with a length of 3.52 m and a minimum diameter of 2.60 m.

### Aerodynamic Lifting Surfaces

Since the Aerodynamic Lifting Surfaces does not influence in any way the choice of engine, most of the considerations from [4] remain valid. A taper ratio of 0.16 and chord length ratio of 0.40 is used based on the Pegasus XL and the HLS concept. The positioning of the surfaces will also be the same as shown in Figure 4.9. The accuracy of this model is not explicitly stated in [4]. However it is mentioned that the model was adopted in order to reduce the complexity and size of the database required for the study of the aerodynamic interactions. This model remains valid since there are no differences between Hybrid and Solid technologies in this respect, thus it was kept for this project.

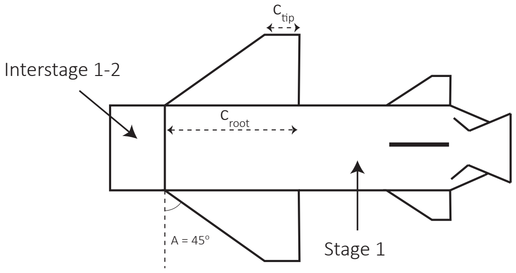


Figure 4.9: Geometry of the first stage with Aerodynamic Lifting Surfaces included [4]

### Nozzle

The nozzle is composed by the convergent and divergent sections as in [4, 23], the calculation of these sections is outlined in Equations 4.16 and 4.17.

$$L_{con} = \frac{D_{fuel\ tank} - D_t}{2 \cdot \sin\beta} \cdot \xi \tag{4.16}$$

$$L_{div} = R_u \cdot \sin\theta + \frac{R_e - R_t - (R_u - R_u \cdot \cos\theta)}{\tan\theta} \cdot \xi \tag{4.17}$$

In Equations 4.16 and 4.17, $\xi$ is a correction factor, $R_u$ is the longitudinal radius of the throat, $R_e$ the exit one and $R_t$ the cross-sectional one. Typical values of $\beta$, the convergence angle, are 30-60°, for $\theta$, the divergence angle, it is usually around 12-18° [4]. These dimensions are illustrated in Figure 4.10. Here a clear assumption is made that the nozzles can be modelled in the same way for the Solid and Hybrid cases. This assumption deserves further investigation but for now is deemed as acceptable since, for the models used throughout this work there is no explicit reason why the nozzle design should not be the same in both hybrid and solid rockets. The particulars of hybrid modelling seem to not have any different direct effect on the nozzle's characteristics.

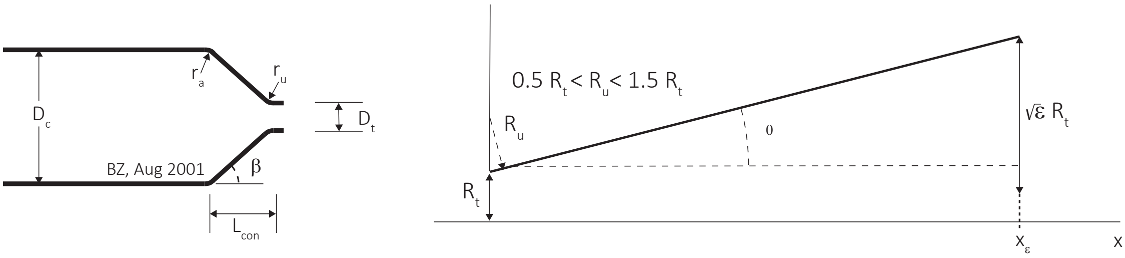


Figure 4.10: Nozzle dimensions and geometry. $D_c$ is the chamber diameter, $\epsilon$ the expansion ratio, $r_a$ and $r_u$ are the radii of curvature at their respective points, and $R_t$ and $D_t$ and the throat's radius and diameter respectively. Figure adapted from [23]

.

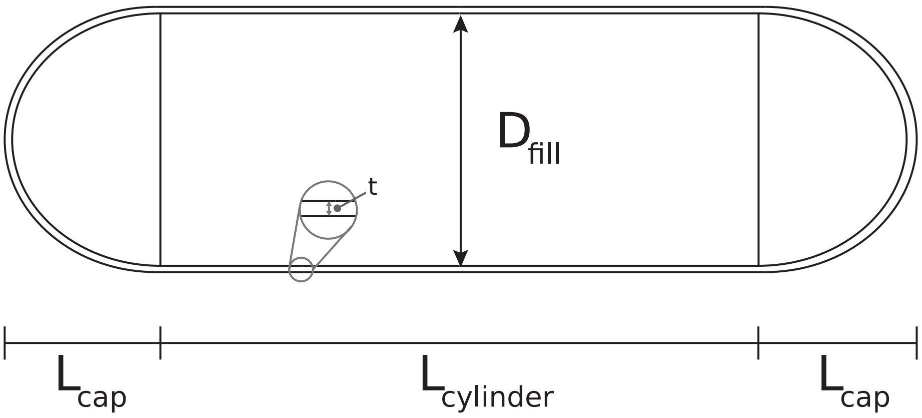


Figure 4.11: Simplified Schematic of an Oxidizer Tank. $L_{cylinder}$ and $L_{cap}$ are the lengths of the cylindrical section and cap respectively. $D_{cap}$ indicates the diameter of the caps.

### Propellant Tank Sizing

The required burn time, $t_b$, a result of the optimization of the launch vehicle, can be used to determine the required amounts of oxidizer and fuel. In the case of SRP's where there is only one propellant, the burn time is a direct measure of the total mass for a given mass flow rate. However, in the cases of Liquid and Hybrid rockets it is the *mixture of fuel and oxidizer* that is burned. As such, to determine the total mass, $t_{burn}$ must be multiplied by the sum of both mass flow rates - Equation 4.18.

$$M_{propellants} = \left(m_{ox} + m_{fuel}\right) \cdot t_{burn} \tag{4.18}$$

Equatio 4.18 is only strictly valid when both mass flows, fuel and oxidizer, are constant. For simplification, in the implementation the maximum values are used. After computing $M_{propellants}$, the mass of the oxidizer and fuel can be derived individually, using the prescribed O/F ratio, as seen in Equation 4.19.

$$\begin{aligned} M_f &= \frac{M_{propellants}}{\phi_{mass} + 1} \\ M_{ox} &= \frac{M_{propellants}}{\frac{1}{\phi_{mass}} + 1} \end{aligned} \tag{4.19}$$

Here $\phi_{mass}$ designates the *initial* O/F ratio. $M_{ox}$ and $M_f$ the total masses for the oxidizer and fuel respectively as required by the burn time. The ratio as used here is the design ratio and should be calculated taking into account the fact that it will change over time (e.g. by adding extra propellant). Note that $M_{ox}$ as defined in Equation 4.19 is the exact required oxidizer mass as required by the $\Delta V$ of the target launch solution. In [23] more mass is added to this value to compensate for various losses like expulsion losses, bail-off losses, a margin for ($M_{error}$) and a design margin ($M_{margin}$) are then also added to the total. The last two are usually a contribution of under 5% to the total error (with the margin relying on a typical efficiency of 0.97 - 0.99). Bail-off and Expulsion vary significantly, with [23] citing up to 25% in some cases, the reasons for this and a deeper investigation on this topic was not conducted as part of this project due to time-constraints. The literature information cited is taken at face-value. Setting the losses will be left to the user as historical data for hybrid rocket is very scarce to have properly motivated default values.

**Liquid Oxidizer Tank** The required amount of oxidizer will influence directly the length of its tank. Oxidizer tanks are generally a combination of cylindrical sections with spherical end-caps [31], there is also the possibility to use elliptical caps or other similar shapes, for simplicity only spherical ones are considered here. Figure 4.11 depicts this geometry. Note that, the cylindrical component's length can be zero, i.e. the tank will become spherical. The ratio between the $L_{cylinder}$ and $L_{cap}$ is known as the **fineness ratio** of the tank [31]. The volume of such a tank can then be easily computed through the required oxidizer mass ($M_{ox}$). Equation 4.20 gives the volume of the oxidizer tank. Note that $M_{ox}$ is composed both by the contributions from Equation 4.18 and all of the losses mentioned in Propellant Tank Sizing.

$$V_{ox} = \frac{M_{ox}}{\rho_{ox}} \tag{4.20}$$

The oxidizer can only occupy the inner volume of the tank. This inner volume is determined by the required wall thickness, $t_{tank\ wall}$, which is determined by the stresses imposed on the tank itself. As for the tank itself this will be based on a simple stress model from [31]. For the cylindrical part the limiting factor is deemed to be the hoop stress and not the axial one (for the spherical component these coincide). The required thickness for the spherical part is taken as one half of the thickness of the cylindrical part [31].

$$t = MEOP \cdot DF \cdot FoS \cdot \frac{D}{2 \cdot \sigma} \tag{4.21}$$

In Equation 4.21 $D$ represents the diameter (of both the cylinder and the sphere) and $t$ a thickness, in this case, the tank thickness. $MEOP$ is the maximum expected operating pressure $DF$ a Design Factor and $FoS$ the Factor of Safety. Both factors come from the launcher's design and from reference literature respectively. The stress case load, $\sigma$, depends on the material used, it will be checked for both the *yield* and *ultimate* strengths of the chosen material. The highest resulting thickness from both results will be the one used. $sigma$ will, in the models, be set by the user. The determined thickness will, nonetheless, remain as a form of determining the geometry of the tank that will add to the overall result for this section. The thickness will contribute as well for the mass (see 4.3.3). Following from the geometry in Figure 4.11, and with the thickness now defined, the tank volume can be computed as per Equation 4.22.

$$V_{ox} = \frac{\pi \cdot (D_{fill} + 2t)^3}{6} + L_{cylinder} \cdot \frac{(D_{fill} + 2t)^2 \cdot \pi}{4} \tag{4.22}$$

Here, $D_{fill}$ is the diameter of the volume of fluid that fills the tank. With a fineness ratio chosen then Equation 4.22 is fully determined and the total length can be calculated as per Figure 4.11 through Equation 4.23.

$$L_{ox} = 2 \cdot L_{cap} + L_{cylinder} \tag{4.23}$$

Note that the length calculated in Equation 4.23 is subject to the optimizer's iterative process. The length is also influenced by whether a pump system or pressurized gas solution is chosen (see Subsubsection Feed System Components and Requirements).

**Solid Fuel** The geometry of hybrid rockets is more complex than the SRM case. Since hybrid systems include a liquid oxidizer and a solid propellant, the system cannot be as simplified as it was. There are however some common points which will facilitate the work that is necessary for this project since this allows for some of the existing code to be reused. In [23] it is stated that the design of the solid fuel tank component can be done like any Solid Rocket Booster (SRB). However the equations used in [4] for the calculation of the length of the Solid Fuel part of the stage can be modified for our case. This is shown in Equation 4.24 where $A_f$ is the cross-sectional of the solid fuel grain itself.

$$L_{fuel\ tank} = \frac{M_f}{\rho_f \cdot A_f \cdot FF} \tag{4.24}$$

As seen in Figure 4.4 the grain geometries are a relevant issue when designing the rocket and thus the size of the case will be highly dependent on it. $L_{solid\ tank}$ defines the case's length and $FF$ is the Fill Factor and defines how much of the volume of the case is actually occupied by solid fuel. The Fill Factor is chosen by the user and varies with the engines and cases being used. For an idea of the magnitude of such a value, in [4] a value of 95% was used. As for the Liquid Oxidizer tank, the thickness is also a concern. For the level of complexity of the models used here, the methodology is mostly the same as for the Liquid Oxidizer tank. The difference here being that usually solid fuel tanks are simply cylindrical with no spherical end-caps. This is explained by the nature of Solid fuels which are usually cast into a simple cylinder unlike fluids which are traditionally store in cylindrical tanks with sperical end-caps. Thus Equation 4.21 still holds but now there is no need to calculate the half thickness that would apply for the spherical cap. End-caps or equivalent mounting structures that interface with the oxidizer's injector or the nozzle are considered for mass purposes but do not affect the length significantly.

**Pre-Combustion Chamber** In the [12] the company Nammo also considers a post-combustion chamber sized $1D_{chamber}$ in length, this will be taken as the reference for the combustion chambers for simplification even though one could use the required $L^*$ to determine this length as for a LRE. In the tool this will be a user-set parameter and will add to the variable $L_{solid\ tank}$ as component of this structure.The thickness of these components is computed on relatively simple stress models, and its length is simple the sum of the parts that they hold multiplied by a quality factor from historical data. Here the considerations in [4] for solid rockets are mostly valid and will be the ones used.

### Feed System Components and Requirements

Hybrid rockets inject the liquid oxidizer into the combustion chamber containing the solid fuel grain. This fact means that, unlike SRM's, there is a need for some feed system components. The main consideration is the assessing the need for an extra tank for a pressurant gas, or pumping systems. As a general rule pumping systems are lighter than pressurizing gas tank systems that involve an extra tank to hold the pressurant[8]. Since the tool will aim to minimize GTOW, and since no cost information is available here, the optimum launch between a pump-fed system and a pressurized gas tank one will always be the pump-fed one. However, since pump-fed systems are more complex (more moving parts, higher leak risk and higher fail risk overall) it could be argued that, all things considered, the pressurized tank would be the better choice. A trade-off where these practical aspects are included is beyond the scope of this investigation, thus the GTOW minimization remains the deciding factor.

**Dimensioning a Pumping System** Following from [8], the dimensions of a rocket pump are mainly a function of how much power is needed. The fact that the conclusions from [8] come from historical data, the regressions might have limited applicability depending on the size of the pumps. Since the pressure will depend on the Operating Pressure of the rocket which, in this project, is a design variable, the choice of the type and size of the pump will be estimated by the optimizer. At this stage, the first parameter to define is the turbopump rate power, defined as the pump hydraulic power divided by the pump efficiency and the mechanical efficiency through Equation 4.25 from [23].

$$P_h = \frac{m \cdot \Delta p_t}{\rho} \tag{4.25}$$

Here $m$ is the mass flow rate through the pump, $\Delta p_t$ is the pressure difference and $\rho$ the density of the fluid. For the case of the hybrid rocket, $\Delta p_t$ is the difference between the pressure in the liquid oxidizer tank and the operating pressure inside the combustion chamber. The mass flow rate, $m$, and the density, $\rho$, are those of the oxidizer. The relations for these parameters are those in [8], here stated in Equation 4.26. The point of these relations is to yield a cylindrical outer envelope for the pump size thus giving an estimate of how much space it will occupy inside the stage. With the volume of this cylinder being the one yielded by the "Power / Power Density" ratio from Equation 4.27 from [8].

$$\begin{aligned} D &= \sqrt[3]{\frac{4 \cdot P}{P_\delta \cdot (L/D) \cdot \pi}} \\ L &= \frac{\pi \cdot (L/D)^2 \cdot P_\delta}{4 \cdot P} \end{aligned} \tag{4.26}$$

$$\frac{P}{P_\delta} = L \cdot \pi \cdot \frac{D^2}{4} \tag{4.27}$$

In Equations 4.27 and 4.26, $L$ is the Length of the Pump and $D$ its Diameter. $P_\delta$ is the Power Density measured in $\mathrm{MW\,m^{-3}}$. The $L/D$ ratio depends on whether the pump is Direct Drive/Geared or Dual Shaft for which the range of values is 1-2 and 0.8-1.3 respectively. [8] presents a set of relations of the estimation of the Power Density included in Table 4.3.

**Pressurized Gas Container** In case a pump is not used, a pressurized gas container is the alternative. Here there are two alternatives, either the system is self-pressurized or there is the need for an external tank with a pressurant gas. In the cases of the Hybrid and the Liquid rockets, the propellant must get to the combustion chamber at the required operating pressure. The storage pressure has to

| Pump Type | Relation | Range |
|---|---|---|
| Direct drive or geared | $0.5085P + 1.3877$ | 0.2 - 40 MW |
| Dual Shafts | $3.6982P + 12.365$ | 1 - 50 MW |

Table 4.3: Turbo-pump power density $P_\delta$ (in MW m$^{-3}$) estimation relationships ($P$ in MW) - adapted from [8]

be high enough to compensate for the drops and losses along the feed system. To achieve the higher pressure a pressurizing gas can be added to the oxidizer tank. This solution constitutes a blow down system. To measure the amount of pressurized gas required we calculate the Blow-Down Ratio, $R$.

$$R = \frac{V_g + V_p}{V_g} \tag{4.28}$$

Where $V_g$ is the volume occupied by the pressurizing gas and $V_p$ that of the propellant. This value is usually around 1.5 to 2.5, thus comprising 40-67% of the initial volume [23]. To this we must also add the vapour pressure ($p_v$), its proportion varies from propellant to propellant. This is defined in Equation 4.29 [23].

$$\ln(p_v) = \frac{A - B}{T + C} \tag{4.29}$$

The parameters A, B and C can be found in tables such as those from NIST. Total pressure is obtained by adding the hydrostatic pressure to the vapour one. The total pressure inside the tank can then be calculated (or adjusted) to comply with the operating pressure requirements. With that defined, the volume and the corresponding density can be used to estimate the final mass of the propellant, the required added gas and the new required tank thickness (again per Equation 4.21).

### 4.3.3. Mass

The mass model follows from the computation of the mass of the components used for the Geometry model in Subsection 4.3.2. However, to continue with the code and architectural consistency of the tool, an analysis of the approach in [4] is highly relevant. Figure 4.12 is an adaptation of the mass breakdown from [4] with the necessary modifications for the Hybrid Rocket case but highlighting the commonalities with the Solid Rocket case. The GTOW is then the sum of all the individual components from Figure 4.12, thus it is necessary to address each one of them.

#### Stage Mass

This subsection deals with the estimation of the mass of all the subcomponents of each Rocket Stage. These are mostly derived from the geometry module.

**Hybrid Rocket Motor Components** The main components of the HRM are the Nozzle, the Solid Fuel Casing and the Liquid Oxidizer Tank. The outer case is treated as a component of the stage itself.

**Nozzle** The Nozzle for a Hybrid Rocket can be modelled identically to the Solid Rocket Motor case. In view of this, the approach in [4] can be kept thus simplifying the process to upgrade the tool and to conduct the work. The method used is a simple historical data regression divided between a large motor range ($T_{vac} > 200$kN) and small motor range ($T_{vac} < 200$kN). The data for each thruster range is then subdivided in two ranges: thrusters with TVC and thrusters without [4]. For the regression to be used directly it would be necessary to have concrete evidence that Hybrid Rocket Motors' nozzles could be modelled like those for SRM without any modification. At first glance, this does seem to be the case since the processes and design variables are similar. The regressions are shown in Equations 4.30 through 4.32 for each case. The size of the datasets used for the following relations and their validity is discussed in depth in [4].

1. Large rocket motors ($T_{vac} > 200$kN):

$$m_{nozzle} = 0.0006T^2 - 0.3214T + 263.82 \tag{4.30}$$

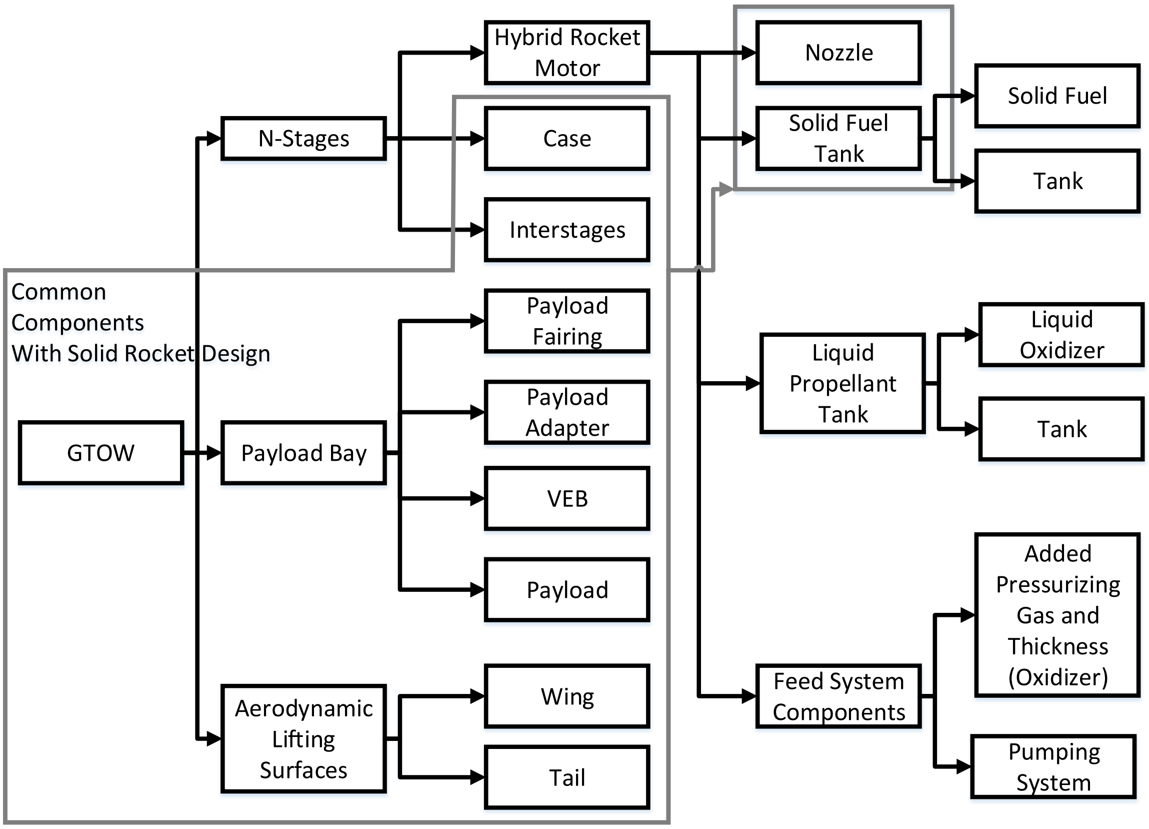


Figure 4.12: Mass breakdown for the model to be used highlighting the commonalities with the Solid Rocket case from [4]

2. For small rocket motors ($T_{vac} < 200$kN) with TVC:

$$m_{nozzle} = -0.0018T^2 + 1.004T - 1.942 \tag{4.31}$$

3. For small rocket motors ($T_{vac} < 200$kN) withou TVC:

$$m_{nozzle} = 0.1605T^{1.2466} \tag{4.32}$$

Equations 4.30, 4.31 and 4.32 are taken from [4] and have an $R^2$ of 0.9991, 0.9425 and 0.9456 respectively. The thrust, $T$ is in kN and nozzle mass, $m_{nozzle}$, in kg.

**Solid Fuel Elements** In this model the mass for the Solid Fuel Elements is assumed to be the sum of the masses of the fuel grain, the case holding the fuel and the post-combustion chamber. The post-combustion chamber is assumed to be of the same material as the case. The thickness is determined in the mass model, with the material being a user input. The material determines the thickness through the expected operating pressure ($p_c$) and its properties, mainly the stress cases (see Equation 4.21 where $\sigma$ is material-dependent). The chosen material's density can then be used (along with the volume inherited from the thickness) to calculate the mass. The mass of the fuel grain is simply the value from Equation 4.18.

**Liquid Oxidizer Tank** As per the Solid Fuel Tank, the principle behind the Liquid Oxidizer tank mass calculation is also a choice of material and determination of the appropriate volume per Equations 4.22 and 4.21. With this, the mass can be calculated in much the same way as it was done for the Fuel Tank. After computing the mass of the tank, the mass of the Liquid Oxidizer itself, calculated through Equation 4.18, must be added to obtain the total mass for the component.

**Feed System Components** The feed system components follow once again the rationales behind the geometry considerations. Mainly if the choice is for a pump then its mass must be calculated. This is done based on [8]. The mass relations from [8] are presented in Table 4.4. If the solution is to use a pressurizing gas or rely on vapor pressure, then the added volume of either solution must be used to calculate the added mass. However, added pressure means that the thickness of the tank must be calculated once more and thus the tanks themselves must be re-dimensioned each iteration.

| TPA Arrangement | Mass Function [kg] | Range [MW] | $R^2$ | RSE | N |
|---|---|---|---|---|---|
| Direct Drive or Geared | $44.396P^{0.836}PR^{0.182}$ | 0.2 - 1.75 | 0.994 | 7.8% | 6 |
| | $3.972E6P^{0.556}PR^{-3.565}$ | 2.25 - 40 | 0.9149 | 26.3% | 6 |
| Dual Shaft | $90.164P^{0.150}m^{0.199}$ | 2 - 40 | 0.910 | 11.7% | 9 |
| | $57.624P^{0.136}m^{0.327}PR^{0.076}$ | 2 - 50 | 0.957 | 9.2% | 9 |

Table 4.4: Mass sizing estimate relationships for turbo-pumps [8]

**Igniter** In Hybrid Rockets, the igniter is often not a component with a significant contribution for the mass of the overall rocket, at least to the precision of this work [23]. In fact, the most common solution is a simple electrically activated ignition or even just a catalyst bed. Some oxidizers like Nitrous Oxide can even act as mono-propellants and self-ignite on contact with the catalyst bed on their own. Electrical igniters have a negligible mass, and catalyst beds have a very small length when compared to the stage's thus, both are disregarded for mass and length considerations.

**Rocket Frame and Interstages** This structure should be dimensioned for each stage as per Sub-subsection The Rocket Case / Rocket Structure as a simple thin-walled cylinder subjected to the various g-loads involved in the launch. The plan here is not to include the optimization of the thickness as part of the variables to be optimized, but to provide a safe guess for the heaviest loading-case. To account for Aerodynamic and Steering Loads the structure's thickness will be simply multiplied by a Factor of Safety ($FoS$). Normally, historical data would be used to select the value for the $FoS$. However, for the case of hybrid rockets this data is not available in a comprehensive and reliable way. Since the main purpose of the study is to compare Hybrid to Solid technology, the Factor of Safety will be set to the same amount as the one used in the Solid Rocket study in [4]: **1.5**.

**Dimensioning a thin-walled cylinder** The following relations will dimension a thin-walled cylinder of an unspecified material which will be left to the user to choose, as such, properties like the Young modulus ($E$) will be left as unknowns. Typical values for loads will be taken from [1] which indicates the values in Table 4.5 as typical for a Launcher System. The loading cases considered for this analysis are as follows:

1. Natural Frequency Rigidity

| Parameter | Typical Value |
|---|---|
| Acceleration | $5-7$ g |
| Vibration | $0.1$ g$^2$ Hz$^{-1}$ |
| Shock | $4000$ g |

Table 4.5: Typical Acceleration Loads on Launch - adapted from [1]

2. Tensile Strength
3. Compressive Strength (with buckling)

The **Natural Frequency rigidity** requirement follows from the fact that the thin-walled cylinder must have a Natural Frequency above a particular value, typically around 10-25 Hz [1]. Here we must consider two separate cases: uniform loading for the structure itself and point-mass loading for the structures above it. For these two cases it follows that each can be subjected to lateral and axial loads. From [1] we obtain the Equations in Table 4.6. By setting the frequency to be higher than a particular amount we can derive a requirement for the cross-sectional area $A$ and thus the required thickness. The next effect to consider is the **Axial Load**, calculated in [1] as the *Equivalent Axial Load* as per Equation 4.33, there $R$ represents the outer radius of the cylinder and $M$ its mass, $P_{eq}$ is the equivalent pressure, $P_{axial}$ the axial pressure.

| Case | Axial Loads | Lateral Loads |
|---|---|---|
| Point-Mass | $f_{nat} = 0.160\sqrt{\frac{AE}{m \cdot L + 0.333 m_B \cdot L}}$ | $f_{nat} = 0.276\sqrt{\frac{EI}{m \cdot L^3 + 0.236 m_B \cdot L^3}}$ |
| Uniform | $f_{nat} = 0.250\sqrt{\frac{AE}{m_B \cdot L + 0.333 m_B \cdot L}}$ | $f_{nat} = 0.560\sqrt{\frac{EI}{m \cdot L^3}}$ |

Table 4.6: Natural Frequency for Axial and Lateral Loads for the Point-Mass and Uniform loading cases [1]

$$P_{eq} = P_{axial} + \frac{2M}{R} \tag{4.33}$$

With the $P_{eq}$ from Equation 4.33 determined, it can be used in a simple compressive strength equation to determine, once more, the cross-section area, $A$, requirement. This is shown in Equation 4.34, where $\sigma_{allowable}$ is the allowable stress.

$$\sigma_{allowable} = P_{eq}/A \tag{4.34}$$

The last effects considered in [1] are *buckling* and internal pressure (hoop stress). However, since the internal pressure is mainly driven by the $MEOP$ and the Storage Pressures, which are in this case contained by tanks that have already been dimensioned, the effect will be disregarded. Buckling is, however, important to take into account. This is defined by the expressions in Equations 4.35a to 4.35c from [1]. Here, $\varphi$ and and $\gamma$ are geometrical factors specific for the case of the thin-walled cylinder. In the stress expression ($\sigma_{cr}$), $t$ defines the thickness of the wall of the cylinder. Here, pressure stiffening effects related to the internal pressure of a liquid tank are neglected in this study.

$$\sigma_{cr} = 0.6 \cdot \gamma \cdot \frac{E \cdot t}{r} \tag{4.35a}$$

$$\gamma = 1.0 - 0.901 \cdot (1.0 - e^{-\varphi}) \tag{4.35b}$$

$$\varphi = \frac{1}{16}\sqrt{R/t} \tag{4.35c}$$

With the stress requirements based on loading factors (typical values also found in [1]) the thickness requirement can be derived and thus, with the material as a user input, so can the mass. This is the way the mass of both the case and the interstages is calculated.

### Payload Bay Mass and Aerodynamic Lifting Surfaces

For the components of the payload bay, the models in [4] will be kept in their entirety. These are here summarized:

1. Fairing: Model based on the wetted area: 12.2kg m$^{-2}$
2. Payload Adapter: $M_{adapter} = 0.004775 \cdot M_{payload}^{1.0132}$
3. Vehicle Equipment Bay: $M_{int} = 0.404 \cdot M_{dry}^{0.6814}$
4. Payload: Two payloads - 10 and 2000 kg are evaluated.
5. Aerodynamic Lifting Surfaces: Model based solely on the area with the mass depending on it by a factor fo 24.4 kg m$^{-2}$.

All of the above relations are from and motivated in [4].

# 5 Multidisciplinary Optimization and Launch Simulation

Multidisciplinary Optimization (MDO) is the process used to test and evaluate each launch solution, attempting to select the best possible one. Though MDO was discussed in depth in [4] there were some significant changes to the approach used there. Since a revision of the concepts surrounding MDO was nonetheless necessary this chapter discusses the topics that were of greater importance for this work in particular. This chapter presents and discusses the approach to implementing MDO in this research (in 5.1) including the theoretical concepts surrounding the Optimization Processes (in 5.1.1). An overview of the models used for the launch simulation is presented in section 5.2.

## 5.1. Multidisciplinary Optimization

Multidisciplinary Optimization is a technique by which the optimization methods are used to solved design problems associated with multiple disciplines. In the case of this work, trajectory optimization and vehicle design are iterated on until a final solution is reached. A solution, in this context, is a vehicle-trajectory pair that successfully reaches a given target orbit. Several optimization techniques can be used to reach what is considered the optimum solution, this chapter section discusses the ones considered and applied here.

### 5.1.1. Optimization Process

The approach in this work utilizes a global continuous optimizer that tries to achieve the parameter set that has the best value for the fitness function. This was the approach in the original version of the tool. However, since some changes were required to the software (see Chapter 6) a small investigation into the optimization process itself had to be done. A brief discussion of MDO as it applies to this research is presented here. The two main concepts here are the **optimizer** and the **fitness function**.

1. Global Optimizer: process through which all the parameters that determine a rocket and its launch trajectory are tuned to yield approximately the best solution

2. Fitness Function: function that defines the quality of each parameter set

The global optimizer is, therefore, an algorithm of set of algorithms that take a parameter set and evaluate its Fitness Function value. The fitness function in the case of this research is the quality of the launch solution. The quality can be evaluated in different ways. The final orbital parameters are the way to measure how well the goal was achieved for most cases. An analysis of the orbital parameters of a particular body is used to measure how close it is to the target orbit. However, getting an optimizer to yield solutions that check for proximity in multiple orbital parameters turns the problem into a multi-objective optimization problem. In measuring orbital injection accuracy this is unavoidable, but not all the orbital parameters are used in the measurement. Here only eccentricity and semi-major axis are used. Since solution quality is not only measured by how well the satellite was injected into orbit, other parameters like the GTOW are also taken into account. Instead of solving a true multi-objective optimization problem, the optimizers in this research always condense the solutions' properties into one single function value that evaluates the quality as a whole.

### 5.1.2. Constraints

Optimization constraints are conditions on both the solution space and its properties that ensure the solutions are realistic or implementable and, ultimately, comply with mission requirements. There are two main kinds of constraints to take into account:

1. **The search-space's boundaries**
2. **Specific characteristics of a solution** that make it undesirable, these can only be checked once a solution has been evaluated.

In many implementations of optimizing algorithms, the constraints imposed on the **characteristics of a particular solution** can be taken into account intrinsically in the optimizer's process. This methodology seems to be the most efficient way to incorporate the constraints into the process. Another way to implement these constraints is to evaluate a solution and use the fact that a particular constraint has been violated (or not) to *change the fitness function's value artificially*. A solution that violates a particular constraint will have a far lower fitness value than one that does not. This approach is inefficient compared to an intrinsic implementation, in the sense that it does not prevent the optimizer from trying solutions that are similar to those with violated constraints directly. However, changing the fitness value is *very simple implementation from a programming standpoint* and was the solution used in the original tool. **The search space's boundaries** are set from the start and its implementation is simply allowing the search to be conducted in a particular range of values for the input parameters.

### 5.1.3. Optimizing Algorithm

Optimizing algorithms are the processes that seek the optimal solution for a particular problem. They range in efficiency and computational cost with some having a complex implementation and others being simpler. For this research three optimizing algorithms were investigated.

1. Monte Carlo
2. Differential Evolution
3. Particle Swarm

**Monte Carlo** is the random guessing of a large amount of solutions and extracting the best one.

**Differential Evolution** requires generating a population of solutions combining them amongst themselves and using these child solutions as the next generation of the population. This process is known as a genetic algorithm as it crosses the DNA of each pair of solutions to yield the the following ones. The idea behind the method of differential evolution itself is that the difference between two vectors yields a difference vector which can be used with a scaling factor to traverse the search space [53]. Along the process, the algorithm adds randomness by varying some of the solutions' *genes*. The DNA being a particular solution as a whole, a parameter set, and a gene being an individual parameter of that solution. **This process was coded in the version of the tool before this research took place**.

**Particle Swarm** This algorithm is the one employed by the tool currently, it is used for all the test cases. Particle Swarm Optimization (PSO) creates a *swarm* of solutions and has them following simple laws of motion to determine how the solutions should vary from iteration to iteration. A variation on this, the Fully Informed Particle Swarm Optimization (FIPSO), "informs" all the swarm's members of the current fitness of all of the solutions. The information on the state of other particles is then used to influence their motion when determining the next generation. This causes the swarm to converge on to a solution space where the fitness improves. The selection of this particular variant of Particle Swarm optimization came from reading [54] which states that it outperforms regular Particle Swarm Optimization in every dependent measure. The reason for choosing this method over the others was its performance for this particular problem during testing (see: 6.5.3)

### 5.1.4. Application to this research

It is then essential to structure the application of these processes to the cases under investigation here. The concept here is that the optimizer improves the solutions as a whole, meaning that **trajectory** and **rocket configuration** are optimized simultaneously by the global optimizer. The **fitness function** is then a measure of how the solution performs as a whole and not simply a measure of the characteristics of the rocket itself. This approach is taken for simplicity, other solutions such as having two independent optimization loops, one for the vehicle and one for the trajectory, are also possibilities. However, overhauling the entire optimization process, and studying its implications, is beyond the scope of this work and left as a recommendation for future research. Furthermore, in [6] where a process similar to the approach in this research is used, **a two loop system is shown to only provide a first guess**. A process that optimizes the system as a whole is used in the end to improve upon the initial guesses. This consisted enough motivation not to pursue it any further.

Chapter 6 describes in detail the implementation of these methods and the construction of the fitness function for the MDO problems in this research. There a performance comparison and a in-depth view on the evolution of the solution space are also presented there.

## 5.2. Launch Simulation

The launch simulation is the cornerstone of the software and its outcome that determines the quality of every vehicle configuration. This section provides an overview of the theoretical approach to it as implemented in the new version of the tool. The Launch process is divided in three stages:

1. Assembling of the vehicle - computing the dimensions and properties of the vehicle from the parameter set given by the optimizer
2. Launching of the vehicle - simulating the launch of the vehicle assembled in **Step 1**
3. Assessing final state - returning the final parameters of the solutions to determine its quality

Between each of these steps, the solutions are checked for violations of the constraints on their characteristics (see: 5.1.2). After Step 1 is completed, constraints related to the vehicle's dimensions and $\Delta V$ are checked. If the vehicle does not violate any constraints the program proceeds to Step 2. Step 2 is the actual Launch Process. During Step 2, constraints related to the physical properties of the rocket: accelerations, bending loads, pressure, flow separation, angle of attack and heat flux. If no constraint is violated the launch is simulated up to the end of the burn time of the last stage. Once Step 2 is completed the state of the vehicle is evaluated regarding its orbital parameters to see if the target orbit was achieved. If the goals are met, the information on the rocket is used to compute its value as a solution.

### 5.2.1. Simulation Modules

Step 2 is the bulk of the simulation process. It is in it that the launch simulation models are applied to the rocket. These models are briefly outline here.

#### Acceleration

The rocket's position along the simulation is calculated through integration. The propagation process requires the rocket's state derivative to be known. Determining this derivative is done using acceleration models. Here three models are to be considered:

1. Thrust Acceleration
2. Aerodynamic Acceleration
3. Gravitational Acceleration

**Thrust acceleration** is calculated as part of the propulsion models associated with each launch vehicle. The implementation of the propulsion model, that generates the thrust information, is described in Chapter 7 but in essence these models return a thrust curve associated with each rocket stage. **Aerodynamic Acceleration** is about the determination of drag as caused by the interaction of the vehicle with the atmosphere. The details of this model are beyond the scope of this research as such the implementation is simply the aerodynamic models that accompany Tudat (The TU Delft's astrodynamics software library). Briefly, these models require geometrical data from the rocket along with a particular Atmosphere Model. The models then use Missile DATCOM (See: 6.3.3) to compute the resulting Aerodynamic Acceleration. Finally, the **Gravitational Acceleration** is computed using also a model from Tudat that computes the effect of the Earth's gravity on the rocket's motion. This model is set to account for the J2 effect.

**Computing the total acceleration** to compute the total acceleration all the individual accelerations are converted to their representation in the same referential and thus the position's state derivative becomes known.

#### Pitch and Control

Knowledge of the vehicle's orientation is required to compute the accelerations, particularly the Thrust one. The orientation of the rocket depends on the intended trajectory which also part of the optimization process. Briefly, the program receives, along with the parameters that make-up the vehicle design, a list of pitch angles. These pitch instructions, included in each solution, are used by the **Pitch module** which computes the history of the Pitch angle value for each instant. With the rocket's orientation known, the **Control module** is then responsible for determining the direction the thrust is pointing. This is the module implements the Thrust Vector Control (TVC) into the rocket guidance if available.

#### Environment

The environment is still heavily based on the original tool. It is a model that considers and updates all the characteristics external to the vehicle itself. It considers an Atmosphere model based on the 1976 Atmosphere Model Tables by NASA as implemented in Tudat [55].

#### Integration and Propagation

Even thought it is not an actual module of the simulation, but rather the process that brings the simulation to life, the integration process is just as crucial as any of the previous models. The integration of the state is done based on all the previously discussed elements. A runge-kutta integration is to be used. Runge-Kutta is a 4-step integration method and thus it is crucial to ensure that the values for all the elements external to the rocket are updated not only before and after but also during the intermediate steps. This is done through the integration with the Tudat tools and reflects an improvement to the previous version of the tool which lacked this. This is further addressed in Chapter 6 where the upgrades to the tool are discussed.

# 6
# The Software: MultiStageLaunchVehicleSimulator

The back-end of any software package is its core, its engine. Getting it to work correctly is of prime importance and without first addressing this, the results of any research that will derive from it cannot be deemed satisfactory. Even though this part of the work might seem less important when compared to the research itself, that is not the case at all. The thorough work required to get the software to work as intended was invaluable. The main reason for the tool to be upgraded is the fact that the architecture it is based on, TUDelft's Tudat, was heavily updated since the tool was first started. The Tudat update combined with the particularities of the tool made it very difficult for anyone new to it to get it to *simply run*. By achieving a robust software base we enable future users to focus on the aerospace engineering research side and not so much on the computer science side. As such, there were two goals that were set for this part of the work: improve usability of the tool, and bring it up to the current standard. The details of this upgrade, the changes in the way the tool now uses and interacts with Tudat are discussed in detail in this chapter. A great deal of importance is given to the improvements that have the larger implications to the results that the software yields for each case. Comparative tests and re-validation of the post-upgrade version of the tool is also covered here through analysis of real cases and unit-tests. It is also important to notice that the tool had to be upgraded in its core, as opposed to just adding or removing certain modules. The upgrading process implied that the way the tool works, particularly its physics, had to be reviewed. This review **pointed out some flaws with the computation that were previously not noticed in the application**. This chapter describes the upgrades that were done, the corrections on the flaws that were found but also gives an overview of the architecture, functionalities and possibilities of the new version of the software. A summary of these changes can be found in Table 6.2 later in this chapter. A crucial thing to understand about the work described in this chapter is the nature of computer programs. The previous version of the tool had *everything hard-coded into it*. I consider two levels of hard-coding: hard-coding of values and hard-coding of functionalities. If previously the values were hard-coded but no significant re-writing was required to evaluate a different case or task then I consider that that particular functionality was already available in the previous version. In case a particular task requires heavy re-writing: changing multiple files in the hierarchy, possibly unlinking and re-linking certain functions, then I **consider that the functionality was not available**. The reason for this distinction is that while the first task is easy and straightforward, the cases in which the second task was required (i.e. a use of the tool that required heavy re-writing) involve a considerable amount of work and understanding of the software. These cases required some work to make them work all the time and for every case. As such, even though the work [4] may already have used the tool for some of the ends mentioned in this chapter, some are still as described "not possible in the previous version". The writing of this chapter counted with the participation of Menno van Kesteren from whom I got excellent feedback and had productive discussions. These personal communications are used throughout the text and were a significant source of material and ideas.

## 6.1. What can the Software Do?

Since the development of this software started with previous students it is important to distinguish **upgrades** and **new features**. The **upgrades** concern the resolution of the issues of the previous version (see: 6.4.1). The **new features** concern not only the addition of Hybrid Rocket Stage simulation capabilities, but also new applications[1] (executable programs for standalone tasks) and new data-output capabilities. Even though some of what is described here was completed by the original authors of the software, most of the code is new. The reason for the existence of so much new code is that even the functionalities that were kept had to be rewritten to some extent. Throughout this chapter the word trajectory is used in the sense of a set of *intended* trajectory instructions - in this case pitch angle instructions. Knowing the set of instructions does not necess Currently the software can perform 3 main separate tasks:

1. Find an Optimized Solution for the Vehicle and its Trajectory.
2. Find an Optimized Solution for the Trajectory of a known vehicle (not possible before the upgrade without some considerable changes to the files).
3. Simulate the launch of a known vehicle with a known trajectory.

Additionally the program now contains a companion application that renders the combustion of a stage in detail printing out the performance curves (Thrust, $I_{sp}$, O/F and others) to a file. The integration of this application is shown in Figure 6.1 with some results of this application shown in Figure 8.3. Currently, this functionality only supports Hybrid Rocket Stages. However, the options for the functionalities in the list have been greatly expanded. Both the launch simulator and the optimizer side of the tool now support **3 kinds of stages**: Solid, Hybrid and Custom. Solid Stages are entirely based on the code from [4], whereas the Hybrid Models come from the literature study that preceded this thesis. The description of the theoretical models for the Hybrid Stages is found in Chapter 4 and the detailed implementation in Chapter 7. The custom rocket stage can take the form of a hybrid or a solid rocket stage and are simply **fully customizable stages** for which nearly all parameters can be manually specified including the input of **Thrust and Mass Flow** curves in the form of data points in a text file. The addition of this functionality comes from the fact that the models that represent both the Solid and Hybrid stages are as close as possible to reality. However, these are also meant to be generic and represent a large swath of rocket options. This fact made that the accurate simulation of demonstrated technology with available data, such as the NAMMO rocket and the DARE Stratos II rockets, was not possible in a consistent way. With the manual specification of the parameters, more accurate results can now be produced. It is also worth noting that the capabilities to simulate Air-launches were retained and made available for all stages types, as is the ability to **simulate rocket vehicles that make use of both hybrid and solid stages** or even including custom stages in the combination. Another very significant addition, especially for the collaboration with DARE is the actual **possibility of simulating Single Stage Rockets**, as well as analysing the motion of Rockets that do not necessarily reach orbit. The work of van Kesteren [4] did include a test with a single-stage rocket using the original tool by Vandamme [3], however later additions and modifications made this no longer possible since this was not of particular interest for the work in [4]. This functionality was restored and streamlined.

## 6.2. Architecture

This section describes the architecture of the tool on three levels. The first level, is a systems view where the overall applications are described. This view shows the relations between the different modules of the tool. The second level describes the relation between the two main systems: the optimizer and the simulator. An overview of these relationships is illustrated in Figure 6.1. In the end of the section, the architecture of the vehicle and of the stage modules is explained.

### 6.2.1. Applications and Processes

To keep the results consistent all of the code is shared as much as possible. In the end, there is **only one launch simulator** even though it is used by multiple applications. This simulator is used by the

[1] "Applications" is used throughout this chapter in the context of a computer program, routine or executable

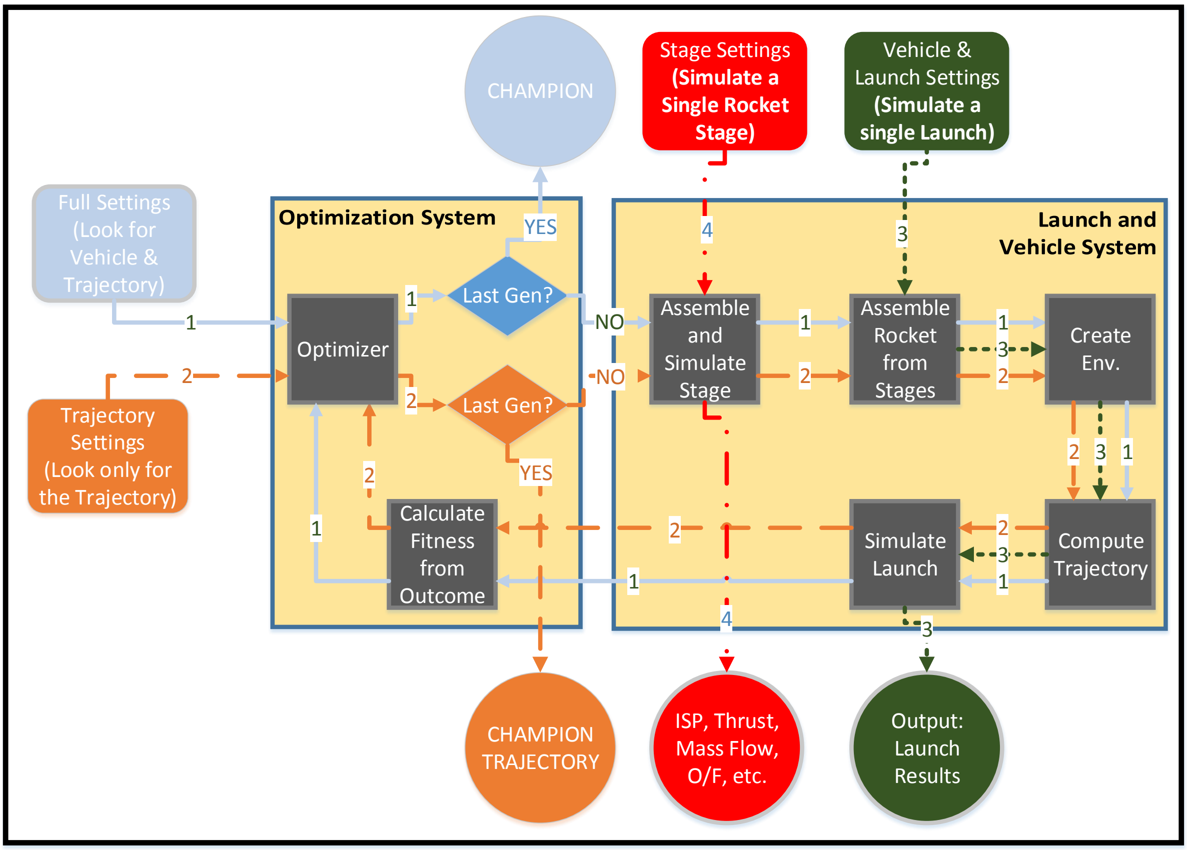

CHAMPION
Stage Settings (Simulate a Single Rocket Stage)
Vehicle & Launch Settings (Simulate a single Launch)
Full Settings (Look for Vehicle & Trajectory)
Optimization System
Launch and Vehicle System
YES
Optimizer
Last Gen?
Last Gen?
NO
NO
Assemble and Simulate Stage
Assemble Rocket from Stages
Create Env.
Trajectory Settings (Look only for the Trajectory)
YES
Calculate Fitness from Outcome
Simulate Launch
Compute Trajectory
1
2
3
4
CHAMPION TRAJECTORY
ISP, Thrust, Mass Flow, O/F, etc.
Output: Launch Results

Figure 6.1: Current architecture of the tool with all of its main components. **Each number designates a different application**, meaning they can be triggered separately.

applications for the different ends described in 6.1. The new underlying architecture is illustrated in Figure 6.1. A total of 4 applications are now part of the tool:

1. LaunchVehicleOptimization: Computes a Solution for **Vehicle** and **Trajectory** that is a global optimum. This solution reaches a pre-determined target defined by requirements in eccentricity and semi-major axis. Target is considered reached once the solution is in within a pre-set range of it. The optimum is a minimization of Cost or GTOW[2]. This application Uses the **PaGMO** library.

2. TrajectoryOptimization: Computes a Solution only for the **Trajectory** that is a global optimum, i.e. the same as application 1 but with a fixed rocket. This solution reaches a pre-determined target (defined by requirements in orbital parameters - eccentricity and semi-major axis). The optimum is a minimization of final mass (i.e. remaining propellant available for future manoeuvres). Uses the **PaGMO** library.

3. MultiStageLaunchVehicleSimulator: Simulates a launch for a known rocket and trajectory. Its primary use is to have a discrete tool to print and save the orbital data of the launch itself for further analysis.

4. StageTester: takes a stage configuration and allows for manual setting of its settings to observe and study its characteristics like Thrust, $I_{sp}$ and other parameters in the form of a parameter history, the same one used by the simulator to calculated the accelerations. **It only supports Hybrid Rocket Stages currently**. An example of a result of this application can be found in Chapter 9, in Figure 8.3.

[2] The optimization of solid rockets is available in both GTOW and Cost, however, for hybrids only GTOW optimization is available for reasons that are discussed in Chapter 4

In Figure 6.1 the arrows described the flow of the possible applications. **Before the upgrade only Application 1 and 3 were possible**, and even these were upgraded during this work. Currently all 4 applications are possible. Meaning that a significant amount of functionality has been added to the tool. Process 1 is the optimization of the full problem: vehicle and trajectory. Process 3 was the simulation of a single launch for a known vehicle and trajectory. The two new processes are:

1. Process 4: Simulation of a Single Rocket Stage.

2. Process 2: Optimization of the Trajectory for a known vehicle configuration

Process 4 is one of the most relevant additions to this project due to the nature of Hybrid Rockets. Process 4 outputs the predicted history of all the relevant parameters of a stage. Meaning that it allows for detailed analysis of the behaviour of the following variables:

1. Thrust

2. Mass Flow

3. O/F Ratio

4. Burning Surface

These parameters allow for a user to separately study and configure the stage model before launch simulations are ran. It is also important for the Verification and Validation studies on Hybrid Modelling. Process 2 is, however, the most important addition. Due to a limitation in the previous version of the code, it was not possible to vary the number of variables to be optimized, at least without considerable changes to the PaGMO files. Some flexibility was achieved here. Primarily it is now truly possible to optimize a single-stage rocket. Previously the lower limit was 2 stages. However Application 2 optimizes no stages. **Instead it only optimizes the trajectory keeping the rocket configuration constant throughout the entire simulation.** This opens up a new range of possibilities for testing known cases. Particularly for the cases of Nammo and DARE where the rocket configuration is already known. The tool can now attempt to maximize a particular rocket's reach through trajectory optimization. This application yields important information regarding the limits of a particular rocket technology and/or configuration.

#### Associated MATLAB scripts

As a companion to the tool some MATLAB scripts were developed to facilitated plotting and analysis of the data resulting from the simulations. The scripts are a streamlined way to process data and to transform it into the plots presented in this work. The scripts include the ability to map to scale the 3D path on the planet of the orbital injection of the rocket's payload.

### 6.2.2. Relation between Optimizer and Simulator

The new additions benefited from the original architecture of the tool where there is a separation of Optimizer and Simulator. The main components: the optimizer and the launch simulator are quite distinct from each other. The optimizer is named PaGMO (Parallel Global Multiobjective Optimizer [56]), which is devloped by the European Space Agency (ESA). PaGMO only requires that the problem it is working on, in this case a launch simulation, computes and returns the value of the fitness function. Although PaGMO supports direct implementation of constraints to its solution space this is not strictly required and not used in this project, an alternate solution for dealing with the constraints was used, this is addressed in 6.5.1. Once the optimizer knows the fitness value of a particular population of candidates it can produce the next generation of that population. The **Launch simulator** enjoys a similar relationship of independence with the **Rocket Launch Vehicle** and **The Environment Model**. The simulator requires only that a vehicle and an environment model be specified. Once the connection between the simulator and the modules is established, the simulator is able to get the vehicle's current State Vector, integrate it and relay the result back to the models. The simulator's main core and the physical modules are therefore *de-coupled*. The more interesting thing here is the modularity has now been achieved in this thesis for the Rocket Launch Vehicle itself.

### 6.2.3. Rocket Vehicle and Stage Architecture

Hybrid Rocket Modelling is the latest addition to the tool and the one that is central to this thesis project. Adding the hybrid modelling was done at the same time as the Rocket Launch Vehicles simulated by the tool were made more flexible. In particular, the rockets are now generic objects that contain **an arbitrary number of stages of any type**. These are developed with the use of *abstract classes* in C++. Using abstract classes means that the basic requirements for building a Rocket Stage are fixed (e.g. it must provide mass and thrust information) but the way these are calculated is of no concern to the simulator. Any user can, in theory, later write different kinds of stages and have the simulator use it for the rocket launch vehicle under study. This change in tool architecture also **made it possible to build rockets whose stages are of different kinds**. This possibility is particularly interesting as there is some interest in building rockets whose last stage is Hybrid, but the first two are Solid, for instance. In Table 6.1 the connection between the different kinds of stages is shown.

| Abstract Object | Normal Object | Specific Methods | Common Methods |
|---|---|---|---|
| RocketStage | SolidRocketStage | SolidStagePropulsion()<br>SolidStageGeometryAndMass() | getLengthStage()<br>getMassFlow()<br>getVacuumThrust()<br>getBurnTime()<br>linearInterpolator()<br>etc ... |
| | HybridRocketStage | HybridStagePropulsion()<br>HybridStageGeometryAndMass() | |
| | CustomRocketStage | getThrustHistory()<br>getMassFlowHistory() | |

Table 6.1: Organization of the Rocket Stage objects in the code

## 6.3. Libraries and other tools

The tool's architecture works by using the 2 main libraries: Boost and Tudat. The first deals with technical programming aspects and the latter with the physical models related to the simulations. When using Tudat and Boost, one must understand also CMake, a tool to facilitate the compilation for projects that make use of all of these tools. Understanding and studying these aspects was important to upgrade and re-do the tool, and thus are summarized in this section for clarification.

### 6.3.1. Tudat

**Tudat** is a software library developed by the TU Delft to aid in the development of Space Missions Analysis software. As per its documentation:

> The TU Delft Astrodynamics Toolbox (Tudat) is a set of C++ software libraries, developed and maintained by staff and students in the Astrodynamics & Space Missions research group at the Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands. This toolbox is intended to provide users with the functionality to be able to simulate various astrodynamics applications. [55]

This library is a set of files written in C++ that cover a wide range of tools including Mathematical tools, Atmospheric and Gravitational models and others. Tudat has been in constant evolution since its original conception perfecting it and making it more and more useful with each new update.

### 6.3.2. Boost and CMake

In order to use Tudat and its tools there was some programming background that I lacked that required studying, particularly **CMake** and **Boost**. **CMake** is used to compile and manage the project. It runs the commands that organize which files are compiled, with which settings. The **Boost** library is a library with vast applications, but here it is mostly used to create smart pointers. Smart pointers allow for efficient memory management which improves the usability of the tool and prevents some programming pitfalls.

Boost greatly is a tool that simplifies a lot of the code-writing but it required learning its syntax to some extent that consumed a very considerable amount of time. **It is my recommendation that every student, that attempts to tackle a similar task in the future, follows the Tudat exercise MSc Course from the TU Delft**. A lot more time is sometimes devoted to set-up and initial testing than it is estimated by students and this is a valuable lesson that I want future students to take from this work.

### 6.3.3. External Tools and Software

There are a few external tools that are essential for this tool and research. These involve both those that were already used, and motivated, in [4] and [3] and a new addition. The tools are Tudat, TransCost, Missile DATCOM and CEA, RPA is the new addition. Since the other tools have already been given motivation for their use, they will not be discussed in this section but their functions are described briefly in the following list.

1. Tudat - Astrodynamics Toolbox from TUDelft
2. TransCost - Vehicle Cost Estimation Model
3. Missile DATCOM - Library Used to Estimate Aerodynamic Coefficients
4. CEA - Thermodynamic Data Reference, mainly used for combustion calculations

NASA's CEA is the primary source of thermodynamic properties to be used in the course of this project. It is an industry standard and its data is considered reliable being used in [4, 41]. RPA is a more modern and newer tool that bases its information on NASA's CEA thus keeping the integrity and reliability of the data [50]. The usage of RPA was tested (see:4.3.1, with the raw data included in the Appendix to this report.

## 6.4. Upgrades to the Software

The software upon which this thesis' research is based is the combined result of three previous master thesis at the Delft University of Technology. The students Frank Engelen [11], Jan Vandamme [3] and Menno van Kesteren [4] made the package that was the starting point for this work. Their work started more than three years ago - time during which the Tudat architecture evolved significantly. This cause significant difficulties in the early stages of this work regarding the set-up of the software and its usability. This section describes the changes made to the tool and the technical functionalities that were added to it. It is divided in two parts. The first part covers the bugs and issues that had to be resolved. The second part deals with the functionalities that were added.

### 6.4.1. Addressing the software's issues

The immediate technical problems with the software were the following:

1. **Issue 1**: The software did not run **"out-of-the-box"**.
2. **Issue 2:** The software requires the Tudat library to work (as expected) but **did not support the version that is currently made available by the university**.
3. **Issue 3:** The coding style was unclear and poorly documented, package included a lot of obsolete useless files [3].
4. **Issue 4:** Once the resolution of items 1 through 3 was achieved, a number errors in the simulation's code itself were found and corrected as well. These are discussed in this section.

The following subsections analyse the resolution of these issues.

[3] The folder that was passed on was a direct "dump" of the complied tool along with all the Tudat files. Even though this is the quickest, and in truth the most common, way of passing on software it was a major cause of the excessive time that the software re-working took. For future students a different practice should be adopted

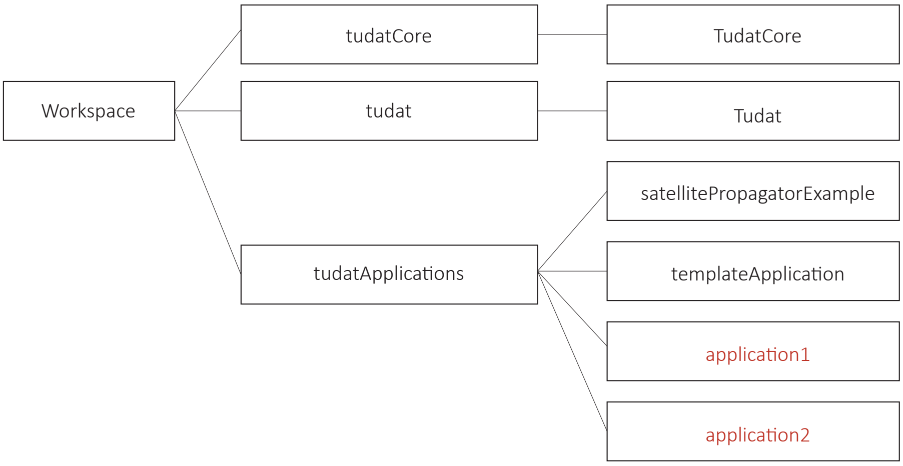


Figure 6.2: Tudat Application Structure. [55]

### Issue 1: The software did not run ”out-of-the-box”

**Explanation** The software that was passed down consisted of a set of different files that made it confusing to deal with. Any software in development has these three sets of files:

1. Source Files - The Source Code that can be compiled to make the program (including library files)
2. Build Files - Files generated by the compiler
3. Applications - The programs themselves

The structure of these files was not provided to me which meant that studying it became one of the main tasks here. In an ideal situation the program would work ”out-of-the-box” meaning that it should produce some sort of output when run by a user. This was **not** the case. When attempting to run the program on my system, I found a **number of warnings of missing files and libraries**. In essence, the build files and the applications seemed useless as they were compiled for the developer’s machine and system. The disorganization surrounding the PaGMO (Optimization) library and the Tudat files also contributed to a lot of difficulties in the beginning. Because of this the tool, as it was, could never work out of the box and why its update was a crucial step for this project.

**Resolution** Getting the program to run was the priority here. The decision was made to move all the files to a Linux system, study its structure, re-organize and get it up and running. As explained in Section 6.3.1, Tudat is the main software library on which the application works. All Tudat tools rely on a consistent structure, illustrated in Figure 6.2. In Figure 6.2, the developed application is located at the lowest level of the hierarchy along with ”templateApplication” and ”satellitePropagatorExample”. All of these applications require the **Tudat** and **TudatCore** components of the library to run. The library is provided by the university as source code files. These source files need to be compiled in the machine they are to be run on. Once this process completed, it is possible to compile the application itself, and it will work as intended. The tool as received had all of these already compiled in a Windows system. It used a very old version of Tudat (meaning the components Tudat and TudatCore were out-of-date and unsupported). Moreover, it was not organized or clean enough to be re-compiled and used on other machines. The first few weeks of this project required extensive study on the tool and how it worked and, with the contribution of Dominic Dirkx, it was cleaned and restructured in a way where it **could now be easily recompiled on any machine**.

### Issue 2: Tudat version update

**Explanation** As explained in the beginning of this section, the version of Tudat on which the tool was built upon was out of date and no longer distributed by the university. This brought two distinct problems. First, by using outdated code requesting assistance or troubleshooting issues becomes much harder as the people that maintain Tudat are no longer familiar with the code. Second, expanding on the tool would be far more difficult as the available materials and documentation available all pertain to the new version of Tudat. Adding to this come the benefits brought about by the upgrades that the Tudat architecture underwent. For these reasons, an update of the tool to be compliant with the new version of the library was clearly necessary.

**Resolution** In order to make the tool compatible to the new version of Tudat many references to it in the code had to be re-written to be in compliance with the way in which Tudat currently works. Not only the references to the library components but also the core mechanic of how the tool should function, particularly regarding the integration and the state derivative models, had to be modified. In this particular part of the work the contribution of Dominic Dirkx was indispensable. This resulted in implementing one of the best new characteristics of the tool: the new *CMakeLists.txt* file. The file, designed to load to and compile programs, makes it easy to develop the tool further and now simplifies the compilation process. Now, one only has to load a clean version of the tool (no build files and no applications) and place it side-by-side with the library components. The required file organization is illustrated in Figure 6.2. With that structure, simply using the new *CMakeLists.txt* full compilation is achieved.

### Issue 3: Coding Style, Documentation and Files

One of the main problems with the tool was fact that it had multiple authors making its code look somewhat disaggregated. This issue made addressing some of the underlying problems with the tool much harder. For example, a particular memory leak caused by the entanglement of the multiple modules in the tool took weeks to find and solve. This leak caused by a recursive addressing of some of the C++ objects to other C++ objects made the program unable to run without exhausting the computer's memory. The reason why it occurred pertains to the integration of the new Tudat version and the update involving the propagator (also described in this section). The problem was caused by a dependency between the Rocket Vehicle module and the Environment module. This dependency was not totally understood in the beginning and required an in-depth study and **rewriting most of the code that governed the main simulation loop**. Though it consumed a lot of work-hours its benefits: accuracy improvements and true modularization of the software, were worth the effort. In fact, solving this one particular problem resulted in performing 3 essential steps for the improvement of the tool:

1. The simulator was re-written, it now consists of a shorter code, simpler steps and clear purposed functions

2. Comments were added to all the steps, including assumptions and dependencies on other modules

3. CMake project compiler was re-made.

The CMake component was **overhauled to compile both the libraries and the code for the tool itself automatically**. This change reduced confusion and simplified the installation and compilation processes. Understanding how CMake works was important both for this upgrade and to identify a very large amount of **useless or obsolete files that were cluttering the tool**. Only the bare minimum number of files is included now. This means that adding or removing features from the tool is much simpler. The folder structure was also simplified to allow the identifications of which files control which part of the program. An example of this is the folders that have the Optimizer code. This is a package from ESA that should be easily replaceable or modified as desired. Instead of being called PaGMO (the library's name) it was inside a number of sub-folders named ”src”. This re-organization clarified all of this. And the example is merely one of many such practices that really made beginning the work quite difficult.

### Issue 4: Simulation errors and bugs

**Explanation** There were three functional errors found within the code that were promptly addressed:

1. Propagator Error

2. Atmospheric Model Error

3. Vehicle Length Calculation Error

The propagation error was an integration error, not a problem with the integrator itself, but with the environment variables that it needs to update the trajectory. Runge-Kutta integrators, like the one used by the tool, rely on a 4-stage integration process. However, prior to the update, the integrator would *not* update the state of the rocket (position and velocity) on each of the 4 steps, but would only update the environment variables on the **first and fourth steps**. This meant that it was **propagating erroneous information** regarding the atmospheric properties, the mass of the rocket and *every other dependent variable* while integrating. Upon the resolution of the propagation error, the Atmospheric Model started yielding unrealistic values for density and other air properties. It is not clear why this only showed after the Tudat update and the integration error correction, but it also became an issue to be addressed. In addition, an error in the calculation of the total length of the vehicle was found. The total length was being over-estimated by adding the nozzle components twice for each stage that had an interstage modelled. This was only the case for 3 and 4 stage launch vehicles where the inter-stages were present (see [4] for why this was the case).

**Resolution** The resolution of the **propagator's problem** was mostly a restructuring of the way the code works In essence, the update of the member variables was separated from the main simulation into a class of its own and could now be called (and updated) at will from any point in the code. This change enables the update method ( *updateMembers()* ) to be called *every integration step*, thus correcting the problem. The application of this fix along with the Tudat version update finally showed the atmospheric model error. The **problem with the Atmospheric model** was solved by simply replacing it with the one included in the new version of Tudat. It now uses the 1976 U.S. Standard Atmosphere, with points taken every 100 m up to 100 km and points taken every 1000 m from 100 km to 1000 km. Finally the **Vehicle Length Error** was corrected by adding the length of the interstages the correct amount of times. This correction was also done because the Rocket Vehicle Model was re-written as part of the new features, in particular, the support for n-stage rockets that currently exists.

**Comparative studies** A version with the old atmospheric model never worked with the tool that I received and for this reason a comparative study of that error was never done. The integration error though, was studied this is examined in 8.2.2.

### 6.4.2. New Features and Properties

A considerable part of the development time was spent in making the tool more versatile. The need for the improvement came, in part, due to a collaboration with NAMMO and DARE, who had specific results they needed in mind. The idea was using the tool to reproduce their cases and to simulate a hypothetical launch of their vehicles. In DARE's particular case, we are dealing with a Single Stage Rocket. The previous version of the tool did not allow for this to be simulated, the current one **supports any number of stages**. Also, to simulate any particular rocket, the most accurate way is to **provide Thrust and Mass Flow curves** along with generic data regarding mass and dimensions. The possibility of supplying these values and bypassing the estimator modules was also added. Another improvement was the addition of the trajectory optimization module. Before the update only a complete problem could be simulated. In the current version, one can provide the rocket configuration and **ask only for an optimization of the trajectory**. This is the result of a major change in the optimizer integration: the number of variables to be optimized is now no longer fixed at 28 but a variable that depends on the number of stages of the rocket (for the full problem) or a fixed 11 for the trajectory optimization. The reason why the trajectory optimization is fixed at 11 points (plus the starting and end point) was because this was the strategy employed by van Kesteren in [4]. Table 6.2 summarizes the changes between versions of the tool. In addition the Hybrid Rocket Module is quite configurable, supporting all the options described in Chapter 7. Most of all, the program is now truly modular, especially in Rocket Technology, adding a Liquid Stage module would be only a matter of time and effort. The creation of a file with the code, its addition to the project and creation of the option in the configuration files is all that would be required. However, improvements other than the modelling capabilities were added, especially regarding usability. Since the tool has to be used for many cases, and each case has its own settings and particularities, some improvements were made on the user experience of the tool. These are described in the following sections.

| Feature | Before | After |
|---|---|---|
| Number of Stages | 2-4 | Any |
| Number of Applications | 2 | 4 |
| Custom Thrust and Mass Flow Curves | No | Yes |
| Separate Trajectory Optimization | No | Yes |
| Rocket Technologies | Solid | Hybrid, Solid & Custom |
| Allows for Rockets with multiple types of Stages | No | Yes |
| Choice of Physical Constants | Requires Recompilation | Available in Conf. Files |
| Choice of Payload Mass | Requires Recompilation | Available in Conf. Files |
| Choice of Launch Settings | Requires Recompilation | Available in Conf. Files |
| Choice of Tolerances, Quality, Design and Safety Factors | No centralized way of doing so | Available in Conf. Files |
| Choice of Problem Settings (# Generations, Pop. Size) | No | Available through commands |
| Progress indication (Generation #) along the computation | No | Yes |
| Monte Carlo Method Available | No | Yes |
| Centralized Generation of Output Files for all variables | No | Yes |

Table 6.2: Summary of the new features added to the tool as part of the upgrade process.

### Usability Improvements

As it stood, the average user could not use the tool because it required constant re-compiling in order to experiment with different cases and settings. This is due to the fact that all the numerical values - physical constants, vehicle settings, configuration parameters, etc. - were hard-coded in the source files. In order to speed up testing and to allow for reproduction of the results, or to make the tool available to other users, a simple "command-line + configuration files" interface was developed. The appendix presents a more detailed explanation on how they work, but the structure is as follows:

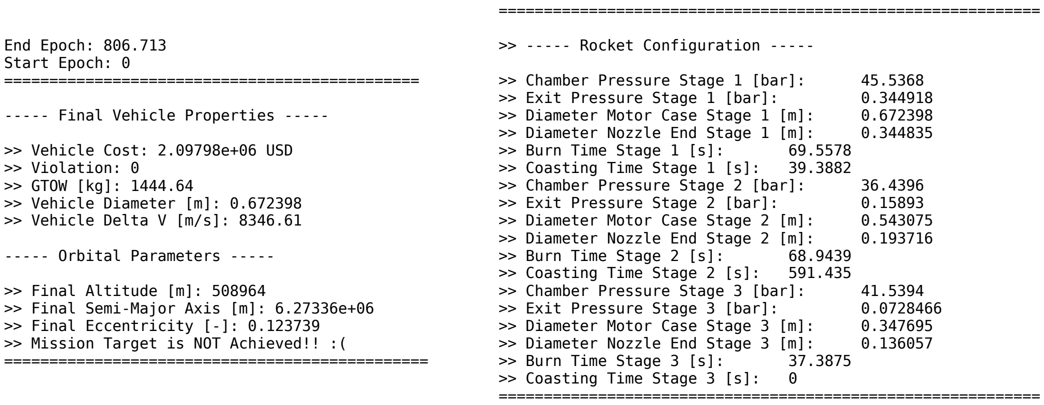
1. LaunchVehicleOptimizer
   (a) arguments: -help, -stc, -ptc, -xv, -numGen, -popSize
   (b) configuration file: *vehicleOptimizationSettings.txt*
2. TrajectoryOptimizer
   (a) arguments: -help, -stc, -ptc, -xv, -numGen, -popSize
   (b) configuration file: *trajectoryOptimizationSettings.txt*
3. StandAloneLaunchSimulator
   (a) arguments: -stc, -ptc, -vtc, -validate
   (b) configuration file: *vehicleSettingsForLaunchSimulator.txt*
4. StageTester
   (a) configuration file: *stageForTesting.txt*
5. Common Settings Files
   (a) *PropellantData.txt* (1 per stage)
   (b) *ConfigurationSettingsStage.txt* (1 per stage)
   (c) *RocketLaunchVehicleConfiguration.txt*
   (d) *inFlightConstraintData.txt*

Along with the input scheme, the output has now also been streamlined. The output is controlled by the program arguments detailed above, and it is written to file in the applications' folders. The primary output of *launchVehicleOptimizer* is optimization data: fitness history, parameter sets for population individuals and final orbital data for each "attempted flight". For *standAloneLaunchSimulator*, all the physical values of most of the important parameters are made available at the same resolution at which they are computed inside the code (i.e. one value per time-step). The appendix presents a complete list of all possible values. In addition to the output files, some information is printed to the console at both the end and beginning of the program's run. The printout includes generic information like if the proposed vehicle has reached the mission's target or a detailed overview of the rocket's design parameters. Figure 6.3 presents an example of the rocket configuration and Launch results.

## 6.5. Fitness, Optimization & PaGMO

The other main component of the software is the optimization. As described in [4], the PaGMO optimization library by ESA is used. I will not expand here on the merits of the system itself since that can be found in [4]. PaGMO is compiled along-side the rest of the tool and causes the simulator part of the code to evaluate multiple launches measuring their fitness and using it to improve on each set of design parameters (see 5.1.3 ). A detailed investigation into this component of the software (and of the research process itself) is not the main focus of this work. However, in the end, a small investigation had to be done for two main reasons:

1. After the upgrade (see: 6.4.1) the optimizer **could no longer reach a feasible solution** - at least using the parameters with which it was configured and also not through any simple re-configuration of them



```
End Epoch: 806.713
Start Epoch: 0
==============================================

----- Final Vehicle Properties -----

>> Vehicle Cost: 2.09798e+06 USD
>> Violation: 0
>> GTOW [kg]: 1444.64
>> Vehicle Diameter [m]: 0.672398
>> Vehicle Delta V [m/s]: 8346.61

----- Orbital Parameters -----

>> Final Altitude [m]: 508964
>> Final Semi-Major Axis [m]: 6.27336e+06
>> Final Eccentricity [-]: 0.123739
>> Mission Target is NOT Achieved!! :(
===============================================
```

```
============================================================

>> ----- Rocket Configuration -----

>> Chamber Pressure Stage 1 [bar]:      45.5368
>> Exit Pressure Stage 1 [bar]:         0.344918
>> Diameter Motor Case Stage 1 [m]:     0.672398
>> Diameter Nozzle End Stage 1 [m]:     0.344835
>> Burn Time Stage 1 [s]:       69.5578
>> Coasting Time Stage 1 [s]:   39.3882
>> Chamber Pressure Stage 2 [bar]:      36.4396
>> Exit Pressure Stage 2 [bar]:         0.15893
>> Diameter Motor Case Stage 2 [m]:     0.543075
>> Diameter Nozzle End Stage 2 [m]:     0.193716
>> Burn Time Stage 2 [s]:       68.9439
>> Coasting Time Stage 2 [s]:   591.435
>> Chamber Pressure Stage 3 [bar]:      41.5394
>> Exit Pressure Stage 3 [bar]:         0.0728466
>> Diameter Motor Case Stage 3 [m]:     0.347695
>> Diameter Nozzle End Stage 3 [m]:     0.136057
>> Burn Time Stage 3 [s]:       37.3875
>> Coasting Time Stage 3 [s]:   0
============================================================
```

Figure 6.3: Representation of the new output formats that allow the user to inspect the launch simulated by the inputted solution.

2. The performance described in [4] and the fitness determination are, to a degree, easily upgradeable and, as will be shown in this section, are worth upgrading and looking into

In the end the changes to the optimization process proved useful, improved the tool's performance and have a solid motivation behind its design choices.

### 6.5.1. Fitness

The optimization performance described in [4] seemed dismal and impractical since it required *10000 generations* with a population of 70 individuals reaching only *67 vehicles that met the target*. This process took, at least with the re-written tool, more than a full day to run. However, the goal here was to design or rework the process such that it became something consistent with a well-defined motivation and this was achieved with the current tool. The performance improvements of the new fitness system are discussed in 6.5.3.

#### Investigation and improvement of the fitness calculation

In the previous version of the tool there was no clear motivation for the values behind the fitness attributed in some situations. Essentially there was brute-force use of enormous values to stress the quality difference between different levels of solutions (often larger than the allowed length of the software variables which made it quite dangerous programming-wise). In this research the fitness calculation scheme was altered to have a more consistent and more programming-friendly design. The main issue pertains to the fact that the fitness cannot be calculated as a single continuous along its entire range. Simply put, it makes no sense to use the orbital parameters of a particular body as a measure of solution quality if the vehicle cannot even reach any orbit. Similarly, it is also meaningless to measure fitness via the GTOW of a particular vehicle if it does not reach the right orbit (even if it reaches another one). The fitness calculation is therefore not a continuous function in this research, instead it is a piece-wise function. Considering all of these characteristics, the new fitness system was designed with the following requirements:

1. The fitness calculation rules shall reward proximity to target before optimizing for cost or GTOW, since reaching the target is a priority in the search process.

2. Given the piecewise nature of the fitness calculation, the value of each level of fitness **shall never be higher than the one that follows it**. This is the reason the current system follows the structure in Equation 6.1.

$$f = f_{level\ max} \cdot q \tag{6.1}$$

Where $f$ is the fitness value of the current solution, $f_{level}$ is the maximum fitness value considered for a particular kind of solution and $q$ is any fraction ($< 1$) that measures the proximity to the target. This applies to all levels except the last one where cost/GTOW is optimized.

### The Fitness Function

The **fitness** evaluates how good a particular set of design parameters is. The evaluation measures *how well each parameter set meets the target*. In the case of this tool the target is to reach a particular orbit. An orbit that is defined by its semi-major axis and eccentricity. A margin of error is obviously acceptable when determining whether or not the orbit was achieved. For instance, [4] uses a margin of 20 km for the semi-major axis and one of 0.01 for the eccentricity.

**Determining the fitness** Calculating the value of the fitness function implies a classification of the current parameter set according to how well the target is met. This classification is not a continuous function it is only piecewise-continuous. This is so because there are four fundamental ways to classify:

1. The vehicle reaches the mission target successfully.
2. The vehicle is feasible and does not violate any constraints during launch but does not reach the target before using all the fuel.
3. The vehicle is feasible but violates a mechanical or trajectory constraint during launch.
4. The vehicle is infeasible,i.e. particular dimensions are incompatible with realistic designs.

Since PaGMO optimizers work by attempting to *minimize* the fitness function (see: 5.1.1) it is essential that *better solutions* get a *lower* (more negative) fitness value. As explained in the beginning of the section, the calculation of the fitness value is done using a piece-wise function, each piece of the function reflects one of the levels in the aforementioned classifications (1 through 4). These levels and their respective fitness functions are summarized in the end of this section in Table 6.3. Thus, and this is a considerable change from the process used in [4], the fitness value is calculated as per Equations 6.2 through 6.8.

**Vehicle is infeasible:**

$$f = +1 \cdot 10^6 \tag{6.2}$$

**The vehicle is feasible but violates a mechanical or trajectory constraint during launch:**

$$f = -1000 \cdot h_{final}/h_{ref} \tag{6.3}$$

In Equation 6.3, $h_{final}$ is the altitude at the launch's final epoch and $h_{ref}$ is the reference altitude. The reference altitude is defined in Equation 6.4.

$$h_{ref} = a_{target} - R_{Earth} \tag{6.4}$$

Here $a_{target}$ is the target semi-major axis and $R_{Earth}$ is the Earth's radius (the Earth radius used in this calculation is set by the user). **Goal:** to have the optimizer prefer the vehicles that attain the highest altitude. One could assume that this would lead the optimizer to also accept vehicles that attain an altitude that is too high, this is, however, not the case. If the vehicle crosses the required threshold to be considered successfully, the last fitness calculation option is automatically triggered (see the last step and Equations 6.8 and 6.9).

**The vehicle is feasible and does not violate any constraints during launch but does not reach the target before using all the fuel**

$$f = -1 \cdot 10^6 \cdot \frac{s_a + s_e}{2} \cdot p \tag{6.5}$$

Here $s_a$ and $s_e$ are the semi-major axis and eccentricity scores. Whereas $a$ and $e$ are the semi-major axis and eccentricity respectively. Both the scores calculated in Equations 6.6 and 6.5.1 are correspond to the relative errors in semi-major axis and to the absolute one in eccentricity (converted into a 0 to 1 score for consistency, i.e. score 1 corresponds to a circular orbit). Note that an absolute error in eccentricity is, in a way, a relative error in "circularity" of the orbit, and already produces a value between 0 and 1 for the cases that are attributed this level of score. Eccentricities outside of this range are definitely possible but are never attributed this level of fitness and are therefore not an issue. The scores are then calculated as follows:

$$s_a = \left| \frac{a}{a_{target}} \right| \tag{6.6}$$

$$s_e = 1 - (e - e_{target}) \tag{6.7}$$

Where $e_{target}$ and $a_{target}$ are the mission's target orbital parameters (semi-major axis and eccentricity). In the previous version an absolute error was taken for the semi-major axis and eccentricity. The absolute error for the eccentricity would be multiplied by 1.000.000 to get it on the "same scale" as the semi-major axis. However, I believe that taking a relative error for the semi-major axis (as in Equation 6.6) and taking the difference between the target eccentricity and the current one is the more logical way of doing so. The reason for this is that the eccentricity is already dimensionless and is thus already a type of relative error. Eccentricity is a relative measure of an orbit's circularity the same way the relative error in semi-major axis measures the distance from the target value. The reason why the value $1$ is added to the eccentricity subtraction is to make it a score $s$, where $s \in [0, 1]$ where $1$ means that target eccentricity has been reached. In Equation 6.5, a multiplier $p$ was also included. The multiplier was included an aid to prevent the optimizer from getting stuck testing design sets that were not closer to achieving the mission target. The determination of the values of $p$ was a purely empirical and adjusted through trial-and-error until the optimization process reached acceptable speeds. The multiplier $p$ assumes the following values ($h_{ref}$ represents a reference altitude in meters related to the target orbit):

1. $p = 2.0$ if $a - R_{Earth} > 0$
2. $p = 4.0$ if $a - R_{Earth} > \frac{6}{10} \cdot h_{ref}$
3. $p = 8.0$ if $a - R_{Earth} > \frac{8}{10} \cdot h_{ref}$
4. $p = 16.0$ if $a - R_{Earth} > \frac{9}{10} \cdot h_{ref}$

**Goal:** Strive for the vehicle that is closer to reaching the mission target.

**The vehicle reaches the mission target successfully:**

$$f = -1 \cdot 10^{10} + 10 \cdot c \tag{6.8}$$

$$f = -1 \cdot 10^{10} + 10000 \cdot w \tag{6.9}$$

**Goal:** Minimize the cost/weight of the vehicle. Where $c$ is the mission cost in millions of FY2013 USD, and $w$ the GTOW in kg. Equations 6.8 and 6.9 are both implemented in the program as mutually exclusive options, **either the vehicle is optimized for GTOW or for Cost**. The choice of using a GTOW problem or a Cost problem is up to the user. This option was implemented in particular because there is no cost model (or enough historical data) for Hybrid Rockets. For Hybrid Rockets all the targets were optimized for GTOW.

### 6.5.2. Optimizer

In my study to improve performance of the tool, it was noticed that the optimizer, using the updated launch simulator, could no longer reach a solution that would meet the mission target. This led to a short investigation into the way the optimization worked and how (or if) it could be improved. In [4] two optimization processes are described: **Differential Evolution** (DE) and **Particle Swarm Optimization** (PSO). The tool as given to me was configured for Differential Evolution (using PaGMO's *pagmo::de* algorithm). In order to try and verify that the tool had continued to function, the priority became to get the optimizer to reach a solution. This investigation began with the changes to the fitness calculation process (see: 6.5.1) but this combined with the DE optimizer was not enough. The first experiments focused in changing the configuration parameters. Finally, the greatest performance

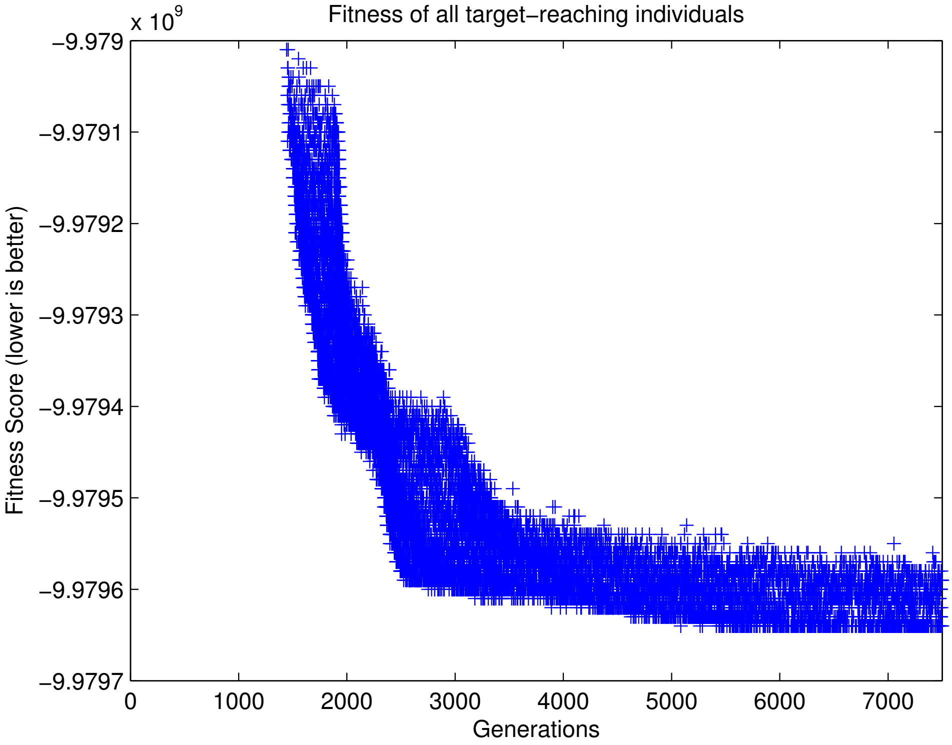


Figure 6.4: Fitness Evolution for the optimization of a 3-stage air launched solid rocket vehicle. The mission target used was an orbit of Semi-Major Axis $a \in [E_R + 770,\ E_R + 790]$ and an eccentricity $e < 0.01$. Only target reaching individuals are shown.

improvement came from changing to PSO. PSO (*pagmo::pso_ generational*) while configured to execute a **FIPSO** (Fully Informed Particle Swarm Optimization, see 5.1.3) a method through which the members of the particle swarm "are aware" of the fitness of their neighbours and use this information to improve their own fitness. Choosing FIPS came from reading [54] which states that FIPS outperforms regular PSO in every dependent measure. Setting this algorithm to use the FIPS strategy (available by setting pagmo::pso generational to strategy 6, a configuration parameter of the function) caused the performance to *improve spectacularly*. The new method reaches a feasible solution regularly under 2000 generations (when, for the new configuration, differential evolution produced no results) with a converged solution reached about 3000 generations into the process. Note that the PSO method does not keep all of the swarm's members compliant but instead using them to search new locations where better solutions might be. Figure 6.4 shows the evolution of the fitness value - only compliant solutions are shown. In the end, determining when to stop the iteration procedure can be done in three ways: stopping and a fixed number of generations, defining a rule based on the proximity of the solutions or defining a rule based on the convergence of the fitness value the mathematical relations for the last two options are shown in Equations 6.10 and 6.11 respectively. In these equations $i$ represents the current generation/iteration, $K$ the precision used in the rule, $f$ the fitness function value and $x$ the (multi-dimensional) solution value. Notice the usage of square brackets to indicate that it is the value of $f$ or $x$ *in a particular iteration* $i$ as opposed to being a function of $i$.

$$\left|f[i-1] - f[i]\right| < K \tag{6.10}$$

$$\left|x[i-1] - x[i]\right| < K \tag{6.11}$$

In this case, the optimizer was run for 7500 generations instead of defining a convergence rule, so that the behaviour of the fitness could be better observed.

### Constraint Handling

Not every set of parameters is considered valid. If the characteristics of the set fall outside of what is deemed acceptable for the problem at hand, such a set has to be discarded. However, doing so in a way that is not proportional to the amount it strays from the norm can lead to poor results. In the case of this project there are two levels of constraints:

1. Search Space boundaries
2. Indirect characteristics of the assembled launch vehicle, its trajectory and its physical properties

The **search space boundaries** are dealt simply by not letting the optimizer search outside of them. This is the simplest kind of constraint tackle. The **characteristics of the proposed solution** are more complex. They are divided in three main kinds:

1. Vehicle Design Constraints
2. Trajectory Constraints
3. Vehicle Integrity Constraints

**Vehicle Design Constraints** are the first kind to be addressed. These concern impossible designs, such as having high disparity between the diameters of consecutive stages or exaggerate L/D ratios (Length over Diameter).

**Trajectory Constraints** During the vehicle's launch simulation some limits are imposed on its trajectory such as exaggerate **angles of attack**.

**Vehicle Integrity Constraints** Represent the most indirect kind of constraint as they can not be estimated quickly in an analytical way. Thus, these require the simulation to be run in order to check for their violation. Examples are excessive levels of **Bending Load**, **Heat Flux** or **Acceleration**.

**Efficiency Concerns and Handling Methods** The way in which all the aforementioned constraints are handled is to **inflict a penalty** on the value of the fitness function. The strategy followed in [4] and [3] was the same, but during testing with the new version this was found to be less than ideal. Frequently **choosing the real level for a particular constraint leads to a poorer search due to the way the different optimizers work**. Often, setting the constraint to more flexible values will yield more solutions and in fewer generations, solutions that **in the end actually fall within the *harsher, more realistic,* constraint values**. Ideally the constraints would be implemented directly into the optimizing algorithm influencing directly the way the successive generations are created. This was not done here due to time constraints but it is a definite recommendation for future work.

### 6.5.3. Conclusions on Optimization and Fitness

In conclusion, improving the fitness determination method and changing the optimizer to a more appropriate one, proved fundamental to this research. Not only because *only after implementation could the tool reach any significant result* but also because a faster tool allows for more extensive testing and experimentation. It is worth noting that both this solution and the preivous one lack complete proof that convergence has been fully achieved, though in fairness this is always a problem when dealing with global optimization. However, the motivations presented are coherent and thus there is no reason to believe that the solution is *excessively* far from the actual optimum. *Excessively* being defined as the case where the distance from the optimum is greater than the error margins associated with the models used in the simulations.

| Property | Before | After |
|---|---|---|
| Fitness of Level 1 Solution | $+10.000.000$ | $+1.000.000$ |
| Fitness of Level 2 Solution | $-1\cdot h/1000$ | $-1000\cdot h/h_{ref}$ |
| Fitness of Level 3 Solution | $\frac{-1\cdot 10^{13}}{(\lvert a_t-a\rvert+1000000\cdot\lvert e_t-e\rvert)}$ | $-1\cdot 10^{6}\cdot\frac{s_a+s_e}{2}\cdot p$ |
| Fitness of Level 4 Solution | $-1\cdot 10^{30}/c$ | $-1\cdot 10^{10}+10\cdot c$ |
| Optimizing Algorithm | Differential Evolution (*pagmo::de*) | Particle Swarm Optimization with FIPS (*pagmo::pso_generational*) |
| Approx. Number of generations to Convergence (trial population of 70 individuals) | **10.000*** - Final Population 67/70 compliant individuals | **2.500** - Converged value reached |

Table 6.3: Comparison of the Optimizer strategies before and after the upgrade. The subscript *t* designates the target value for that same variable. The fitness levels correspond to the concepts discussed in Subsection 6.5.1.

*Value based on [4], however, in the results and appendix of [4], van Kesteren presents several examples where 3.500 and 2.500 generations were used. Since I could not get his optimizer to work for the case given in the same reference, the comparison in performance here can be considered biased.

# 7
# Implementation

This chapter discusses implementation of the models. The software and the models are the experimental park of this work. Chapter 6 discussed the software product as a whole and delved into the optimizing part of the code. However, a discussion on the implementation of Hybrid Rocket models to this project is done separately here. This chapter focuses on the details of the implementation, discussing the simulation of the stages' combustion processes and the estimation of their properties. This Hybrid Rocket Stage is a C++ object that is interchangeable with the Solid Rocket Stage object for vehicle design purposes. The implementation of the Solid Rocket Stage is discussed in detail in the thesis of Van Kesteren [4]. The final section of this chapter discusses the implementation of a Custom Rocket Stage, as described in the Software Chapter (Chapter 6).

## 7.1. Module Overview

The model for the hybrid rocket stages is divided into two main components: **Propulsion Performance** and **Geometry & Mass**. The propulsion module sizes the propellants according to input data. The Geometry & Mass module takes the information returned by the propulsion module and sizes the rocket's components accordingly. Both modules generate a considerable amount of internal variables. These variables are used to compute the final results of each module, but, the vast majority, is not very important once the modules are done running (and the simulation starts). These internal variables are kept for analysis later. However, the main results are mass flow and thrust for the propulsion module, and length, mass and cross-section for the Geometry & Mass module. The determination of these parameters is described in this chapter.

## 7.2. Propulsion

The propulsion module is responsible for two main tasks:

1. Determine the propellant requirements and size the mass properties accordingly.
2. Predict and record the history of the thrust and mass flow parameters.

This module is not responsible for the live simulation of the *thrusting process* itself. It is responsible for sizing and estimating the propulsion-related properties (see: 7.2.1) of the rocket and save this information to memory. The propulsion information is then used by the Mass module, Thrust module and Rocket Vehicle Module. The Mass module requires the propulsion information in order to size the dependent parts accordingly (see: 7.3). Finally, Thrust module requires the vacuum thrust curve to use against the local atmospheric conditions to simulate the thrust force (see: 7.2.3). The overall propulsion estimation and combustion simulation process is summarized in Figure 7.1. This figure is shown here because it links all the elements of this section together, however it shows the combustion process in greater detail than the rest. The combustion process is addressed in detail in 7.2.2.

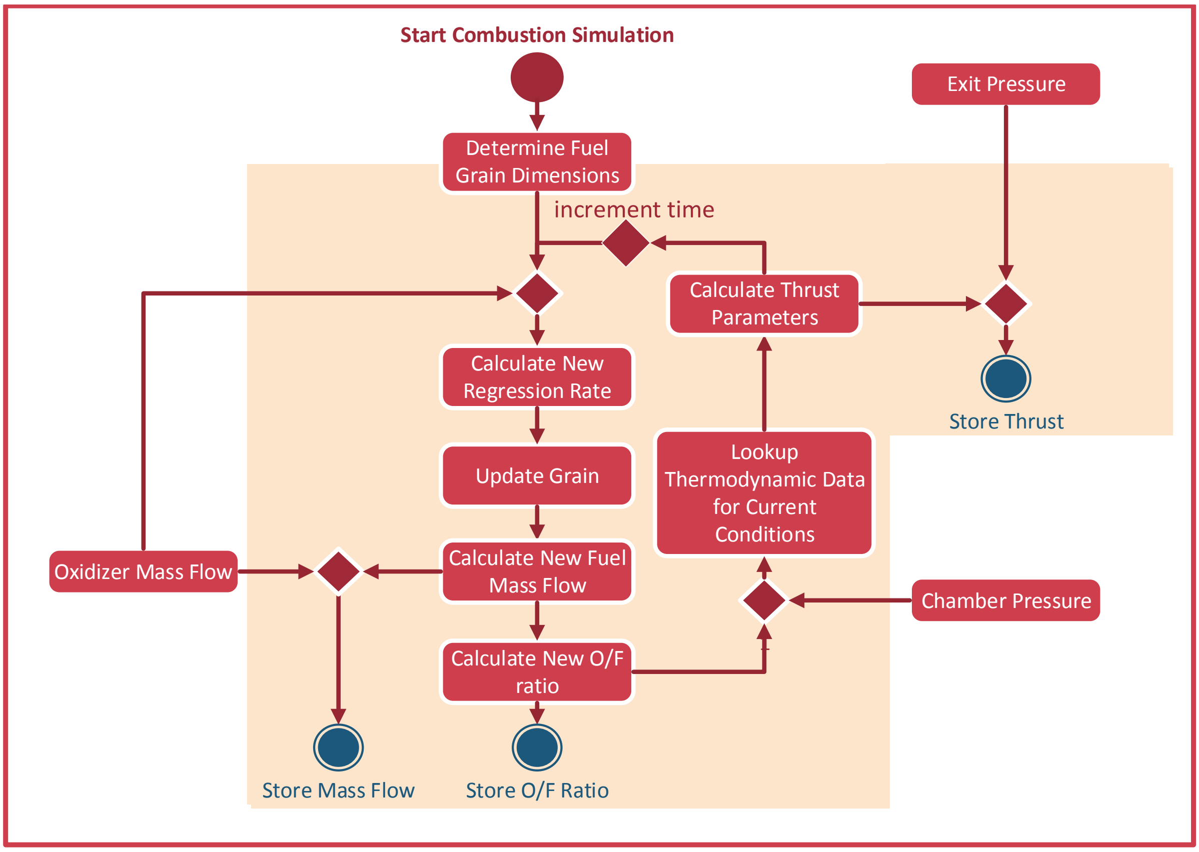

Start Combustion Simulation
Exit Pressure
Determine Fuel Grain Dimensions
increment time
Calculate Thrust Parameters
Calculate New Regression Rate
Store Thrust
Update Grain
Lookup Thermodynamic Data for Current Conditions
Oxidizer Mass Flow
Calculate New Fuel Mass Flow
Chamber Pressure
Calculate New O/F ratio
Store Mass Flow
Store O/F Ratio

Figure 7.1: Combustion process as simulated by the tool for a Hybrid Rocket Stage. The three main outputs are stored: evolution of Mass Flow, Thrust and O/F ratio along the process.

### 7.2.1. Fuel Grain Sizing

In comparison with the previous version of the tool, the Hybrid Rocket Stage implementation is far more complex than the solid one but it retains some of the important base characteristics, a comparison of some features was shown, from an architecture point of view, in Figure 6.1. The overall process goes as follows:

1. Calculation of expansion ratio through given chamber pressure and exit pressure (Equation 7.1).
2. Calculation of throat area through expansion ratio and the diameter of the nozzle exit (Equation 7.2).
3. Calculation of total mass flow through throat area, chamber pressure and thermodynamic data (Equation 7.3).
4. Use obtained mass flow and given O/F ratio to determine the oxidizer mass flow (already addressed in 4.18).
5. Use the oxidizer mass flow and regression rate law parameters to estimate the grain's required web thickness (Equation 7.6).
6. Use determined web thickness, and the fuel and oxidizer mass flows to determine the length of the grain. (from Equation 7.9)

The steps are sequential and depend mathematically on each other. These are implemented through Equations 7.1 through 7.3. **In Figure 7.1 all of the above steps pertain only to the "Determine Fuel Grain Dimensions" block**. The pressure ratio $p_{ratio}$ is used to the determine the expansion ratio $e_{ratio}$ which in turn is used to determine the throat area $A_t$.

$$e_{ratio} = \frac{\Gamma}{\sqrt{\frac{2\gamma}{\gamma-1} \cdot \left(\frac{p_e}{p_c}\right)^{\left(\frac{2}{\gamma}\right)} \left(1 - \left(\frac{p_e}{p_c}\right)^{\left(\frac{\gamma-1}{\gamma}\right)}\right)}} \tag{7.1}$$

The throat area $A_t$ can then be determined using the expansion ratio and the given exit area $A_e$.

$$A_t = \frac{A_e}{e_{ratio}} \tag{7.2}$$

With the throat area known the total mass flow, $m$, is determined.

$$m = \frac{p_c A_t \Gamma}{\sqrt{T_c R}} \cdot \xi \tag{7.3}$$

In Equation 7.3, $p_c$ is the chamber pressure, $A_t$ is the throat area, $\Gamma$ the Vandenkerkhove parameter, $T_c$ the chamber temperature and $R$ the gas constant. The parameter $\xi$ is a combustion quality, it is manually set by the user, typical values between 0.85 and 0.99 [23]. All the thermodynamic parameters ($T_c$, $\Gamma$ and $R$) are obtained using **thermodynamic data tables** that are read from a file. These tables must be provided for every propellant used (different fuels from stage to stage are possible). Following this an attempt is made to dimension the fuel grain properly. The word "attempt" is important here since what the program tries to do is conciliate all the important factors: burn time, mass flow, diameter of the motor case and the regression rate law of the propellants. Thus, expressions, 7.4 through 7.12 form a non-linear system that holds the solution for the grain design. In the center of this system lies the regression rate law. The law used in the program is of the form found in Equation 7.4 (for details on it see: 4.3.1).

$$r = a_0 \cdot \left(\frac{m_o}{A_p}\right)^k \tag{7.4}$$

In Equation 7.4, $m_o$ represents the mass flow of the oxidizer, which can be determined through the total mass flow and the user-specified Oxidizer-to-Fuel Ratio ($O/F$), since $O/F = m_o/m_f$. Since the goal is to carry as little fuel as possible, the amount of fuelmust be such that the **combustion burns through the grain completely along the burn time**. Since the regression rate is known, the grain is designed to **have such a thickness that it matches this requirement exactly**. In this implementation of the software, **only cylindrical grains with cylindrical ports were considered**. It is my recommendation to test, in the future, other kinds of grains, as the cylindrical burn is inherently inefficient in providing high mass flows and suffers from a considerable $O/F$ shift.

#### Grain Length

Taking the cylindrical port case, the requirement regarding fuel amount can be ensured by establishing Equation 7.5.

$$r \cdot t_b = W_T \tag{7.5}$$

Where $t_b$ is the burn time and $W_T$ is the Web Thickness. The Web Thickness for cylindrical ports corresponds to the thickness of the fuel grains annular cross-section. The relation between these parameters is illustrated in Figure 7.2 and Equation 7.6.

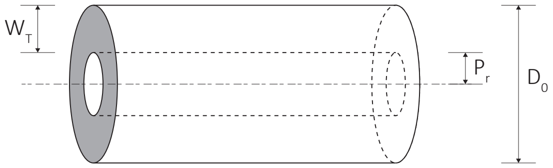


Figure 7.2: Representation of a fuel grain with a cylindrical port. The web thickness, $W_T$, port radius, $p_r$, and outer diameter, $D_o$, are indicated.

$$W_T = \frac{D_o}{2} - p_r \tag{7.6}$$

Substituting Equation 7.4 in Equation 7.5, and $W_T$ Equation 7.6, we obtain Equation 7.7.

$$a_0 \cdot \left(\frac{m_o}{A_p}\right)^k \cdot r = \frac{D_o}{2} - p_r \tag{7.7}$$

Considering only cylindrical cases, the port area, $A_p$, and $W_T$ can be written as a function of a single variable: the port radius $p_r$. By doing so and substituting in Equation 7.7, we can find a port radius that fits it properly. Equation 7.7 is, however, non-linear, meaning it can have multiple solutions or even negative solutions. The **negative port radius solution** implies that the fuel burns through too quickly for the grain diameter provided. For these solutions, even if the whole case was filled with fuel it would still not be enough to last the required burn time at the specified regression rate law. When there are **two proper solutions**, the smallest one is usually the best, since a smaller port radius implies a smaller grain length. Smaller grain lengths are preferable in most cases since it reduces the mass of the rocket case. In the case of no solution or negative solution the program sets the value of the port radius to half of the case diameter. All the solutions are obtained by the tool using an iterative Newton-Raphson method solver. **The program also supports a mode that uses a fixed regression rate**. In this circumstance, the program calculates the port radius by simply solving $W_T/t_b$.

### Grain Length

Once the port radius is known the grain length can be determined. The grain length influences directly the fuel mass flow. This because the fuel mass flow is dependent on the burning surface. For the case of a cylindrical grain, the burning surface, $S_b$, is the surface area of a cylinder of length $L$ and radius equal to that of the port radius $p_r$ as shown in Equation 7.8.

$$S_b = 2 \cdot \pi p_r \cdot L \tag{7.8}$$

The fuel mass flow, $m_f$, of a fuel grain with regression rate $r$, for a cylindrical grain burning radially is then given in Equation 7.9, from substitution of 7.8 in Equation 4.6.

$$m_f = \rho_f \cdot r \cdot 2 \cdot \pi p_r \cdot L \tag{7.9}$$

To maximize the energy from the combustion we attempt to set a grain length such that the $O/F$ ratio specified by the user is achieved. This is a design choice that originates in the fact that the Oxidizer Mass Flow is fixed. Since all parameters vary with the port radius, and the port radius is the parameter that varies the most along the simulation, we must pick a moment to use and choose that as the reference port radius. The effects of the varying port radius have already been discussed in 4.3.1. To select the port radius to use, either some method of maximizing the time in which optimum O/F is achieved is used or a single reference value is selected. Adding another optimization loop would considerably increase the computational time and such a detailed approach to grain simulation is beyond the scope of this project. For these reasons, the tool assumes the variation of $p_r$ will be small enough such that the **initial value can be used**. By observing the results of a couple of trials, the user can overcompensate and demand a higher (or lower) O/F ratio from the program, therefore **adjusting manually for this effect**. Equation 7.10 is the result of applying the O/F ratio requirement to Equation 7.9.

$$\frac{m_o}{O/F} = \rho_f \cdot r \cdot 2 \cdot \pi p_r \cdot L \tag{7.10}$$

The final equation is, once more, non-linear, since $r$ has to be substituted by Equation 7.7. *A second newton-raphson method solver is introduced to solve this equation as well.*

$$m_f = \rho_f \cdot r \cdot S_b \tag{7.11}$$

For the case where the **regression rate is constant** the solution is simpler. In fact, in this case the O/F ratio is ensured to be the specified one at half-burn to approximate an optimized burn. The relation that illustrates this is shown in Equation 7.12.

$$m_f = \rho_f r 2\pi \left(\frac{D_o}{2} + r\left(t_{1/2} - t_b\right)\right) L \tag{7.12}$$

Where $t_{1/2}$ is the half-burn time. For the more complex case where the **regression law is used**, a similar approach is taken but it involves substituting $r$ in Equation 7.12 by its law (the one in Equation 7.4). Notes on the process described above for grain design:

1. Regression rate is average over the length of the grain.
2. All equations are solved using a simple newton method implemented.
3. Enabling different grain shapes is the most significant future recommendation for work on this part. It is my opinion that the removal, or mitigation, of the O/F shift would bring significant improvements in performance.

With the length determined the fuel grain is completely sized and the combustion process can begin to be simulated.

### 7.2.2. Combustion Process

Even before estimating the dimensions and mass of the rocket, the combustion process is simulated and stored. The process is analogous to using a raw thrust curve, but in this case the thrust curve is simulated according to the sized grain. The combustion process makes use of the Equations developed in 7.2.1 and those in 4.3.1. The general principle is that a thrust curve should be calculated to the same time-resolution of the launch simulation itself. To generate the curve, the values of the thrust are calculated once per instant. After each-time step the grain information is updated to account for the part that burned. With the grain updated, the remaining dependant parameters can then be calculated. In Figure 7.1 the combustion process corresponds to the blocks inside the shaded area. The process is repeated in a loop until the end of the burn time is reached. This process is illustrated in Figure 7.1, where each step is addressed by an single equation:

1. **Calculate Regression Rate** - Equation 4.7.
2. **Update Grain** - Equation 7.7.
3. **Calculate New Fuel Mass Flow** - as indicated by the figure, by combining Equations 7.3 with 7.9
4. **Calculate New O/F Ratio** - Trivial since both fuel mass flow and oxidizer mass flow are known: $O/F = m_{ox}/m_f$.
5. Lookup Thermodynamic Data - See Subsection Propellants and 6.3.3 for the details on this step.
6. **Calculate Thrust Parameters** - direct implementation of Equation 4.2 since all the parameters are now known.

At each loop, the values for the thrust, mass flow, O/F ratio and regression rate are saved as curves. The thrust and the mass flow are essential to for the launch simulation itself. The remaining parameters are kept so that an evaluation of the rocket's performance can be done on multiple levels. The Rocket Vehicle simulation module uses the generated curves to know what the thrust force is in every instance, combining the information with the pressure one from the Atmospheric Models.

### 7.2.3. Thrust Module

The thrust module is not a direct part of the propulsion module. It is instead an external part of the program that takes the pre-computed information generated by the propulsion module, and uses it in every instant along with environmental information to generate the actual thrust level. Essentially, the vacuum thrust curve produced by the propulsion module, or the one manually provided in the case of custom stages (see: 7.4), does not directly correspond to the actual thrust produced by the rocket. Calculating real thrust requires live information of the atmospheric pressure at the current location of the rocket vehicle. This information is available to the simulator as a whole but not to the propulsion module itself. The reason for this program design is that the sizing of the rocket is done before the simulation is actually run, and projecting the thrust levels before actually deciding to go ahead with the launch simulation - the most computationally intensive part of the program - saves on computational time. Therefore the thrust module has to exist outside of the Propulsion Module and the Vehicle building parts of the program to be able to like the environmental information along the propagation process with the thrust curve saved during the assembly of the vehicle. Its main goal is to introduce the atmospheric pressure effect on the thrust along the flight. This is shown in Equation 7.13.

$$T_{real} = T_{vac} + A_e \cdot (p_e - p_a) \tag{7.13}$$

Here, $T_{real}$ represents the actual value of thrust produced by the rocket after accounting for the atmospheric pressure effect. The value of $T_{vac}$ is extracted from the pre-computed curve (or given curve, in the case of custom stages) at the current instant. The thrust module is also responsible for converting the thrust force into an acceleration. As explained in Chapter 6 the program's integrator is acceleration-based meaning it requires all the acting accelerations in order to compute the state derivative. Since the information this module outputs is a thrust *force* it must be converted to the corresponding *acceleration*. This is done using Equation 7.14.

$$a_{thrust} = \frac{T_{real}(\tau)}{M(\tau)} \tag{7.14}$$

Here $a_{thrust}$ is the acceleration induced by the thrust force and $M(\tau)$ is the total spacecraft mass at the current instant $\tau$. $T_{real}$ maintains its adding the specification of the instant to be consistent with the value for the mass, i.e. *current* thrust must be paired with the *current* mass.

## 7.3. Mass & Geometry

The Mass & Geometry module of a Hybrid Rocket Stage is divided into the following components:

1. Lifting Surfaces
2. Nozzle
3. Fuel, Case and Combustion Chamber
4. Oxidizer and Oxidizer Tank
5. Pressurant Gas and Pressurant Gas Tank
6. Global Parameters

Components **1, 2 and 3** are implemented just like in [4] and follow strictly the models from Chapter 4. The structural components related to the fuel (component **3**) are modelled quite similarly to a Solid Rocket Stage. Components **4, 5 and 6** were developed from scratch. The approach to doing so is described in this section.

### 7.3.1. Oxidizer and Oxidizer Tank

When dimensioning the Oxidizer Tank there some things have to be asserted: whether or not a pump system is being used, and what are the storing pressures and densities of the gases inside the tanks. The pressures are particularly significant since the oxidizer's density depends on its phase and therefore on its pressure. Only by knowing the density of the oxidizer can the required volume for the tank be computed. To acquire this information there are two design options:

1. Set the tank pressure as a "maximum allowed pressure"
2. Prove tables with phase information so that the program can determine the density for a particular pressure

Option **1** is the simplest one, however, there are some limitations to it. By setting the pressure manually, and the oxidizer's density along with it, we are fixing the pressure at the exit of the tank. For systems that do not have pumps, this pressure must always be higher than the operating pressure. Otherwise the rocket cannot operate. The simple solution is to accept such a limitation and set the optimizer **to only search solutions with a chamber pressure lower** than the one provided for the oxidizer tank. If the system has a pump that has been sized correctly then the pressure can be set as wished as long as enough power is provided to the pump. Option **2** is the most complete one. By providing the program with phase information on the oxidizer, the tanks dimensions can be sized for any pressure. If this is the case, the pressure on the oxidizer tank is taken as the chamber one multiplied by a design factor, $DF$, and by a coefficient to account for pressure drops, $\Delta p_{loss}$, in the feed system. This is illustrated in Equation 7.15. Currently this second option is not implemented due to the lack of data for most cases and the lack of time to implement this in the late stage of the project. This is left a future recommendation.

$$p_{ox} = DF \cdot \Delta p_{loss} \cdot p_c \tag{7.15}$$

With the value for the oxidizer pressure $p_{ox}$ is known, and, with it, the density to use. These parameters enable the sizing of the oxidizer tank. As explained in 4.3.2, **the Oxidizer Tank is dimensioned as a thin-walled cylinder** with thin-walled hemispheres as caps. The program first attempts to fit all of the required oxidizer in a sphere no larger than the outer diameter set by the oxidizer. In case a single sphere is not enough, the program adds as much cylindrical section as required. **The thickness of the tank**, Equation 4.21, is dependant on $p_{ox}$ as well.

### 7.3.2. Pressurant Tank and Pressurant

For the systems that require a pressurant tank with a pressurant gas, additional models are required. Mainly one must calculate how much pressurant is necessary. After this is determined, the dimensioning for the pressurant is the same as for the oxidizer tank. In [31] we find that there must be enough pressurant such that, when released and expanded to the pressure required inside the oxidizer tank, it fills it completely. Once more, either the data on density and pressure for the pressurant gas is provided in the form of thermodynamic tables, or a few maximum values are set. The same limitation that applied to the oxidizer in relation to the chamber pressure applies here. In the simplest of solutions at least three values must be provided:

1. Pressurant storage pressure - to size the tank that holds it.
2. Pressurant density at storage pressure.
3. Pressurant density at the pressure it expands to inside the oxidizer tank.

Once these are known, through either process, a relation analogous to Equation 7.15 is established and the pressurant tank system dimensioned. This is based on the process from the lecture notes in [31].

### 7.3.3. Pumping System

The use of a pumping system, as described in 4.3.2, is often beneficial from a GTOW standpoint, especially when compared to a pressurant tank option. The pumps are a usable option in the software. The program sizes the pumps based on mass and required power. The relations for estimating the pump values come from [8]. The process followed to size the pump is as follows:

1. Determine the required power using: oxidizer mass flow, $m_{ox}$, pressure difference between the oxidizer tank and the chamber, $\Delta p_{pump}$ and the density of the oxidizer $\rho_{ox}$.
2. Once the power is known, the pressure ratio is computed (since the estimating relations require it).
3. With the Pressure Ratio, and the required Power, the Mass is determined.
4. The Power Density, $P_{delta}$, is then determined, also through the relations from [8].
5. Finally using a pre-set value for the L/D Ratio (Length over Diameter) of the pump, the Length and Diameter are determined.

The programs causes pumps with a diameter much larger than the case of the rocket to make the rocket invalid. The default value for the L/D ratio of the pumps is the mid-range value from [8]. This means that 1.5 is used for the small pumps and 1.05 for the larger ones. The relations used in this part of the modelling are presented in Chapter 4 in Table 4.3. The length of the pumps is covered with a case like that of the fuel grain. Sized according to the specified density and wall-thickness.

### 7.3.4. Total values

In the end, from the Mass & Geometry module, the Launch Simulator requires very little information. Only the length of the stage, the diameter of the cross-sectional area and the total mass.

1. The length of the stage is given by adding all dimensioned components (fuel grain information taken from the propulsion module).
2. The cross-section area, is given by the diameter of the widest stage.
3. The total mass is given by adding all the components mentioned in this chapter (propellant and structural mass are separate here).

## 7.4. Custom Rocket Stage

The simulation of existing stages with proven experimental data was an essential part of the cooperation with Nammo and DARE. Since the Hybrid Rocket Module of the program achieves to size its own stage, this code is not appropriate for such a use. When all data is known, there is no point in estimating most parameters. For the simulator, the significant parameters are:

1. Overall Mass
2. Variation of the Mass along the burn time
3. Variation of the Thrust along time
4. Exit pressure on the nozzle
5. Diameter of the cross-section
6. Total Length

The parameters above, couple with the launch circumstances, are all that matters for a full simulation to be achieved. As such the custom rocket stage was created. When a Custom Rocket Stage is used, the user sets all of the parameters in the list. The Rocket Vehicle is specified nonetheless, with instructions of the Thrust Scheduling (Coasting times and Burn Times), and on the Trajectory to follow both being required. A full simulation can then be run on these settings.

# 8 Verification and Validation

Without the proof that the models developed are realistic and reliable there is no point in developing them. In order to perform V&V we must first discuss the meaning of this process. NASA has their definitions of Verification and Validation, stated in their NASA Procedural Documents, specifically NPR7120.5 [57].

> "The verification process compares the product to the specifications to see if it meets them. Validation, however, compares the product to the stakeholder's expectation to make sure those are met (i.e., that the right product was built). If the requirements and succeeding specifications were not correctly based on the stakeholder's needs, the product could be verified but it still won't meet the stakeholder's needs and can't be validated. Therefore, the validation process starts at the beginning of the development by validating the requirements with the stakeholder and continues throughout the life-cycle process. The final validation takes place on acceptance and usually consists of some type of end-to-end test in which the proper operation of the system is demonstrated (i.e., that the right product was in fact built)." [57]

This chapter's sections go through the Verification and Validation for this project.

## 8.1. Verification

For the Verification, the tool's intended specifications will be the main gauge of how it performs. The specifications come from the goals outlined in 1.4, the goals are the following:

1. Simulate Hybrid Rocket launches, air- and ground-launched, to the same standards that the current software does for Solid Rockets.
2. Produce the optimum, measured by lowest GTOW, Hybrid Rocket vehicle configuration for a particular target mission.
3. Produce not only the optimum Hybrid Launch, but also allow for multi-stage vehicles to include stages of both types and get the optimum launch for mixed (hybrid and solid stages) configurations as well.

One must then defined what "optimum" solutions and what are the requirements in terms of inputs that should produce those outputs. An optimum solution is a solution from which the program cannot improve upon. The program was not outfitted with convergence rules for either the fitness or the solutions' as discussed in Subsection 6.5.2. The reason for this is that there is no reasonable motivation behind choosing a particular value range for the rules. As such, the **optimum value is defined as the value of the best solution after a fixed number of generations**. It is up to the user to observe the fitness figures and determine whether the value had converged enough for the problem at hand. Another concept that requires clarifications is "the same standards" as achieved by the previous version of the tool. The standard is summarized as follows:

1. The level of complexity in terms of number of modules and general parameters of a rocket should be the same
2. The complexity of each module is the same or greater

**With the concepts defined we proceed to assess whether or not the tool is verified.**

1. *Simulate Hybrid Rocket launches, air- and ground-launched, to the same standards that the current software does for Solid Rockets.* ✓

   - (a) The same number of modules exists for hybrids as it did for solids: Propulsion, Geometry and Mass, Environmental (Gravity and Aerodynamics), Thrust and Control
   - (b) The same general parameters that characterized a solid rocket in [4] are also obtained here: stage length, inert mass, propellant mass, diameter of the cross-sectional area, thrust data and mass flow data.
   - (c) The hybrid model is actually even more complex than the solid one, simulating also combustion, grain regression, and variable thrust and mass flows. The geometry model accounts for more parts like pumps and pressurant tanks making it also more complex than in the solid case which had no need for this.

2. *Produce the optimum, measured by lowest GTOW, Hybrid Rocket vehicle configuration for a particular target mission.* ✓
   - (a) This is effectively achieved, the program successfully optimizes an n-stage hybrid rocket looking for the configuration with the lowest GTOW. An extra option: maximizing left-over fuel mass (for problems were the trajectory was the only optimized part) was also added.
   - (b) In general the definition of a convergence rule by the user would be possible. Right now what I found was the best practice was to run a broad search and check if the fitness of the solutions tended towards an asymptote-like behaviour, once this is achieved, the search-space can be refined and a better optimum found. I leave the improvement of this point as a recommendation for future work.

3. *Produce not only the optimum Hybrid Launch, but also allow for multi-stage vehicles to include stages of both types and get the optimum launch for mixed (hybrid and solid stages) configurations as well.* ✓
   - (a) As discussed in Chapter 6 the new architecture supports multiple technologies in the same rocket
   - (b) Custom stages that are independent of the technology that powers them and that have their data fed manually (performance curves, mass and geometry data) were also added.

The tool is considered to fulfil its purpose and to be verified as per the goals established. Examples of the aforementioned properties of the software and examples can be found in Chapter 9 with the main results of this research.

## 8.2. Validation

When continuing upon someone else's work validation is different than when for work from scratch. The models that come directly for previously validated, good quality, work will not go through validation a second time. However, by altering previous work a version consistency validation must be made.

### 8.2.1. Modules from previous work

Some models come directly from [4]: nozzle sizing, payload and VEB sizing, and the gravity model, these models will not undergo V&V. These models are considered suitable, having already undergone this process by the authors. There are two important aspects to consider when deciding to keep the modules: whether they have to be adapted for hybrid rockets or not and if using them without adapting them incurs in added error to the hybrid rocket models. The gravity model is obviously applicable and still valid, one could make some changes to it, like including effects beyond J2 for added precision, but this is unrelated to whether the rocket is hybrid or solid and thus not the issue at hand here. The payload and VEB sizing aspects are also, at least for the purposes of this work, considered completely independent from the technology that powers the rocket. However, there are, for example, payloads that are sensitive to acceleration and have to be transported in lower acceleration conditions. This could result in a situation whether one technology might have different requirements for the payload adapter when compared to the other to compensate for this fact. This is deemed to be above the level of precision used in this study and beyond the scope of this investigation and therefore left as a future recommendation for further work. Finally the nozzle sizing can also be kept intact since the propulsion principles behind its design are independent of the technology behind it at least to the precision of this work. The modules that are the most important to validate here are Propulsion, Geometry and Mass. These are significantly different from previous work and will be compared to sources of experimental data encompassing a variety of different rockets. The cases with which this project cooperates are the most suited to perform V&V since there is more data available. Particularly from: [13, 15] (DARE) and [12] (Nammo). Some additional information provided through personal communication is also used.

### 8.2.2. Validation for consistency between versions

Since the tool was updated it must undergo verification and validation in order to check if it is still *functional* and what are the changes in its results that follow from the corrections described in 6.4.1. Here *functional* is defined as a state in which the tool can perform the same operations that it could do in the previous version, primarily:

1. Individual simulation of 2 to 4 solid-stages rocket launches both air and ground-launched.

2. Vehicle design optimization for achieving particular mission targets using the same range of vehicles defined in item 1.

### Output comparison for a sample launch between versions of the tool

The main goal of this project is to explore the possibilities of the hybrid technology along with the development of the tool. Upgrading the tool then presented an interesting challenge validation-wise. Addressing this is the purpose of this subsection. In order to compare the tool's output to its previous version the sample launch chosen should encompass the broadest range of features possible. For this reason a 3-stage air-launched rocket scenario was chosen.

**Atmospheric Model Comparison Tests** The faulty atmospheric model error was only noticed post-upgrade. Since the tool *did not run* prior to the Tudat version upgrade and the fixing of the compiling and build process, a comparative study of both could not be done here.

#### Test parameters

1. 3-Stage Rocket.
2. Air-Launched at 15 km altitude and 250 $\mathrm{m\,s^{-1}}$ initial velocity.
3. Latitude: 0° Longitude: 0°.
4. Payload Mass: 10 kg.
5. No lifting surfaces.
6. Propellant Configurations from [4] for direct comparison where possible.

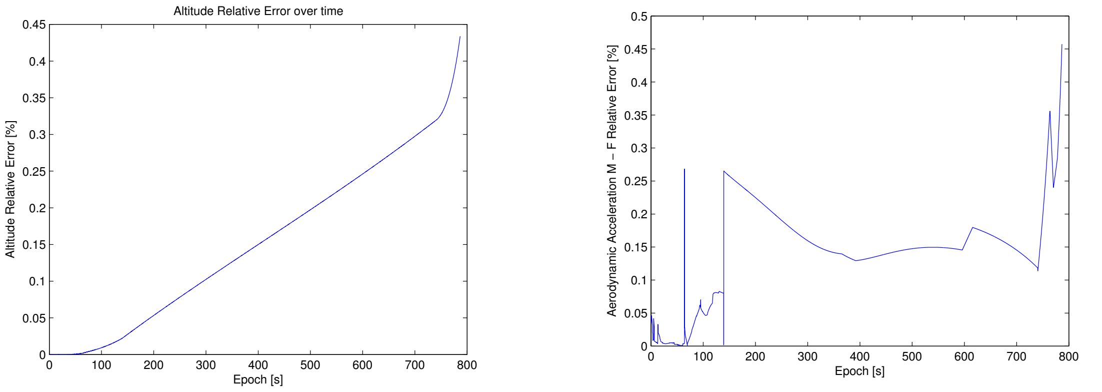
**Thrust Acceleration** The thrust acceleration showed differences lower than 1% between both tools and only in outlier points caused by a 0.02 s shift between both time-frames. This value is considered irrelevant in the context of this software and thus the thrust acceleration values, yielded by the new version of the tool, are considered to be working and satisfactory.

**Aerodynamic Acceleration** Figure 8.1b shows a comparison of the aerodynamic acceleration values for this test. The differences derive mainly from the change in the integration process described in 6.4.1. Again, since the previous atmospheric model could never be reproduced entirely, this test is more of a comparison of the effects of changing the integration process. These have implications on the calculation of the density and pressure in the vehicle's surroundings. The density and pressure affect, through a feedback loop effect, the drag and therefore the velocity of the vehicle which affects the aerodynamic parameters again. The relative error averages 0.25% with a peak near 0.5%.

**Altitude** The altitude comparison is perhaps the most interesting one to explore. Since the error that was corrected in the software was intrinsic to the integration process itself, the error - expectedly - rises as the time (the epoch) advances. Shifts on the parameter's derivative can be observed to be synchronized with application of thrust. For the total action time of approximately 800 s the relative error peaks in the final instant at approximately 0.45%, as seen in Figure 8.1a. This value represents a deviation in the order of 30 km.

### Conclusions on the upgrades

The tool is considered to be re-validated and to perform as intended even though past solutions as presented in [4] no longer seem to be valid. This fact can be due to either the improvements in the integration process or the new atmospheric model used but could also be an improper steering law. The solution presented in [4] only shows the vehicle configuration and not the pitch angle history thus it is possible, though unlikely, that there is a way of making the presented solution work.



(a) Altitude - relative error in percentage

(b) Aerodynamic Accel. - relative error in percentage

Figure 8.1: Error plots comparing the results for the air-launch of a 3-stage solid rocket at 15 km altitude and 250 m/s carrier aircraft velocity. Comparison for the same test case in both versions of the tool.

### 8.2.3. Validation of the Hybrid Rocket Stage

Even though there is not enough data available on full blown Hybrid Launch Vehicles, there were quite some engines built and tested. Since a Rocket Vehicle's main components are its stages this section attempts to validate the models for them. **Some of the data in this chapter is redacted because it's proprietary data.**

#### Nammo Rocket Engine

From [12] and a personal communication from the company, the following details on the experimental set-up are given:

1. Fuel: HTPB/C
2. Oxidizer: 87.5% Hydrogen Peroxide.
3. Chamber Pressure: 35 bar
4. Grain Outer Diameter: 320 mm.
5. Motor Length: [redacted].
6. Regression Rate: 1.48 or 2.49 mm $s^{-1}$ (Axial and Vortex Injection respectively) [12].
7. Expansion Ratio: [redacted].
8. Combustion Quality: 0.99

**Geometry and Mass:** Table 8.1 shows how the Geometry and Mass properties of the simulated stage compare to the actual engine. The parameters are within expected margins, with errors close to [redacted]. The differences in mass are most likely caused by the enforced O/F ratio which my attempts to keep as close to optimum as possible. This behaviour is not specified for the Nammo test set-up simply stating that the optimum value was used. The fact that Nammo states 2 alternatives for the average regression rate: 1.48 mm $s^{-1}$ and 2.49 mm $s^{-1}$ is also a possible source of error. I used the more optimistic value of 2.49 mm $s^{-1}$ leading me to underestimate some of the values. The larger difference is noted for the motor length. The simulation value: [redacted] is the actual grain length only. The experimental set-up included a post-combustion chamber and a catalyst unit this is illustrated in Figure 8.2. Provided that the illustration is to scale are almost the same size as the fuel grain. Meaning that an equivalent set-up, using the simulated fuel grain, would actually twice as long: [redacted], much closer to the indicated [redacted].

| Parameter | Real Engine | Simulated Engine | Relative Deviation |
|---|---|---|---|
| Total Propellant Mass | [redacted] | [redacted] | [redacted] |
| Fuel Mass | [redacted] | [redacted] | [redacted] |
| Oxidizer Mass | [redacted] | [redacted] | [redacted] |
| Oxidizer Volume | [redacted] | [redacted] | [redacted] |
| Motor Length | [redacted] | [redacted] | [redacted] |

Table 8.1: Geometry & Mass properties of the simulated engine compared to the real engine.

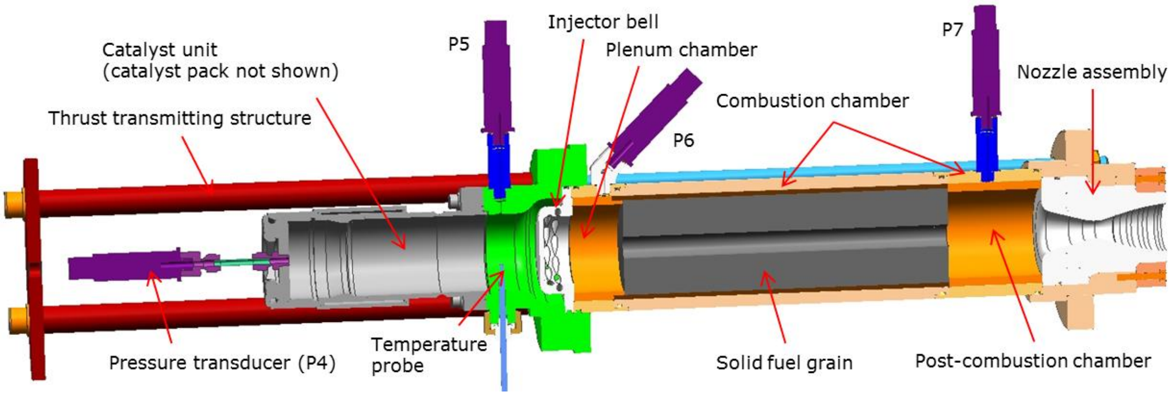


Figure 8.2: Lab-scale engine by Nammo, from [12].

**Propulsion Performance:** In order to validate the propulsion models, the settings were set as close as possible. Note that the model generates some settings automatically (like the grain's length). The results from a simulation with the above settings are shown in Table 8.2. The values in Table 8.2 are coherent with the expectations. Specific impulse is over estimated by less than 7% and the thrust level underestimated by less than 5%. Mass flow seems to be the value that is shifted the most from the target and yet it is only approximately 12% off. These values are considered good enough to deem the tool as valid. The small deviations can be caused by some assumptions and simplifications, namely the fact that HTPB/C is a proprietary mixture (as discussed in Propellants) which makes it partially unreproducible on a model. The values present the proper order of magnitude, what remains is to analyse the behaviour of these parameters through the respective performance curves. The performance curves corresponding to the values in Table 8.2 are presented in Figure 8.3. The most important information to gather from these curves is summarized in the following list:

1. In the $I_{sp}$ vs. $O/F$ curve one sees the expected peak at the optimal O/F ratio and an appropriate decline before and after. This is compliant with the theoretical predictions in Figure 4.5. The curve in the bottom right image of Figure 8.3 shows the value and the slope are in the right neighbourhood.
2. The thrust curve (top left) shows an increase in thrust over time, this is the expected behaviour since the

| Parameter | Model Mean Value | Test/Provided Data | Difference |
|---|---|---|---|
| Specific Impulse Vacuum (s) | 277.0338 | - | - |
| Specific Impulse Sea Level (s) | 248.1069 | 232.01 | +6.938% |
| Mass Flow (kg s$^{-1}$) | 10.9801 | 12.3 | -12.021 % |
| Vacuum Thrust (kN) | 29.844 | - | - |
| Sea Level Thrust (kN) | 26.729 | 28 | -4.7 % |

Table 8.2: **Mean values** for the main performance characteristics of a stage simulated to the same settings as the Nammo Hydrogen Peroxide - HTPB/C engine and comparison to provided data.

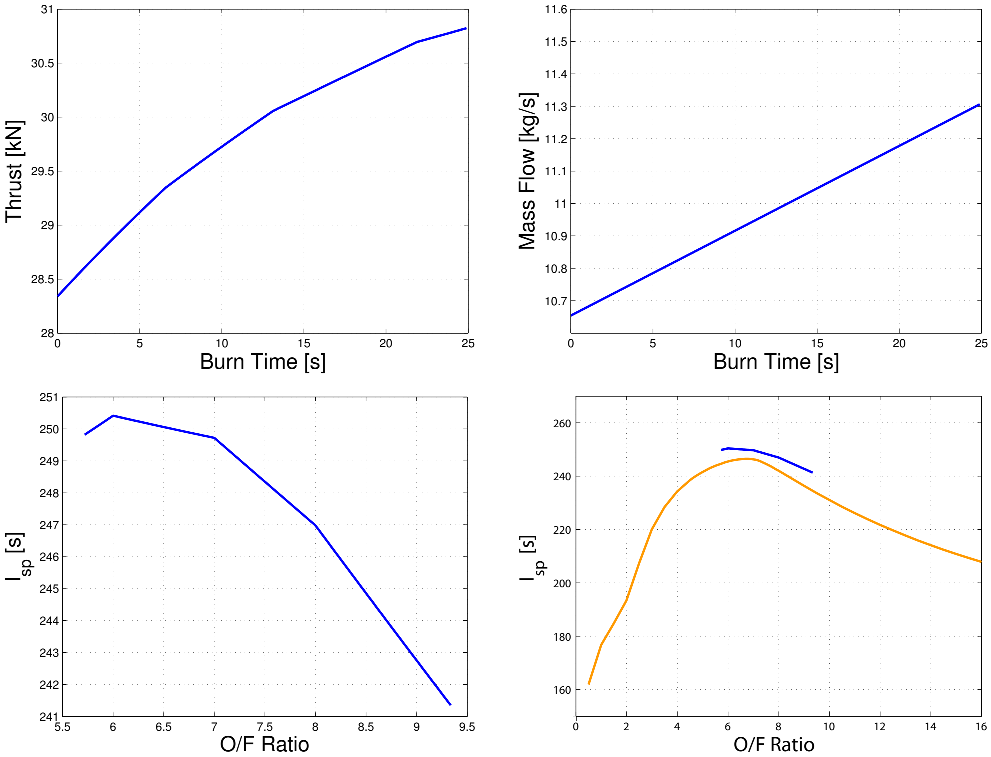


Figure 8.3: Performance curves for the test stage simulated to Nammo's engine specifications. The bottom curves both illustrate $I_{sp}$ as a function of the O/F for the burn duration. **Bottom left** is a detailed view, and **bottom right** is the same curve but overlapped with the theoretical curve as computed by RPA.

burning area increases which increases mass flow (given that $m_{ox}$ is constant) therefore increasing the thrust level.

3. The mass flow curve shows the expected increase for the reasons explained in item **2** of this list.

### Conclusions on validation of the hybrid rocket stage

Though the data used here is not a very big I consider the principle to be proven. The tool is capable of correctly estimating performance parameters for a given rocket engine within a reasonable accuracy. Reasonable is defined here as having the right order of magnitude and displaying the expected behaviour and evolution. Different assumptions and simplifications could yield better results and this in on itself shows the flexibility of what was built and the quality of the results that can be achieved. This simple example with the Nammo stage showed positive results that allow conducting the tests in Chapter 9 with a great degree of confidence. More engines and more fuel combinations could have been tested but this is quite an exhaustive process and time constraints prevented a survey of more fuels. The time that has to be invested in producing appropriate thermodynamic property tables and the correct quality factors (like the combustion quality and thrust coefficient) was deemed to be too much for this work. However it could definitely be done by future students using the same software for a broader of different research or even as a part of a scientific paper that may result from this thesis using the tool to investigate other hybrid rocket engines.

# 9
# Results

The development of this tool enabled a large array of possible tests. Answering the research question (see: 1.3) is the main goal but there are other important ones. In the domain of the collaboration with the companies and rocket groups that might be end-users of the software, there are a lot of possibilities as well. Simulating actual cases, for instance, is one of the more interesting possibilities. Re-purposing technology is also an interesting goal: air-launching rockets that are usually ground-launched for instance. Creating and experimenting with the performance of mixed rockets (Solid and Hybrid Stages in the same rocket) is also of great interest. This chapter describes all the results obtained in the research. Firstly with the cooperation with the external entities and in the end attempting to answer the research question.

## 9.1. Tests Based on Existing Rocket Engines

As mentioned extensively throughout this report, the collaborations with Nammo and DARE were quite important due to the data that they made available. This section explores their technologies and what results the optimizations and simulations yielded.

### 9.1.1. Nammo Hybrid Rocket Engine

Nammo developed a Hybrid Rocket engine based on Hydrogen Peroxide and a fuel they designate by HTPB/C a variation of the traditional Hydroxyl-Terminated Polybutadiene. The first step in simulating such a system is to get the thermodynamic data for the fuel-oxidizer pair. This was computed using CEA. The settings for the fuel are listed in Table 9.1. The fuel-oxidizer combustion was simulated for all the pressures between 30 and 100 bar and for O/F ratios between 1 and 9, both with intervals of 1. The tool interpolates between O/F ratios but rounds the pressure to the closest point. If the O/F ratio of a particular rocket goes outside of the data, the tool **automatically extrapolates** linearly based on the two closest points. From [12] we find that the regression rate for this experiment was of about 3.5 mm/s. No regression rate law could be found for this particular fuel-oxidizer pair, thus the fixed rate was the setting given to the tool. Giving an average fixed rate can result in an under or overestimation when compared to a power law. The exponent and coefficient of said law can vary significantly (see Subsection Propellants). Thus, the error between a power law and the chosen fixed value can also vary significantly. However, for the precision of this work average values seem to yield valid results nonetheless as seen in Chapter 8. I leave as a recommendation for future work to investigate the influence of using power laws vs. an averaged value for the regression rate. Since implementing both options was the main personal goal of this work which was acheived, it was chosen not to investigate this as well due to time constraints. As for the feed system details, the paper by Nammo describing their experimental set-up shows a pressure-fed system, with an external tank pressurising the Oxidizer using air.

| Setting in CEA | Fuel - HTPB | Oxidizer - $H_2O_2$ |
|---|---|---|
| Formula | $H_{302}C_{240}O_2$ | 87.5% H2O2 : 12.5% H2O |
| Heat of Formation | 997.8856 ($\mathrm{kJ\,mol^{-1}}$) | −187.8 : −241.818 ($\mathrm{kJ\,mol^{-1}}$) |
| Density | 958 ($\mathrm{kg\,m^{-3}}$) | 1376 ($\mathrm{kg\,m^{-3}}$) |

Table 9.1: Thermodynamic settings used in the computation of the combustion in CEA.

**Mission: Ground Launch 10 kg to Orbit** With the data at hand the first tests focus on exploring if using a comparable system one could reach a particular mission target. The experiment was then configured the following way:

1. Orbital Semi-Major Axis: $E_R$ + 780 km
2. Eccentricity under 0.01
3. Equatorial Eastward Ground Launch
4. Carry a 10 kg payload to orbit
5. Use the HTPB-$H_2O_2$ O/F pair
6. Set the chamber pressure to vary in the neighbourhood of 35 bar (Nammo's set-up)

Since Nammo's technology is, at this point, simply the engine, a set-up for the oxidizer cannot be completely emulated. Since it is pressure-fed $H_2O_2$, either a pump or a pressurant tank can be used. For this reason both scenarios were simulated for a ground-launch. For the pressurant tank, two pressurant gases were considered: Nitrogen and Helium at 150 bar. However, neither yielded a valid rocket that could reach the target. Only the pump-based rocket was successful.

**Results** The values for the optimized configuration for this case are compiled in Table 9.2. The overall characteristics of this rocket are summarized in Table 9.3. The results show that a rocket is feasible using the engine and its specifications. However, it relies heavily on the fact that the pumping system is quite light. The relations presented in 4.3.2 yield values for the pump mass of under 1 kg. This is deemed unrealistic. A minimum pump mass of 10 kg was set as a boundary for all simulations here. The 10 kg value corresponds to the lightest turbo-pump found

| Parameter | Optimized Using a Pump System |
|---|---|
| Chamber Pressure Stage 1 | 26.4171 (bar) |
| Exit Pressure Stage 1 | 0.516606 (bar) |
| Diameter Motor Case Stage 1 | 0.796852 (m) |
| Diameter Nozzle End Stage 1 | 0.558447 (m) |
| Burn Time Stage 1 | 76.1738 (s) |
| Coasting Time Stage 1 | 12.7548 (s) |
| Chamber Pressure Stage 2 | 28.5353 (bar) |
| Exit Pressure Stage 2 | 0.353367 (bar) |
| Diameter Motor Case Stage 2 | 0.791235 (m) |
| Diameter Nozzle End Stage 2 | 0.249511 (m) |
| Burn Time Stage 2 | 97.2367 (s) |
| Coasting Time Stage 2 | 443.345 (s) |
| Chamber Pressure Stage 3 | 22.2188 (bar) |
| Exit Pressure Stage 3 | 0.170968 (bar) |
| Diameter Motor Case Stage 3 | 0.764231 (m) |
| Diameter Nozzle End Stage 3 | 0.15553 (m) |
| Burn Time Stage 3 | 58.9192 (s) |
| Coasting Time Stage 3 | 0 (s) |

Table 9.2: Rocket configuration for a 3-stage launch vehicle, based on the technology by Nammo.

| GTOW | Diameter |
|---|---|
| 3982.44 (kg) | 0.796852 m |

Table 9.3: Overall characteristics of the rocket resulting form the settings in Table 9.2.

in literature, in this case for the XLR132 rocket engine [58]. For comparison, 10 kg is also the considered payload mass meaning that 10 kg is by no means an insignificant value and it definitely influences the overall mass of the rocket. An overview of the evolution of the rocket parameters during the launch simulation is give in Figure 9.1.

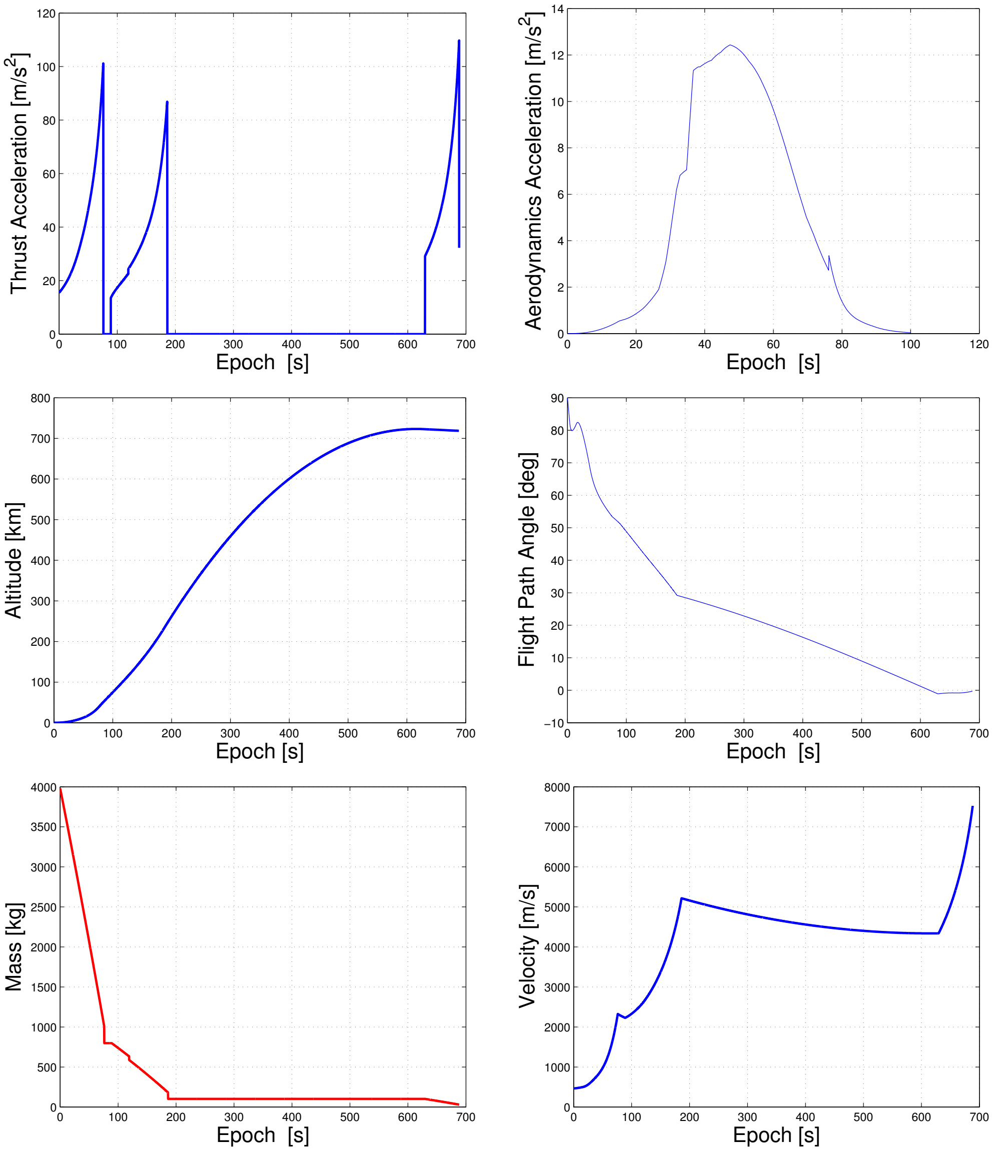


Figure 9.1: Ground Launch, 10kg Payload.

## Mission: Air Launch 10 kg to Orbit

1. Orbital Semi-Major Axis: $E_R$ + 780 km
2. Eccentricity under 0.01
3. Equatorial Eastward Air Launch
4. Air-Launch at 15 km altitude and 250 m s$^{-1}$ carrier aircraft velocity
5. Carry a 10 kg payload to orbit
6. Use the HTPB-$H_2O_2$ O/F pair
7. Set the chamber pressure to vary between 10 and 50 bar (close to Nammo's set-up of 35 bar)

From [4] there was the perspective that an optimized air-launch would yield a lighter vehicle than its counter-part ground launch. Thus the equivalent assignment was given to the optimizer and the results are presented in Table 9.4

**Results** The results are compiled in Table 9.4. The overall characteristics of this rocket are summarized in Table 9.5. The evolution of the rocket parameters during launch simulation are illustrated in Figure 9.2.

| Parameter | Optimized Using a Pump System |
|---|---|
| Chamber Pressure Stage 1 | 27.4159 (bar) |
| Exit Pressure Stage 1 | 0.274324 (bar) |
| Diameter Motor Case Stage 1 | 0.606669 (m) |
| Diameter Nozzle End Stage 1 | 0.387106 (m) |
| Burn Time Stage 1 | 91.3578 (s) |
| Coasting Time Stage 1 | 15.4944 (s) |
| Chamber Pressure Stage 2 | 27.086 (bar) |
| Exit Pressure Stage 2 | 0.182537 (bar) |
| Diameter Motor Case Stage 2 | 0.41532 (m) |
| Diameter Nozzle End Stage 2 | 0.169327 (m) |
| Burn Time Stage 2 | 78.0953 (s) |
| Coasting Time Stage 2 | 372.345 (s) |
| Chamber Pressure Stage 3 | 28.0207 (bar) |
| Exit Pressure Stage 3 | 0.106187 (bar) |
| Diameter Motor Case Stage 3 | 0.286988 (m) |
| Diameter Nozzle End Stage 3 | 0.146805 (m) |
| Burn Time Stage 3 | 56.6791 (s) |
| Coasting Time Stage 3 | 0 (s) |

Table 9.4: Table with the results for the Nammo experiments.

**Comparison with the ground launch** The saving in the total mass of the rocket is quite high. GTOW for the Air-Launch case is 34.6% of the GTOW for the ground launch case for the same mission. This correlates well with expectation and is aligned with the findings in [4] which finds a GTOW of 930.6 kg for an Air-Launch scenario and 3087 kg for a ground launch one. Therefore the air launched vehicle in [4] is 30.1% of the weight of its ground launched counter part. All the results from [4] mentioned here are for optimized GTOW. The results as optimized for GTOW from [4] do not always correspond with those that were optimized for cost. The differences can be found in the cost models that van Kesteren used a discussion of which is not made here since no cost models were used for this study. Since a general correlation between GTOW and mass was proved for solid rockets in [4] it is assumed here that, to the extent of this project, the correlation holds. However, to reduce bias in the comparisons, one must still used the results as optimized for GTOW as the particulars of each solution that cause some lighter solutions to be more expensive are also not analysed in detail here and would lack motivation if used. As a future recommendation an investigation on these particular cases and on a detailed cost model could be done for hybrid rockets as this would be interesting to explore as part of the technology's "selling point".

### 9.1.2. Engine based on the DARE Aurora Engine

In this section the engine used is referred to by the DARE engine or Aurora but it is in fact the best approximation possible from the available data. Guessed values like $I_{sp}$ are considered conservative thus the final result will be so as well. Similarly to the case of Nammo, the rocket engine by DARE can undergo the same analysis and experiment. DARE uses a Nitrous Oxide-Sorbitol rocket engine. As in the Nammo case, a file with the thermodynamic data for

| GTOW | Diameter |
|---|---|
| 1378.47 (kg) | 0.60669 m |

Table 9.5: Overall characteristics of the rocket resulting form the settings in Table 9.4.

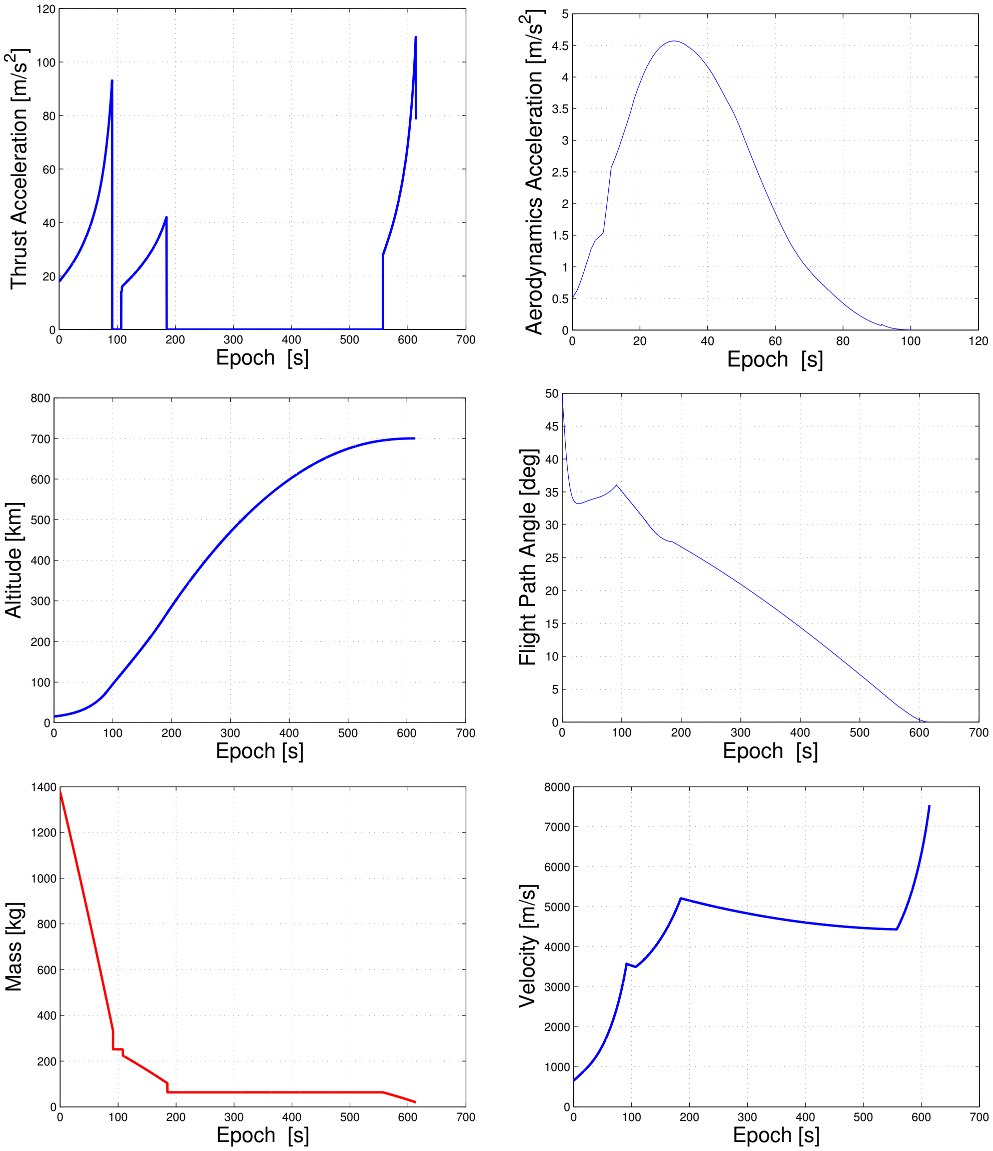


Figure 9.2: Air Launch, 10kg Payload.

the fuel-oxidizer pair's combustion was created. The pressure range was also 30 - 70 bar with a 1 bar resolution and the O/F range between 2 and 6 with intervals of 1. Stratos uses a self-pressurized oxidizer tank system so the modelling was done according to specifications. The Oxidizer Tank was set operate at 60 bar meaning the chamber pressure could only be lower (pressure drop in feed lines is accounted for). The settings for the fuel computation are presented in Table 9.6.

| Setting in CEA | Fuel - Sorbitol | Oxidizer - $N_2O$ |
|---|---|---|
| Formula | $C_6H_{14}O_6$ | $N_2O$ |
| Heat of Formation | -1353.7 ($\mathrm{kJ\,mol^{-1}}$) | 82.05 ($\mathrm{kJ\,mol^{-1}}$) |
| Density | 1489 $\mathrm{kg\,m^{-3}}$ | 729.20 $\mathrm{kg\,m^{-3}}$ |

Table 9.6: Thermodynamic settings used in the computation of the combustion in CEA.

**Mission: Ground Launch 10 kg to Orbit** Here mission conditions are the same as for the previous vehicles. Instead of the Nammo based engine, this configuration uses the DARE Aurora Engine that uses Sorbitol and Nitrous Oxide. The oxidizer tank was set to 70 bar pressure, which is considered high for a feed system. All the chamber pressures were only allowed to vary under this value.

**Results** The results are compiled in Table 9.7. Evolution of the rocket parameters during launch are shown in Figure 9.3 The overall characteristics of this rocket are summarized in Table 9.8. These results are considered unrealistic. This mass is quite low and on par with what one would expect from an optimized air launch. In [6], the air-launched optimized rocket achieves a mass of approximately 1300 kg. This ground launched rocket is too close to this mass. It is my opinion that the inert mass corrections, unavailable for Hybrid Rocket Vehicles in the form of historical data is too generous (set to about 1.2 in these settings based on solid rocket numbers from [4]). In the future, it is my recommendation that this is investigated to motivate and use a proper correction factor.

| Parameter | Optimized Vehicle |
|---|---|
| Chamber Pressure Stage 1 | 64.4731 (bar) |
| Exit Pressure Stage 1 | 0.592998 (bar) |
| Diameter Motor Case Stage 1 | 0.424857 (m) |
| Diameter Nozzle End Stage 1 | 0.261943 (m) |
| Burn Time Stage 1 | 75.4546 (s) |
| Coasting Time Stage 1 | 17.6405 (s) |
| Chamber Pressure Stage 2 | 34.2415 (bar) |
| Exit Pressure Stage 2 | 0.123208 (bar) |
| Diameter Motor Case Stage 2 | 0.300605 (m) |
| Diameter Nozzle End Stage 2 | 0.237396 (m) |
| Burn Time Stage 2 | 79.4571 (s) |
| Coasting Time Stage 2 | 377.584 (s) |
| Chamber Pressure Stage 3 | 41.1167 (bar) |
| Exit Pressure Stage 3 | 0.0808019 (bar) |
| Diameter Motor Case Stage 3 | 0.281286 (m) |
| Diameter Nozzle End Stage 3 | 0.134838 (m) |
| Burn Time Stage 3 | 75.9164 (s) |
| Coasting Time Stage 3 | 0 (s) |

Table 9.7: Table with the results for rockets based on DARE's Aurora Engine.

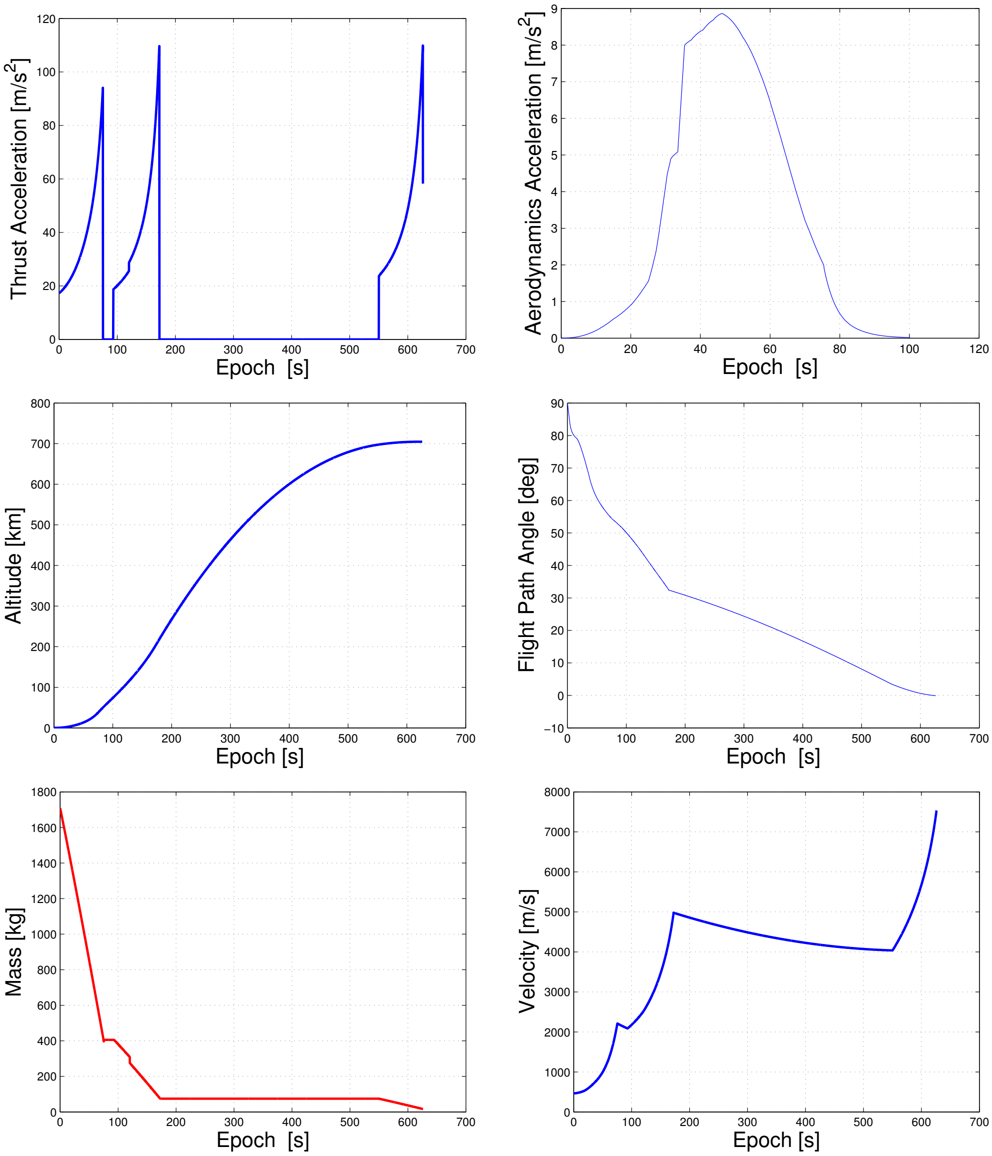


Figure 9.3: Ground Launch, 10kg Payload.

| GTOW | Diameter |
|---|---|
| 1709.89 (kg) | 0.424857 m |

Table 9.8: Overall characteristics of the rocket resulting form the settings in Table 9.7.

## 9.2. Adding Solid Stages to Hybrid Rockets

Since one of the upgrades to the tool was the addition of the possibility of mixed rockets, i.e. rockets that have its stages using different propulsion technology, some tests were made for this case. Since the model from [4] was based on Solid Rocket technology, the investigation focused on using Solid and Hybrid Stages of the same vehicle. This investigation seemed promising because Hybrid technology has been pointed out as a promising solution for the final stage of rocket launch vehicles [33]. Its throttling capabilities, allow for fine orbit injection, and the ability to re-start a stage, which is impossible with solid rocket stages, brings the possibility of complex injection schemes. For instance, one could fire the engine on a first instance, cut-off thrust, wait for an advantageous orbital position and then perform a manoeuvre such as an apogee raise.

### 9.2.1. Mission and Results

For a sample case that fitted in this research, air-launching a solid-hybrid rocket vehicle seemed the most interesting scenario to investigate. The characteristics are the following:

1. Air Launch Eastward at the Equator
2. 10 kg payload to orbit
3. First two stages are **Solid** and the final stage is **Hybrid**
4. Solid stages are HTPB monopropellant, based on the model in [4]
5. Hybrid Stages are Hydrogen Peroxide - HTPB based on the model by Nammo [12].

After the optimization process, the tool yielded the Rocket Vehicle configuration in Table 9.9. The overall characteristics of this rocket are summarized in Table 9.10. The evolution of the rocket parameters during launch simulation are illustrated in Figure 9.4 A decrease in GTOW when compred with the Fully Hybrid Solution can be observed. GTOW for the mixed case is 34% less. This is justified by the lower inert mass of the Solid Rocket technology. This is a very positive result as it indicates feasibility of a mixed rocket concept. For comparison, in [4] the GTOW for a pure 3-stage solid rocket is 930.6 kg (when optimized for cost and GTOW). Since this is a novel experiment it is relevant to analyse the thrust profile for this case, here shown in Figure 9.5. Figure 9.5 illustrates the expected operation mode for such a rocket. Thrust Force is constant in the Solid Rocket Model used here. During the firing of

| Parameter | Optimized Vehicle |
|---|---|
| Chamber Pressure Stage 1 | 22.7657 (bar) |
| Exit Pressure Stage 1 | 0.188601 (bar) |
| Diameter Motor Case Stage 1 | 0.457818 (m) |
| Diameter Nozzle End Stage 1 | 0.358947 (m) |
| Burn Time Stage 1 | 79.308 (s) |
| Coasting Time Stage 1 | 11.226 (s) |
| Chamber Pressure Stage 2 | 25.6637 (bar) |
| Exit Pressure Stage 2 | 0.195897 (bar) |
| Diameter Motor Case Stage 2 | 0.399981 (m) |
| Diameter Nozzle End Stage 2 | 0.149063 (m) |
| Burn Time Stage 2 | 83.2812 (s) |
| Coasting Time Stage 2 | 423.111 (s) |
| Chamber Pressure Stage 3 | 27.7922 (bar) |
| Exit Pressure Stage 3 | 0.102966 (bar) |
| Diameter Motor Case Stage 3 | 0.29729 (m) |
| Diameter Nozzle End Stage 3 | 0.0.148946 (m) |
| Burn Time Stage 3 | 53.1629 (s) |
| Coasting Time Stage 3 | 0 (s) |

Table 9.9: Table with the results for a rocket whose two first stages are solid-based (base on the model from [4]) and the final one being based on Nammo's Hybrid Rocket technology (Hydrogen Peroxide - HTPB).

| GTOW | Diameter |
|---|---|
| 1025.22 (kg) | 0.457818 m |

Table 9.10: Overall characteristics of the rocket resulting form the settings in Table 9.9.

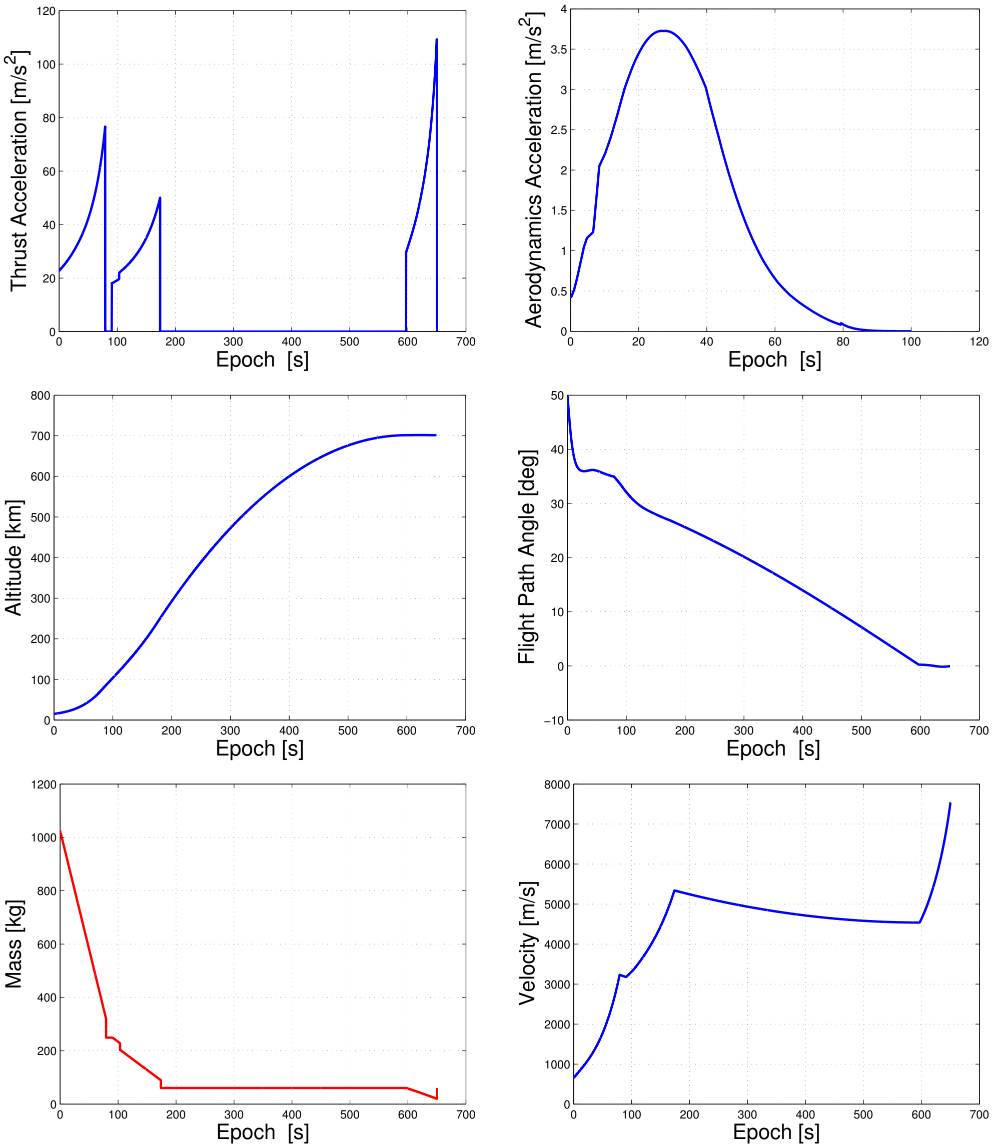


Figure 9.4: Ground Launch, 10kg Payload. First two Stages are Solid, the last one is Hybrid. Solid model from [4], Hybrid Stage based on the Nammo Engine.

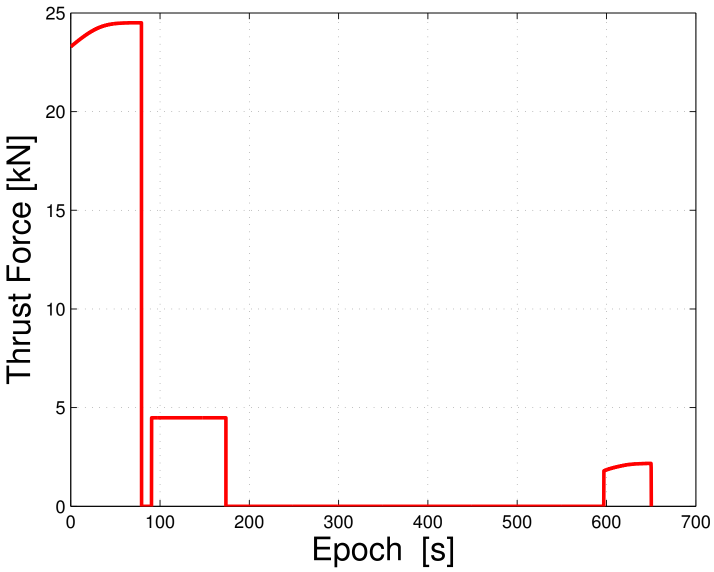


Figure 9.5: Thrust Force for the Solid-Hybrid Rocket.

the First Stage the variation is due to the variation in atmospheric pressure. When the Second Stage is operating, the pressure effect is already far lower due to the higher altitude and the constancy of the force is clearly observed. However, the final firing, of the Hybrid Stage, shows the expected varying thrust level caused by the increase in mass flow observed due to the change in the grain's dimensions and the O/F shift.

## 9.3. Launch Simulation for Known Rockets

As a proof-of-concept the results pertaining to the simulations of actual known rockets are presented in this section. This analysis intends to reveal how well a known rocket could perform if launched. This differs from the tests in Section 9.1 because there is no optimization involved, not even in the trajectory. For these simple tests, and since a sounding rocket has a relatively short travel time, pitch instructions will remain constant and vertical (90 degrees) throughout the entire burn. The rockets are specified to full-detail and performance curves, for mass flow and thrust, are provided. These results are possible from the data shared by Nammo and DARE.

**DARE** The DARE rocket is a Single Stage Hybrid Rocket vehicle. Its objective is to reach an altitude of 50 km. The thrust curve for one of their longest tests was provided to me personally by the project members. It is shown in Figure 9.6. The test illustrated in Figure 9.6 did not use the entire oxidizer which makes it a pessimistic curve to use. The team produced an expected curve by adapting the one on Figure 9.6, this was provided as a series of data points and was the one used for the testing in this report. The projected thrust curve is shown in Figure 9.7. In both Figures 9.6 and 9.7 a significant and sudden drop in the thrust is observed at approximately 8 and 21 seconds respectively. This corresponds to the end of the liquid phase of the oxidizer. After this point the mass flow is reduced quite significantly, which dramatically reduces thrust. Note that the difference between the two curves is, therefore, quite significant since the liquid phase is more than twice as long for the projected curve. The curve from Figure 9.7 is corrected for sea-level pressure since the program's architecture requires input thrust curves to be vacuum thrust. The following element to acquire would be the mass flow curve. This was not provided to me in the same detail as the thrust curve, it was then approximated from data from [15]. From [15], we find that the average, intended, oxidizer mass flow is about 4 kg. Assuming the optimum O/F ratio of 3 is used (it is the optimum one as observed in Figure 4.5 from the peak in $I_{sp}$), total mass flow should average 5.3 kg s$^{-1}$. By keeping a constant $I_{sp}$ we can then use average thrust, from the curve, with the average mass flow, to determine average $I_{sp}$. This technique proved imprecise since it yielded a mass flow that was too high, i.e. the propellant mass was exhausted in far less time than the total time of the thrust curve. The solution was to adjust a mass flow curve using an $I_{sp}$ that corresponded with exhausting all the propellant mass in that time. The computed final mass flow curve is shown in Figure 9.8, the total mean mass flow is 2.764 kg s$^{-1}$, lower than the indicated 5.3 kg s$^{-1}$, which are, however, closer to the mean liquid phase of 4.13 kg s$^{-1}$. With both curves all that remains is to set the required rocket settings. The settings are summarized in Table 9.11 along with the aforementioned parameters. Regarding the trajectory, the Pitch Angle was set to a fixed vertical 90 degrees.

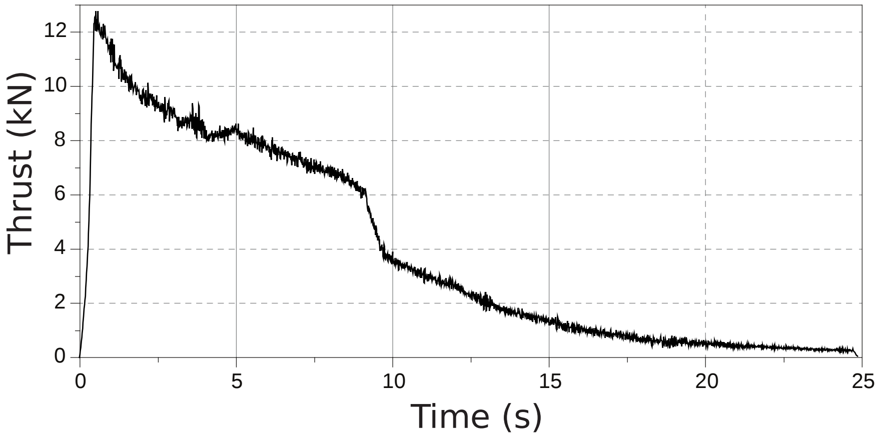


Figure 9.6: Thrust curve (sea-level) for the test firing of the Aurora engine, used in the Stratos II rocket. Data adapted from information provided personally by project members.

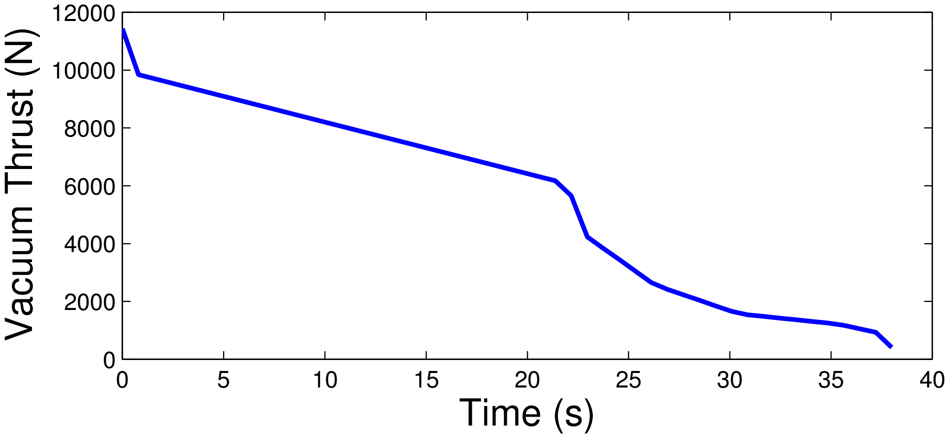


Figure 9.7: Projected vacuum thrust curve for the aurora engine once it is fired using all of the oxidzer.

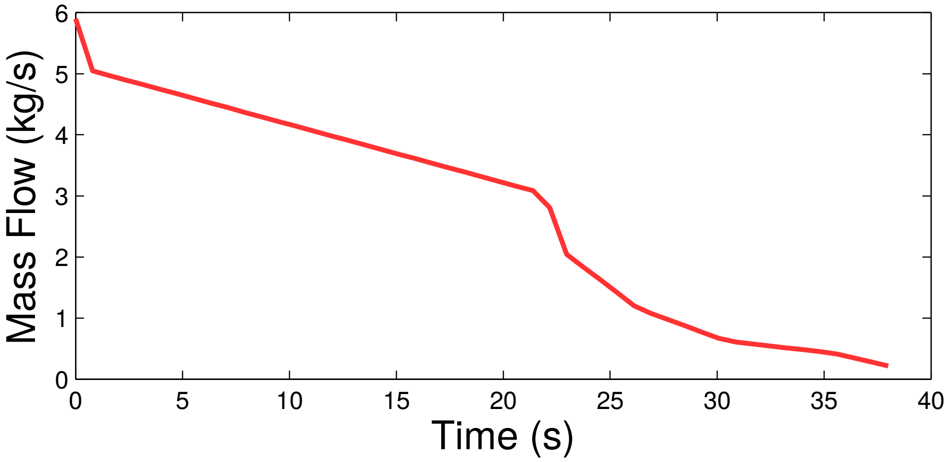
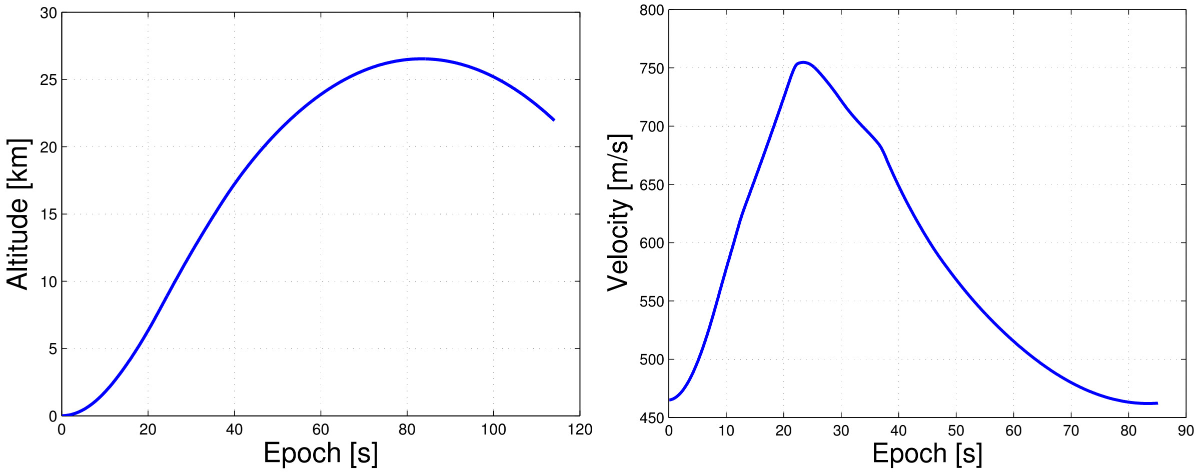


Figure 9.8: Projected vacuum thrust curve for the aurora engine once it is fired using all of the oxidzer.

| | |
|---|---|
| Inert Mass | 84.5 (kg) |
| Propellant Mass | 104.8 (kg) |
| Length | 7.5 (m) |
| Diameter Throat | 0.0564 (m) |
| Burn Time | 38 (s) |
| Diameter Case | 0.19 (m) |
| Diameter Nozzle End | 0.1438 (m) |
| Exit Pressure | 0.75 (bar) |

Table 9.11: Settings used to simulate the Stratos II Rocket.



(a) Altitude (m).

(b) Velocity ($\mathrm{m\,s^{-1}}$).

Figure 9.9: Estimated altitude and velocity evolution for the settings in Table 9.11. Velocity is in the inertial reference frame.

**Results** The simulator yields a considerable number of curves for the simulations, only the more relevant ones are presented here. As per the burn time, thrust was cut-off at 38 s. Maximum reached altitude was 26.53 km. This is lower than the projected 50 km. Possible explanations for this:

1. The mass flow curve was estimated using a lower than real $I_{sp}$ due to the artificial way the mass curve was constructed. For the **tests where *not* all the oxidizer was used**, [15] mentions an $I_{sp}$ of 177 s. An extensive study into this topic was not conducted due to lack of time and is left as a future recommendation.
2. The mass data used is based on loosely approximated figure provided in personal communication, could be incorrect. However, the idea was to prove this sort of study can be done with the software.

The evolution of the rocket parameters along flight are shown in Figure 9.10

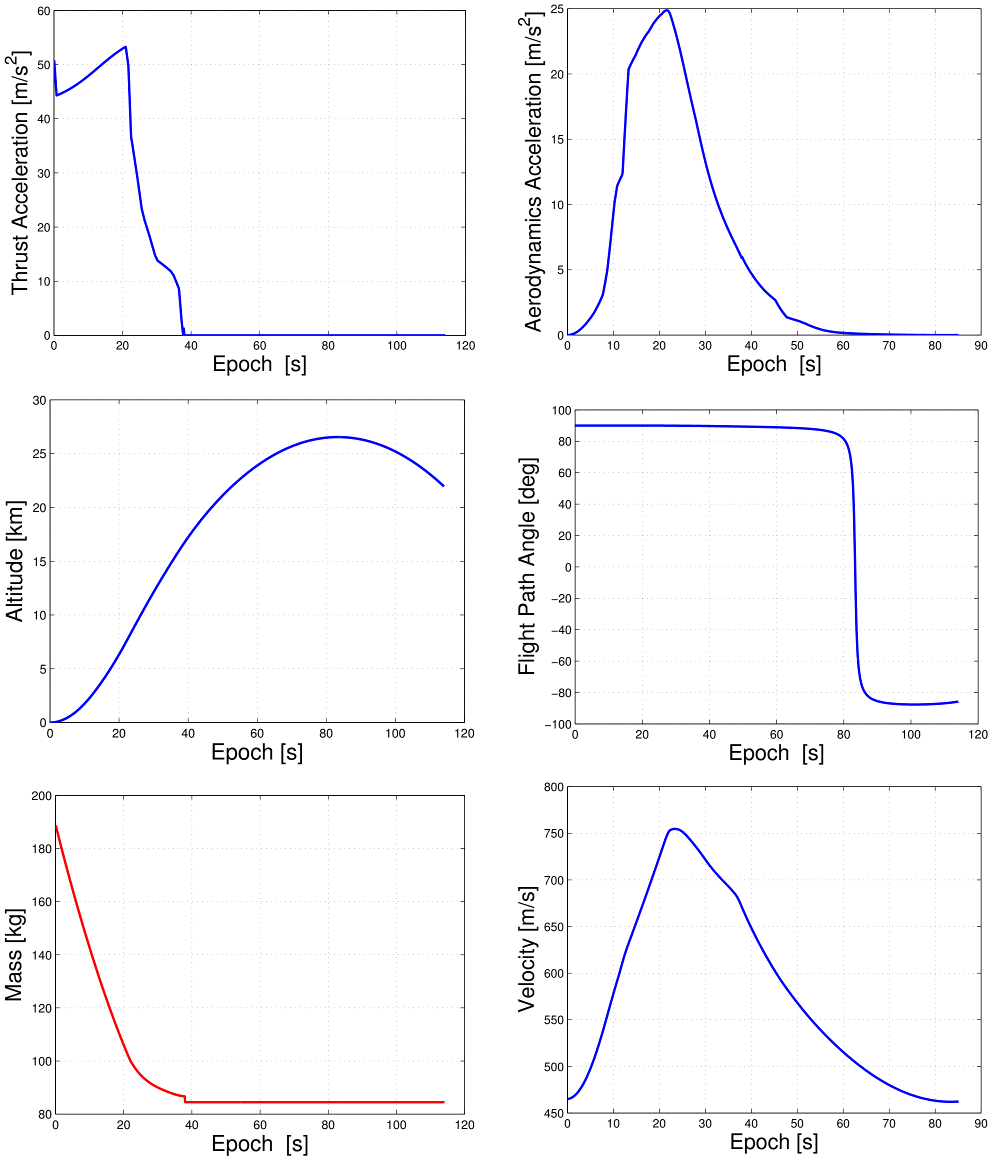


Figure 9.10: Ground Launch, 10kg Payload.

# 10

# Sensitivity Analysis

The aim of this chapter is to provide some insight into the sensitivity of the solutions presented in Chapter 9. Here the case of an Air-Launched Hybrid Rocket Vehicle with a 10 kg payload to a 780 km altitude circular orbit, is used as the reference. This corresponds to the case described in Table 9.4 in Chapter 9. This case was chosen because it represents the most complicated case tested apart from the Solid-Hybrid mixed rocket. The pure Hybrid case was chosen over that one, because Hybrid Rocket research is the main focus of this project.

## 10.1. One At A Time Approach

The sensitivity analyses presented in this study focus on the design parameters of each rocket stage:

1. Chamber Pressure
2. Exit Pressure
3. Chamber Diameter
4. Nozzle Exit Diameter
5. Burn Time
6. Coasting Time

The approach is to vary each of these variables one at a time and see how the result varies. This is called the One At a Time approach (OAT). Two characteristics will be checked:

1. Is the solution still valid after the variation?
2. How does the variation impact the fitness of a solution?

The fitness of the solutions is measured, in this project, in terms of GTOW. In order to test the sensitivity of a particular solution the program is fitted with a sensitivity analysis mode. A working solution is required for this mode to be used. The program takes the working solution - ideally an optimal one - and attempts to vary its parameters one by one until the outcome is an *invalid solution*. Once the solution becomes invalid the program records the last allowed value and proceeds to the next parameter. The reference case for this analysis is an Air-Launch of a Hydrogen Peroxide - HTPB rocket. This case was chosen because it is the most complex one and also the most relevant one for this research. When searching for the optimal outcome the program usually takes a + - 10 km window, for the sensitivity test a more permissive + - 25 km value is used. In the plots contained in this chapter, parameters are referred to by their number in the simulation. Table 10.1 shows the correspondence between parameter name and number. Using the parameter numbers in Table 10.1 as reference, Figure 10.1 shows how much each parameter can vary before the solution becomes invalid. In Figure 10.2 the effect on the final altitude is shown for each parameter. It corresponds to the variations showed in Figure 10.1, i.e. the variations shown for parameter **n** in Figure 10.1 cause the altitude variations seen in Figure 10.2 when parameter **n** was shifted. Note that the altitude shift is never more than 3.2 % either positive or negative. This is because **for this test the allowed variation was +/- 25 km from the target**, as opposed to the traditional +/- 10 km used **to find the optimum value**. Thus varying any parameter until a particular solution fails, corresponds to getting any one solution to *"worsen" until the +/- 25 km margin is reached instead of the 10 km one*. This is confirmed since 3.2% of 780 km, the optimum value, **is 25 km**. Parameters with smaller influence on the final altitude like Parameter 9, the diameter of the case for stage 2 and Parameter 18, the pitch angle immediately after launch, correspond to some of parameters with the largest allowed variation (Figure 10.1). This is coherent since, by not influencing the altitude as much, they also do *not influence solution validity that much*. This is also seen

| Parameter Name | Parameter Number | Parameter Name | Parameter Number |
|---|---|---|---|
| Chamber Pressure Stage 1 | 1 | Diameter Case Stage 3 | 15 |
| Exit Pressure Stage 1 | 2 | Diameter Nozzle End Stage 3 | 16 |
| Diameter Case Stage 1 | 3 | Burn Time Stage 3 | 17 |
| Diameter Nozzle End Stage 1 | 4 | Pitch Angle 1 | 18 |
| Burn Time Stage 1 | 5 | Pitch Angle 2 | 19 |
| Coasting Time Stage 1 | 6 | Pitch Angle 3 | 20 |
| Chamber Pressure Stage 2 | 7 | Pitch Angle 4 | 21 |
| Exit Pressure Stage 2 | 8 | Pitch Angle 5 | 22 |
| Diameter Case Stage 2 | 9 | Pitch Angle 6 | 23 |
| Diameter Nozzle End Stage 2 | 10 | Pitch Angle 7 | 24 |
| Burn Time Stage 2 | 11 | Pitch Angle 8 | 25 |
| Coasting Time Stage 2 | 12 | Pitch Angle 9 | 26 |
| Chamber Pressure Stage 3 | 13 | Pitch Angle 10 | 27 |
| Exit Pressure Stage 3 | 14 | Pitch Angle 11 | 28 |

Table 10.1: Parameter names and their respective numbers as used in the plots throughout this chapter.

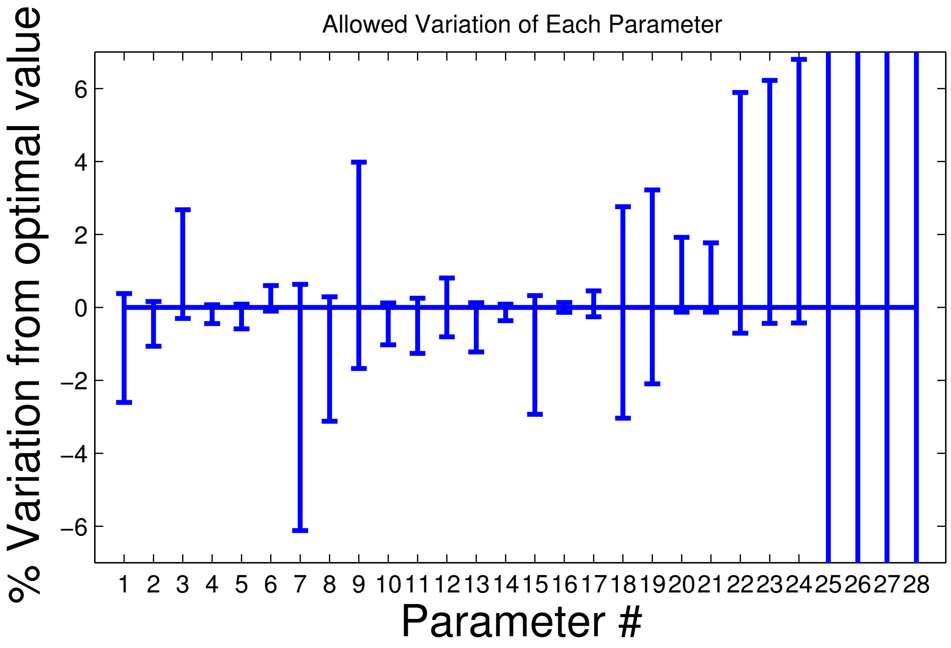


Figure 10.1: Allowed relative variation as a percentage of the optimal value of each parameter in its own units. Figure is cropped at + - 7%

for the last three parameters, the final pitch angles which have a much smaller influence on the outcome than the initial ones, their influence is so low that their allowed variation is acutally greater than 10% in the case of the last two, this is not shown in Figure 10.1 since it was cropped for clarity. Figure 10.3 shows the effect that the

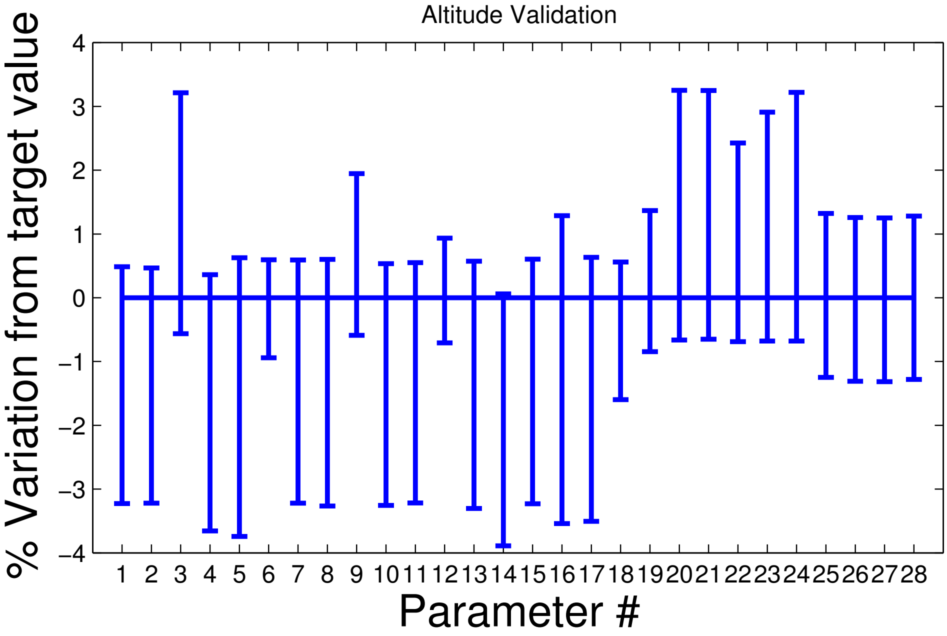


Figure 10.2: Relative variation in the final altitude for the allowable variation of each parameter. The variation is indicated as a percentage of the final altitude for the optimal case used here. Figure is cropped at + - 4%.

same shifts have on GTOW. Since GTOW is a direct measure of fitness in this study, this this figure is considerably important. An increase in GTOW means an increase in cost - assuming the direct proportionality relation between GTOW and cost for solids established in [4] also holds here. Unsurprisingly, changes to the first stage cause the largest change in GTOW. Since all the stage parameters affect the GTOW (except for coasting time) the larger the mass of a particular stage, the larger the overall impact will be on the whole vehicle. Figure 10.3 clearly shows this decay - the influence of each stage is smaller than that of the previous one. Globally, changing the diameter of the nozzle's exit seems to be the most impacting change in GTOW, followed by the chamber pressure, except for the case of the first stage, where the burn time is more influential. The reason for this is that both factors control the amount of thrust generated, this is easily predicted since in Equations 7.1 and 7.3 in Chapter 7 the dependence between the different parameters in quite explicit. The way positive and negative variations from the optimal solution affect the launch outcome is different. Reductions in all parameters except the case diameter and the burn and coasting times cause a direct reduction in thrust. This either yields an invalid vehicle or not enough energy to reach the target altitude. An increase in these values, yields, in most cases, a violation of the acceleration constraint for the first stage (maximum is set at 110 $\mathrm{m\,s^{-2}}$) and causes the vehicle to not reach target when done in the final two stages. The only parameter whose increase does not cause a violation of the physical constraints is the burn time. A burn time decrease causes insufficient fuel to be sized for the stage, an increase may cause two negative effects: too much fuel (mass) on board, or a burn timing that is incompatible with the pre-computed trajectory settings. Interestingly enough, changing the Diameter of the Motor Cases seems to have little impact on overall GTOW. This is due to the fact that the program automatically compensates the increase in Diameter with a reduction in Length, thus the overall mass is more or less maintained.

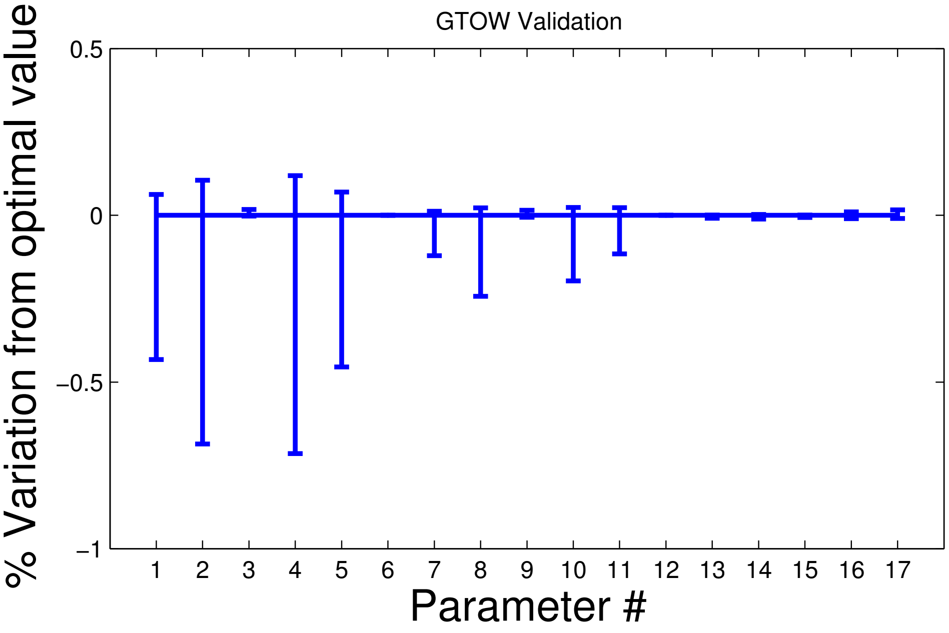


Figure 10.3: Relative variation in the final altitude for the allowable variation of each parameter. The variation is indicated as a percentage of the final altitude for the optimal case used here. Figure is cropped at + - 4%. Parameters 17-28 are not shown since changing the pitch angles has obviously no effect on the GTOW.

## 10.2. Solution Availability

As seen in the previous section, a 6% variation on most parameters causes the solution to fail (all but three last three pitch angles as seen in Figure 10.1). However, this does not mean that there is no flexibility in the solution. In fact, most variations could be compensated by varying other parameters due to non-linear effects of the interaction of one parameter with the others. This is can be seen in the optimization figures for individual parameters. By observing the evolution of the value of each parameter along each generation of solutions and filtering out the invalid ones, the space where solutions may exist is clearly visible. A proper monte carlo search that would vary some or all the parameters simultaneously at random would be the traditional way of searching for solution availability. However, this approach is computationally intensive and the data that was saved during the FIPSO search has enough breadth to also produce a useful figure. Figure 10.4 shows the evolution of the relative error for all the viable candidates ordered by value of the fitness function **for the evolution of the Chamber Pressure of Stage 1**. The left of the plot has those with the smaller (less negative) fitness value and the right those with the highest. Here fitness that is *more negative* is better, since the optimizer works as a *minimizer*. As expected the worse fitness values allow for a broader range of solutions. As the fitness improves the particles converge and show the range of solutions narrowing. Since along the generation of each solution, all the other candidates that are also viable are saved to a file it is possible to see where in the interval are solutions located. Parameters that are only located in a small part of the interval are **less flexible** and parameters that can be set to a large amount of its search interval are **more flexible**.

**Search Intervals** The search intervals for this test are presented in Table 10.2. For instance taking Figure 10.4 we can observe that the maximum possible value for the chamber pressure is +10% of the optimum and the minimum is -20% giving a bandwidth of 30% in total, this corresponds to approximately 8.25 bar: **33 % of the allowed search interval for this parameter**. Repeating this process for all parameters we can observe how much of each interval actually contains viable solutions. This is illustrated in Figure 10.5. While some parameters are quite flexible and have solutions scattered between 30 and 40 % from its optimal value, most parameters are only found to have a 20% allowable range. Unlike the OAT approach, where one could argue that a variation on a parameter could be compensated by changing another one, here that is not the case. **All the viable solutions are already presented here**, as such, if a particular parameter strays outside of its availability interval there is no **possibility for a solution to be exist, no matter how much the other parameters are changed**. The case of parameter 12 - the coasting time for the second stage - is the most interesting one. Only 3.48% of its interval presents viable solutions. However, its optimized value is 372.345 s, making 3.48% to be 19.14 (s). For comparison, the coasting time of stage 1, parameter 6, has an allowed variation of approximately 50% but there it corresponds to approximately **only 7.7 s**. Does even though the *relative sensitivity* of parameter 12 is much

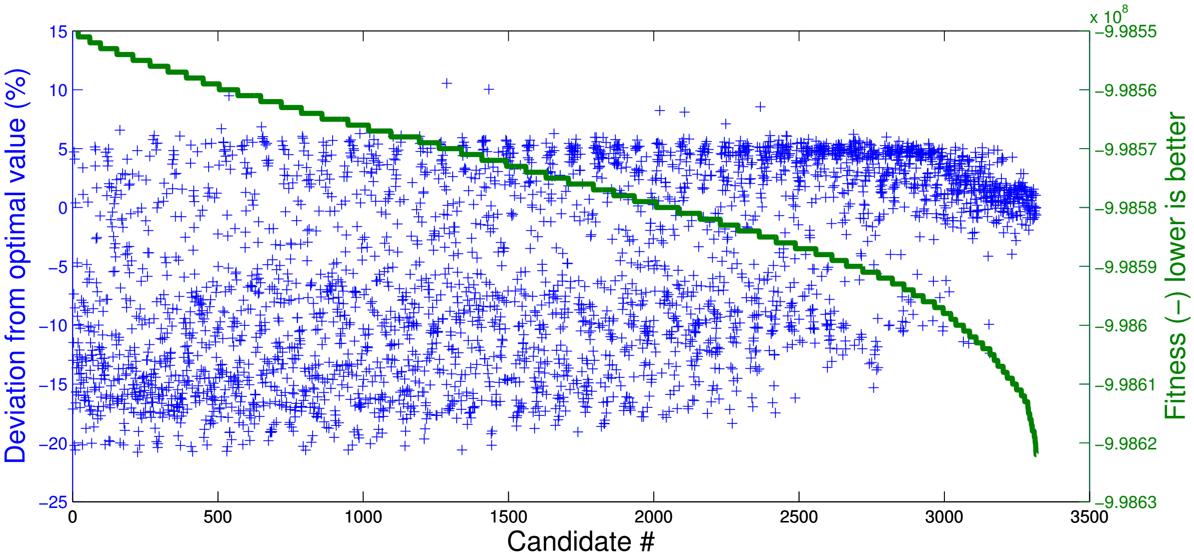


Figure 10.4: Solution availability for the Chamber Pressure of Stage 1 for the space in which the optimizer looked in. Fitness is measured on the right-side y-axis, deviation from the optimal value (27.419 bar) is shown in percentage on the left side.

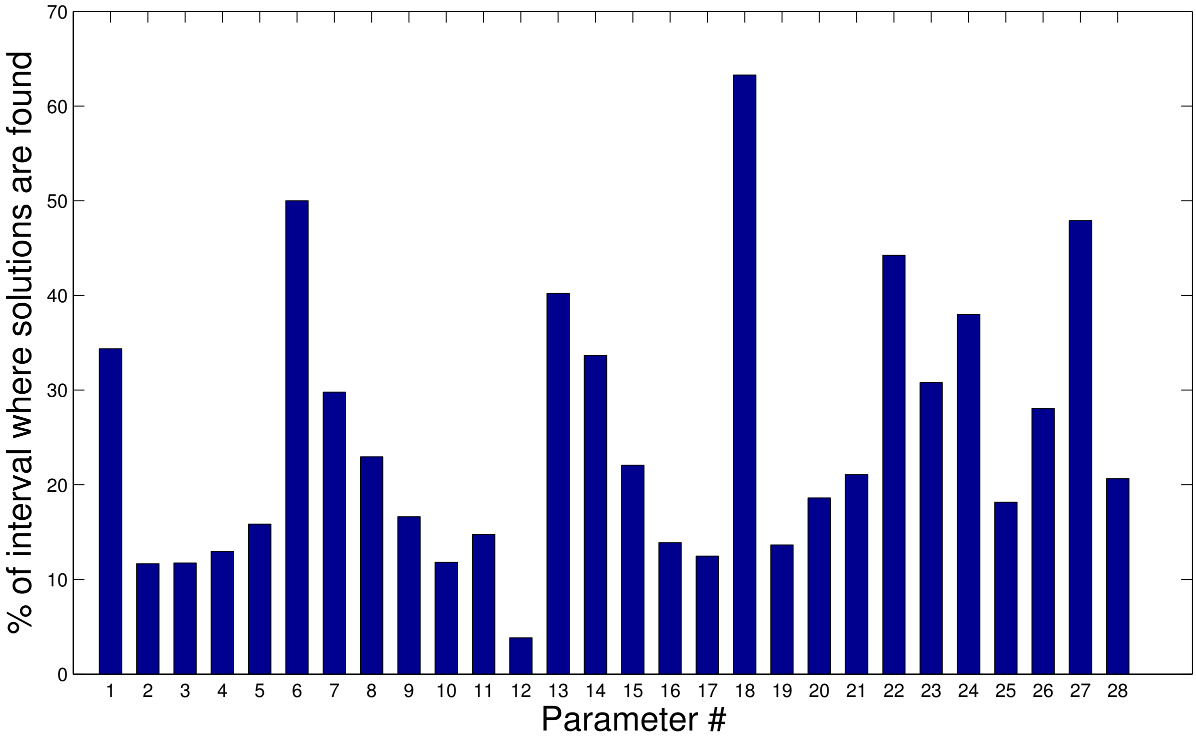


Figure 10.5: Percentage of each interval that contains viable solutions. Parameters 1-17 are the Rocket Parameters, parameters 18-28 are Trajectory Parameters. **Refer to Table 10.1 for the meaning of the parameter numbers.**

| Rocket Parameters | | Trajectory Parameters | |
|---|---|---|---|
| Chamber Pressure Stage 1 | [10, 35] (bar) | Pitch Angle Point 1 | [30, 55] (deg) |
| Exit Pressure Stage 1 | [0.05, 0.4] (bar) | Pitch Angle Point 2 | [30,60] (deg) |
| Diameter Motor Case Stage 1 | [0.3, 0.7] (m) | Pitch Angle Point 3 | [25, 55] (deg) |
| Diameter Nozzle End Stage 1 | [0.1, 0.5] (m) | Pitch Angle Point 4 | [25, 55] (deg) |
| Burn Time Stage 1 | [50, 110] (s) | Pitch Angle Point 5 | [15,45] (deg) |
| Coasting Time Stage 1 | [0, 20] (s) | Pitch Angle Point 6 | [10, 40] (deg) |
| Chamber Pressure Stage 2 | [10, 40] (bar) | Pitch Angle Point 7 | [10, 40] (deg) |
| Exit Pressure Stage 2 | [0.05, 0.45] (bar) | Pitch Angle Point 8 | [0, 20] (deg) |
| Diameter Motor Case Stage 2 | [0.25, 0.6] (bar) | Pitch Angle Point 9 | [0, 20] (deg) |
| Diameter Nozzle End Stage 2 | [0.1, 0.27] (m) | Pitch Angle Point 10 | [0, 15] (deg) |
| Burn Time Stage 2 | [50, 100] (s) | Pitch Angle Point 11 | [0, 10] (deg) |
| Coasting Time Stage 2 | [100, 650] (s) | | |
| Chamber Pressure Stage 3 | [10, 50] (bar) | | |
| Exit Pressure Stage 3 | [0.01, 0.21] (bar) | | |
| Diameter Motor Case Stage 3 | [0.1, 0.5] (m) | | |
| Diameter Nozzle End Stage 3 | [0.05, 0.30] (m) | | |
| Burn Time Stage 3 | [20, 60] (s) | | |

Table 10.2: Search intervals for each parameter. Parameters 1-17 are designated the Rocket Parameters. Parameters 18-28 are designated Trajectory Parameters.

higher than that of parameter 6, the *absolute sensitivity* is actually lower. This makes parameter 12 one of the most sensitive in the analysis in a relative sense, but, for instance, if the coasting time is controlled by an electronic system the accuracy of this system and it's actuating mechanisms could in fact be *lower* for the coasting time of the second stage than for the first one. This shows that these comparisons always have to be taken into account using appropriate context. The plots were made using relative deviations to allow for a global look on the problem, when looking at the engineering of such a vehicle though, one must look at each parameter in detail. As an example of a relatively insensitive parameter we have parameter 18. Parameter 18 can exist in 63.3 % of its interval. Since the interval is 25 degree in range, this means the parameter could vary 15.825 deg, or 26.93% of the optimum value and still be part of a viable solution. The comparison between these two parameters gives an idea on how the different parameters contribute to the outcome. The design's outcome is therefore, much more sensitive to the coasting time of stage 2, parameter 12, than to the Pitch Angle Point 1 (parameter 18).

# 11 Conclusions and Future Recommendations

This chapter describes the conclusions drawn from the results and leaves some recommendations for future work. Firstly with the Conclusions in Section 11.1 and afterwards the Future Recommendations in Section 11.2.

## 11.1. Conclusions

The conclusions begin with understanding whether or not the research question was answered satisfactorily. The research question was the following:

> How do Hybrid Rockets perform when compared to competing technologies in both Air and Ground Launched scenarios?

The answer is: Hybrid Rockets perform on-par with comparable Solid technology. The research indicates that with some improvement Hybrid Technology might very well surpass the solid one. Furthermore, by just being at the same level, Hybrid Technology can already be considered a better alternative due to its practical advantages (see 4.1). The quantification of this comparison is done in the Chapter 9 and summarized in the remaining sections of this chapter. After the research questions one must analyse whether the proposed goals were all achieved, this is summarized in Table 11.1. All the goals that were proposed both for the thesis and initially as part of the Literature Study were completed. Individual conclusions for the main topics of this project are presented in the following subsections.

| Goal | Completion Level |
|---|---|
| Support at least 1 hybrid fuel-oxidizer combination | ✓- 2 supported |
| Support at least 1 grain regression law | ✓- Concept proved for $N_2O$-Sorbitol |
| Simulate combustion of a hybrid rocket | ✓- All performance cuves ($I_{sp}$, Thrust, $m$) were obtained |
| Support design (Mass & Geometry) of Solid Fuel and Oxidizer Tank elements | ✓- Both were achieved |
| Support the choice and design of pumping feed systems and pressurized solutions | ✓- Both were achieved |
| Nozzle and Fairing dimensioning based on [4] | ✓- Both were achieved |
| Improve User Experience of the Tool: remove the need to recompile the program for every new case | ✓- Completely successful |
| Upgrade tool's back-end to current versions of the libraries | ✓- Completely successful |
| Facilitate further upgrade of the tool | ✓- Code has extensive commenting, simpler structure, folders were cleaned and simplified |
| Create work worth of publishing in international journal or conference | ✓- Current abstract was submitted to IAC with positive comments from the members of the Space Department at the TU Delft |

Table 11.1: Summary of the goals from Chapter 1 with their completion level and some comments. IAC is the International Astronautical Congress.

### 11.1.1. Hybrid Rocket Technology

Simulation of Hybrid Rockets is considered to be achieved for this project's goals. The mass and propulsion estimators achieve to an acceptable precision the goal of replicating the underlying characteristics for a set of design parameters. For the two rocket types simulated the results are as expected. Both approximate the weights from the results of the solid rocket from [4] but with a larger mass comparatively. This is a result of the higher inert mass present in Hybrid Rockets. Another interesting thing is to notice that the concept of a pressurant tank seems to effectively outweigh the pump system, at least for the tested cases. This was expected from the Literature Study and confirmed here. For Hybrid Rockets it seems clear that either a system where the vapour pressure is enough to power the oxidizer through the feed system or a pump is used. Pressurant tanks might be viable for much larger rockets, but the focus here was to investigate the opportunities for the small payloads (< 10 kg) and, for these, there is a clear indication that pressurant tanks are not viable. It is worth noting that both the rockets tested used fuel-oxidizer pairs that are not the ones with the best performance. However, I consider that a comparison of specific pairs in order to prove that a particular pair where the performance is better, is not proof of the technology's superiority over its alternatives. It is, though, proven that Hybrid Rocket can achieve comparable performance to Solid Rockets. Since there are a number of Hybrid Rocket fuels with specific impulses over 300 s, [33], and Solid Rockets have, in general, a lower specific impulse, there is clear evidence that the technology is at the same level of Solid rockets. Even if the extra inert mass resulting from the presence of feed system components makes the rocket heavier, this can be compensated with propellants of higher performance. Adding to this, the advantages discussed in 4.1 should make Hybrid technology take the lead for the cases where the performance is the same.

### 11.1.2. On Air-launching

In the comparisons between ground and air-launches the results are also unequivocal. Air-launching a comparable rocket, to a comparable target, with the same payload results in a save in total mass of approximately 73.5%. The comparison in cost was not made here but there is a correlation between GTOW and Cost established in [4] for Solid Rockets. This indicates that in a point where both technologies would have the same maturity, heavier rockets in general would be more expensive. Thus, Air-launching retains its property of reducing the cost of a launch for Hybrid Rocket technology as well, at least from a vehicle cost perspective. The historical data for more detailed comparisons is unavailable and thus not done here. However, the general trend indicating a save in cost is supported by this research.

### 11.1.3. On Hybrid Rocket Research

The research definitely supports that Hybrid Rockets have the potential to surpass Solid technology in many aspects. An optimized rocket for a comparable technology achieved a GTOW of 1378.47 kg against a 930 kg GTOW for a Solid Rocket in [4]. Although the difference is still considerable, more efficient fuel-oxidizer pairs exist, and and hybrid rockets are currently estimated quite conservatively whereas solid rocket are not because of the higher level of maturity of the technology. The results for the Rocket's based on the Nammo Engine are considered satisfactory and consistent with literature, especially with [4] which modelled Solid Rockets using the previous versions of this tool. In [6], where the researchers take a very similar approach to obtain the optimized Hybrid Rocket Vehicle, a GTOW of 1244.91 kg is found for the optimum solution. All of this seems to support the consistency and quality of the results. For the Aurora Engine of the Stratos Rocket, the results are less satisfying, with the yielded GTOW being too low to be realistic leading to believe that some further investigation is required to get the modelling done properly. A harsher correction for the inert mass fraction is perhaps part of the solution. However, this does not affect the very interesting results obtained by simulating a launch of the Stratos II Rocket. This simulation was done using the Custom Stage Module, which takes all its parameters manually. Using the data provided on the rocket, the model yields that the maximum altitude reached by the rocket will be in the order of 25 km, lower than the projected 50 km. Explanations for this discrepancy are discussed with the Results in Chapter 9.

### 11.1.4. On the Tool

The development of the tool is considered to be quite successful. A number of useful features were added to the tool along with some polish on the code and increase in the ease of use. This work turned scientific code into a proper executable software that can be used to reproduce this results and do further modelling in the future.

## 11.2. Future Recommendations

This section outlines some of the more important recommendations for future work, not just on Hybrid Modelling but on the software itself as well.

### 11.2.1. Recommendations on Improving the Software

The software has been improved from last version in many aspects, but, due to time constraints, there are some undone changes highly recommended for future work:

1. Use parallel processing for simulating the launches - even though the optimization process itself is not easy to turn into a multiprocessor task, the simulation of each of the individual 70 launches should be doable. This would improve the performance greatly.
2. Implement the constraints of the solution-space directly into the optimizer to improve the search for new solutions.

### 11.2.2. Recommendations on the Modelling

1. Model the separation of the rockets from the carrier vehicles more adequately, including minimum wait time before firing, and some trajectory and pointing constraints.
2. Study the effects of throttling solutions for fuel savings, maximum acceleration requirements and other unforeseen possibilities. This suggestion comes from the literature surveyed in [7] which is briefly review in Chapters 1 and 4.
3. Include and study different solid fuel grain shapes and study the implications and possible benefits of using more optimized shapes. The currently supported cylindrical shape has a smaller burning surface than other comparable solutions (see Figure 4.4) meaning a longer grain is required for the same mass flow which brings more associated case mass for instance. Using grain geometries with neutral burning, i.e. constant burning surface over time, can limit or even annul the shift in the O/F ratio. This will permit a particular engine to operate closer to its optimal point for the entire burn increasing performance.
4. Have the Hybrid Stage propulsion module accept the O/F as an optimizable variable instead of an input. One would ideally assume that keeping the O/F as close as possible to the optimum always yields the optimum solution. However, if the added mass from the grain length and shape required to keep the O/F at a certain point could ultimately be worse than selecting an alternative O/F ratio. This is not studied here or mentioned in the literature surveyed for this work but seems to hold some promise.

# Bibliography

[1] W. J. Larson and J. R. Wertz, *Space Mission Analysis and Design*, 7th ed., edited by J. R. Wertz and W. J. Larson (Microcosm Press and Kluwer Academic Publishers, El Segundo, CA, 2005).

[2] P. J. Linstrom and W. G. Mallard, *The nist chemistry webbook: A chemical data resource on the internet,* Journal of Chemical & Engineering Data **46**, 1059 (2001).

[3] J. Vandamme, *Assisted-Launch Performance Analysis Using Trajectory and Vehicle Optimization*, Ph.D. thesis, Delft University of Technology (2012).

[4] M. W. V. Kesteren, *Air Launch versus Ground Launch: a Multidisciplinary Design Optimization Study of Expendable Launch Vehicles on Cost and Performance*, Master's thesis, Delft University of Technology (2013).

[5] D. DePasquale, A. Charania, H. Kanayama, and S. Matsuda, *Analysis of the earth-to-orbit launch market for nano and microsatellites,* in *AIAA SPACE 2010 Conference and Exposition*, Vol. 30 (2010).

[6] Y. C. Choi, K.-H. Noh, J.-W. Lee, Y.-H. Byun, and B.-K. Park, *Optimal air-launching rocket design using system trades and a multi-disciplinary optimization approach,* Aerospace Science and Technology **13**, 406 (2009).

[7] F. Miranda, *A Modelling Tool for the Study of Air and Ground Launched Hybrid Rockets*, Literature study for master thesis, Delft University of Technology (2014).

[8] B. Zandbergen, *Turbo-pump assembly mass estimation*, Tech. Rep. (TU-Delft, faculty of Aerospace Engineering, 2013).

[9] A. van Kleef and B. Oving, *Affordable Launch Opportunities for Small Satellites*, Tech. Rep. (National Aerospace Laboratory NLR, 2012).

[10] M. V. Kesteren, *Air Launch versus Ground Launch - Literature Study*, Literature study for master thesis, Delft University of Technology (2013).

[11] F. M. Engelen, *Quantitative risk analysis of unguided rocket trajectories*, Master's thesis, Delft University of Technology (2012).

[12] J.-E. Ronningen and J. Husdal, *Nammo Hybrid Rocket Propulsion TRL Improvement Program,* 48th Joint Propulsion Conference and Exhibit , 1 (2012).

[13] A. Fraters, B. Zandbergen, A. Weustink, M. W. Eiche, I. Gerth, R. Hermsen, V. Huijsman, T. Knop, S. Powell, and R. Wildvank, *Development of a Hybrid Rocket Motor for the Stratos II Rocket,* 62nd International Astronautical Congress , 1 (2010).

[14] T. Knop, B. Zandbergen, A. Cervone, R. Huijsman, S. Powell, R. Werner, J. Ehlen, F. Lindemann, J. Wink, C. Becker, and K. Samarawickrama, *Sorbitol-Based Hybrid Fuel Studies with Nitrous Oxide for the Stratos II Sounding Rocket,* 49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference , 1 (2013).

[15] J. Wink, T. Knop, R. Huijsman, S. Powell, K. Samarawickrama, A. Fraters, R. Werner, C. Becker, F. Lindemann, J. Ehlen, A. Cervone, and B. Zandbergen, *Test Campaign on a 10 kN class sorbitol-based hybrid rocket motor for the Stratos II Sounding Rocket,* in *Space Propulsion Conference, Cologne, Germany* (3AF, Cologne, 2014) pp. 1–8.

[16] J.-W. Lee, K.-H. Noh, and Y.-H. Byun, *Preliminary Design of the Hybrid Air-launching Rocket for Nanosat,* 2007 International Conference on Computational Science and its Applications (ICCSA 2007) , 290 (2007).

[17] F. Costa, R. Contaifer, J. Albuquerque, S. Gabriel, and R. Marques, *Study of Paraffin/H2O2 Hybrid Rockets for Launching Nanosats,* 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit Joint Propulsion Conferences (2008).

[18] F. D. S. Costa and R. Vieira, *Preliminary Analysis of Hybrid Rockets for Launching Nanosats into LEO,* **XXXII**, 502 (2010).

[19] F. D. S. Costa, J. Albuquerque Junior, and R. A. Contaifer, *Development of a test bench for hybrid thrusters,* in *20th International Congress of Mechanical Engineering* (2009).

[20] M. Sarigul-klijn and N. Sarigul-klijn, *A Study of Air Launch Methods for RLVs,* AIAA Space 2001 - Conference and Exposition, Albuquerque, NM, Aug. 28-30, 2001 (2001).

[21] N. Sarigul-klijn, M. Sarigul-klijn, and C. Noel, *Air-Launching Earth to Orbit : Effects of Launch Conditions Velocity Budget to Reach Low Earth Orbit,* Journal of Spacecraft and Rockets **42** (2005).

[22] J. C. Whitehead, *Air Launch Trajectories to Earth Orbit,* 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit , 9 (2006).

[23] B. Zandbergen, *Thermal Rocket Propulsion - Class Notes*, 2nd ed. (Delft University of Technology, 2010).

[24] M. Bille and E. Lishock, *NOTSNIK: The Secret Satellite,* 40th Joint Propulsion Conference and Exhibit (2002).

[25] G. E. Lozino-Lozinsky, E. N. Dudar, V. K. Chvanov, and B. I. Katorgin, *MAKS-D - Experimental Rocket Powered Plane - Demonstrator of Technologies,* AIAA/CIRA 7th International Space Planes and Hypersonics Systems and Technologies (1996).

[26] R. R. Kazmar, *Airbreathing Hypersonic Propulsion at Pratt & Whitney – Overview,* AIAA/CIRA 13th International Space Planes and Hypersonics Systems and Technologies , 1 (2005).

[27] D. E. Koelle and H. Kuczera, *Sanger II, an advanced launcher system for europe,* Acta Astronautica **19**, 63 (1989).

[28] Virgin Galactic, *Overview - Safety,* http://www.virgingalactic.com/overview/safety/ (2014).

[29] O. S. Corporation, *Pegasus Mission History,* http://www.orbital.com/SpaceLaunch/Pegasus/pegasus_history.shtml (2014).

[30] O. S. Corporation, *Pegasus - Factsheet*, Tech. Rep. (Orbital Sciences Corporation, 2014).

[31] B. T. C. Zandbergen, *AE1222-II : Aerospace Design & Systems Engineering Elements I Part : Launcher design*, 1st ed. (Delft University of Technology, Delft, 2013).

[32] M. J. Chiaverini and K. K. Kuo, *Fundamentals of hybrid rocket combustion and propulsion* (American Institute of Aeronautics and Astronautics, 2007).

[33] C. Oiknine, *New perspectives for hybrid propulsion,* 42nd Joint Propulsion Conference and Exhibit , 1 (2006).

[34] A. Karabeyoglu, J. Stevens, D. Geyzel, and B. Cantwell, *High Performance Hybrid Upper Stage Motor,* 47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit (2011).

[35] A. Karabeyoglu, *Hybrid Rocket Propulsion for Future Space Launch,* (2008).

[36] M. A. Karabeyoglu and D. Altman, *Dynamic Modeling of Hybrid Rocket Combustion,* Journal of Propulsion and Power **15**, 562 (1999).

[37] W. H. Knuth, M. J. Chiaverini, J. A. Sauer, D. J. Gramer, and O. T. Corporation, *Solid-Fuel Regression Rate Behavior of Vortex Hybrid Rocket Engines,* **18** (2002).

[38] B. Zanbergen, *Hybrid rocket motors,* http://www.lr.tudelft.nl/en/organisation/departments/space-engineering/space-systems-engineering/expertise-areas/space-propulsion/propulsion-options/chemical-rockets/hybrid/ (2015).

[39] J. McFarlane, R. Kniffen, and J. Lichatowich, *Alaa 93-2551 design and testing of amroc's 250,000 ibf thrust hybrid motor,* (1993).

[40] B. Zandbergen, *Hybrid Rocket Motors (data collection).pdf,* (1995).

[41] K. K. Rajesh, *Thrust Modulation in a Nitrous-Oxide/Hydroxyl- Terminated Polybutadiene Hybrid Rocket Motor,* 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit , 1 (2006).

[42] J. Rowland, L. Reese, and M. Laurel, *A Hybrid Rocket Engine Design for Simple Low Cost Sounding Rocket Use,* 29th Joint Propulsion Conference and Exhibit (1993).

[43] Y. Kamm and A. Gany, *Solid Rocket Motor Optimization,* 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit , 1 (2008).

[44] D. Altman, *Hybrid rocket development history,* 27th Joint Propulsion Conference and Exhibit (1991).

[45] M. Smiley, M. Veno, and R. Bell, *Commercial Crew Development — Round One, Milestone 3: Overview of Sierra Nevada Corporation's Hybrid Motor Ground Test,* 47th Joint Propulsion Conference and Exhibit (2011).

[46] R. Brown, J. Sellers, and M. Paul, *Practical experience with hydrogen peroxide hybrid rockets,* in *1st Hydrogen Peroxide Workshop, Guildford, Surrey, UK* (1998).

[47] A. Bettla, F. Moretto, E. Geremia, N. Bellomo, D. Petronio, and D. Pavarin, *Development of 20 kn hybrid rocket booster,* 49th Joint Propulsion Conference and Exhibit , 1 (2013).

[48] A. Karabeyoglu, T. Falconer, B. Cantwell, J. Stevens, S. P. Group, H. Altitude, and M. Missions, *Design Of An Orbital Hybrid Rocket Vehicle Launched From Canberra Air Platform,* 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit Joint Propulsion Conference and Exhibit , 1 (2005).

[49] C. Dupont, M. Bullock, M. Prevost, Y. Maisonneuve, R. Bec, N. Pillet, and R. Barenes, *The PERSEUS Student Launcher Project and Associated Hybrid Propulsion Activities,* 42nd Joint Propulsion Conference and Exhibit , 1 (2006).

[50] A. Ponomarenko, *Rocket Propulsion Analysis,* http://propulsion-analysis.com/index.htm.

[51] K. Lohner, J. Dyer, E. Doran, Z. Dunn, and G. Zilliac, *Fuel Regression Rate Characterization Using a Laboratory Scale Nitrous Oxide Hybrid Propulsion System,* 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit , 1 (2006).

[52] B. J. Cantwell, *Hybrid Rockets,* in *AA283 Aircraft and Rocket Propulsion - Course Reader* (Stanford University, Stanford, CA, 2007) pp. 1–14.

[53] B. Hegerty, C.-C. Hung, and K. Kasprak, *A comparative study on differential evolution and genetic algorithms for some combinatorial problems,* in *Proceedings of 8th Mexican International Conference on Artificial Intelligence* (2009).

[54] R. Mendes, J. Kennedy, and J. Neves, *The fully informed particle swarm: simpler, maybe better,* Evolutionary Computation, IEEE Transactions on **8**, 204 (2004).

[55] Delft University of Technology, *Tudat Wiki,* http://tudat.tudelft.nl/projects/tudat/wiki (2014).

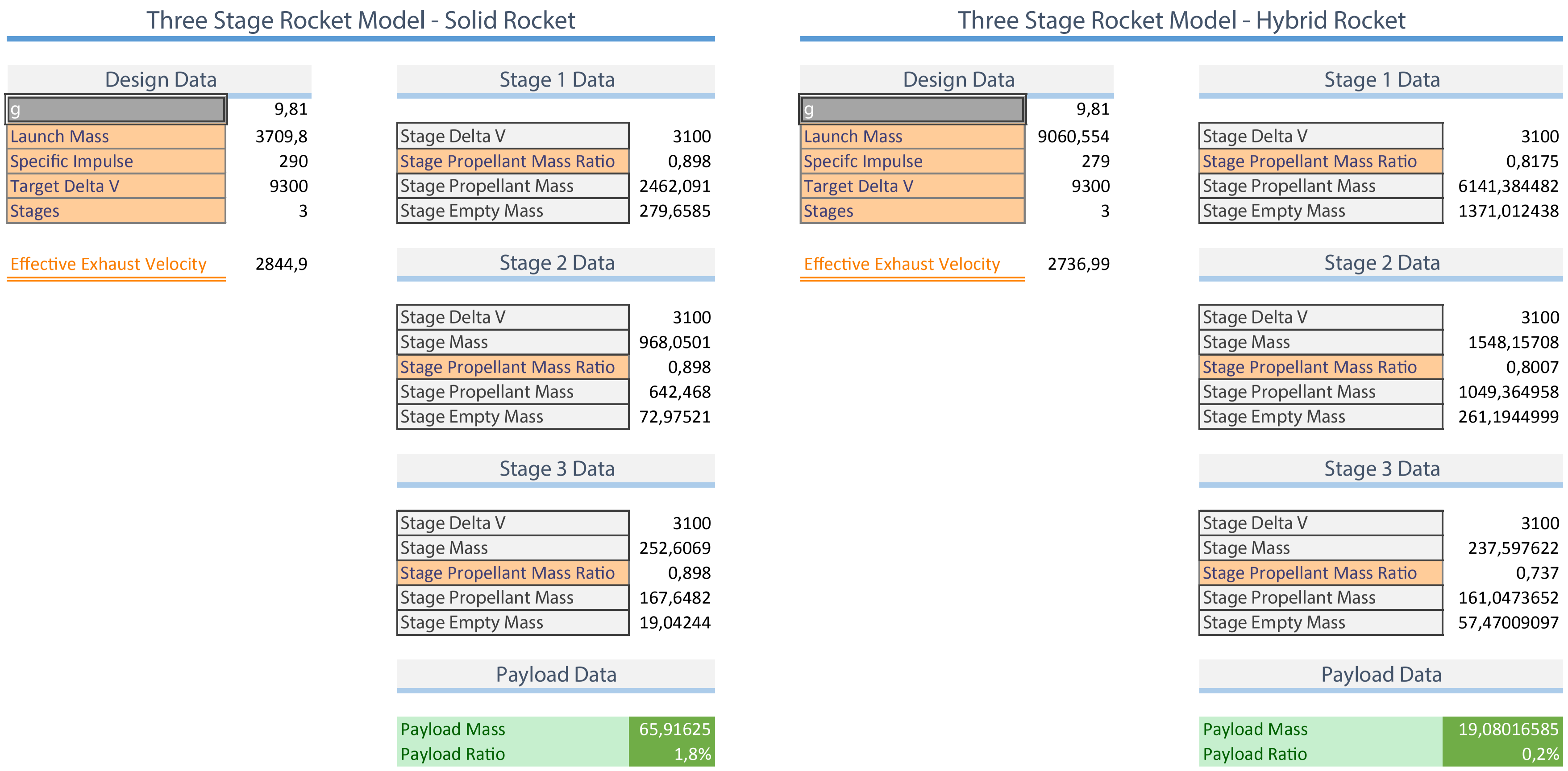
[56] D. Izzo, *Pygmo and pykep: open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization),* in *Proceed. Fifth International Conf. Astrodynam. Tools and Techniques, ICATT* (2012).

[57] NASA, *NPR 7120 . 5, NASA Space Flight Program and Project Management Handbook,* NASA's Procedural Requirements (2010).

[58] A. Siebenhaar, *Xlr-132a storable propellant rocket engine turbopump development status,* (1987).

# A

# Preliminary Models



## Three Stage Rocket Model - Solid Rocket

### Design Data

| | |
|---|---|
| g | 9,81 |
| Launch Mass | 3709,8 |
| Specific Impulse | 290 |
| Target Delta V | 9300 |
| Stages | 3 |
| Effective Exhaust Velocity | 2844,9 |

### Stage 1 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Propellant Mass Ratio | 0,898 |
| Stage Propellant Mass | 2462,091 |
| Stage Empty Mass | 279,6585 |

### Stage 2 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 968,0501 |
| Stage Propellant Mass Ratio | 0,898 |
| Stage Propellant Mass | 642,468 |
| Stage Empty Mass | 72,97521 |

### Stage 3 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 252,6069 |
| Stage Propellant Mass Ratio | 0,898 |
| Stage Propellant Mass | 167,6482 |
| Stage Empty Mass | 19,04244 |

### Payload Data

| | |
|---|---|
| Payload Mass | 65,91625 |
| Payload Ratio | 1,8% |

## Three Stage Rocket Model - Hybrid Rocket

### Design Data

| | |
|---|---|
| g | 9,81 |
| Launch Mass | 9060,554 |
| Specifc Impulse | 279 |
| Target Delta V | 9300 |
| Stages | 3 |
| Effective Exhaust Velocity | 2736,99 |

### Stage 1 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Propellant Mass Ratio | 0,8175 |
| Stage Propellant Mass | 6141,384482 |
| Stage Empty Mass | 1371,012438 |

### Stage 2 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 1548,15708 |
| Stage Propellant Mass Ratio | 0,8007 |
| Stage Propellant Mass | 1049,364958 |
| Stage Empty Mass | 261,1944999 |

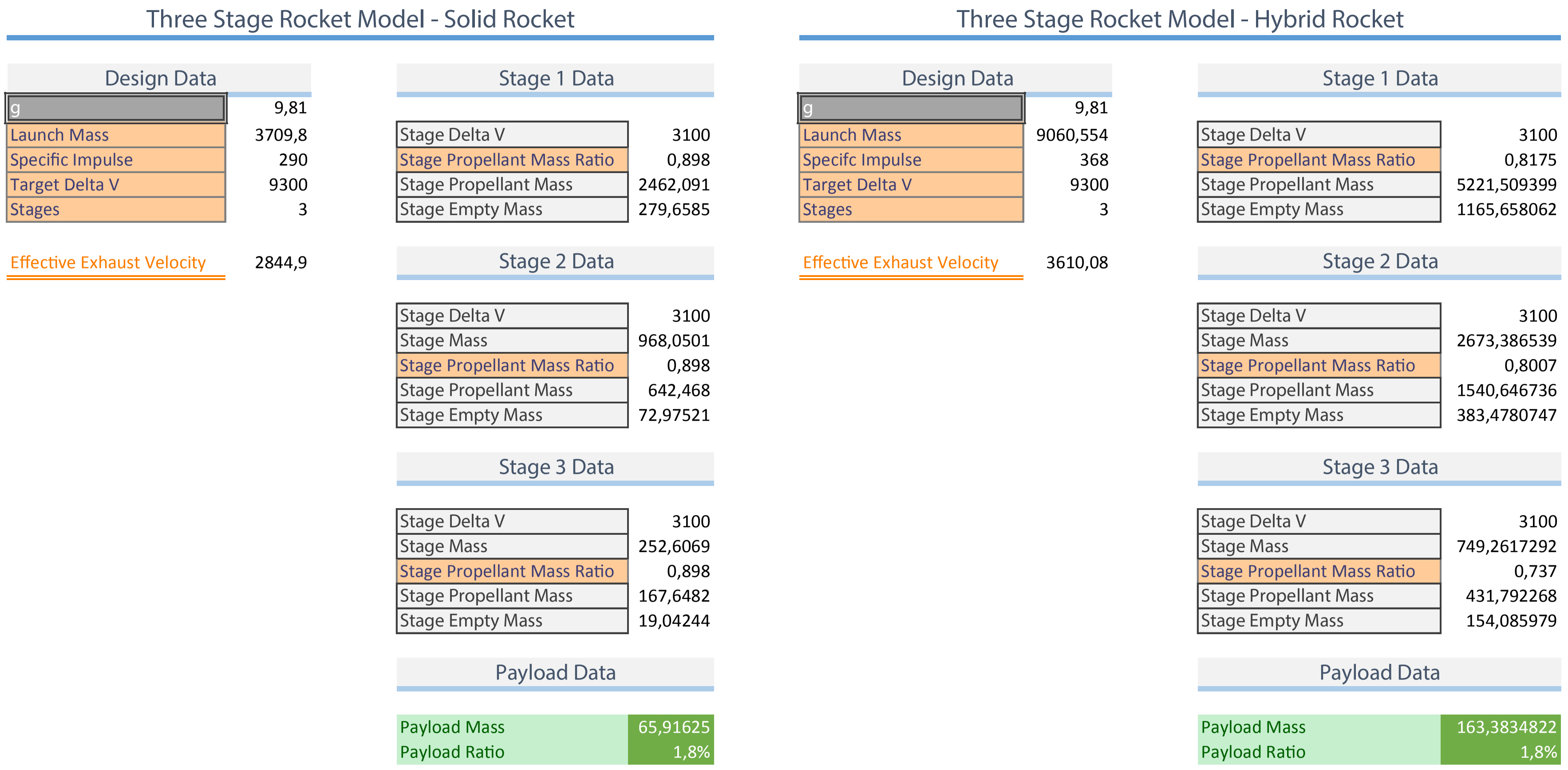
### Stage 3 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 237,597622 |
| Stage Propellant Mass Ratio | 0,737 |
| Stage Propellant Mass | 161,0473652 |
| Stage Empty Mass | 57,47009097 |

### Payload Data

| | |
|---|---|
| Payload Mass | 19,08016585 |
| Payload Ratio | 0,2% |

Figure A.1: Simple Rocket Model based on the Tsiolkovsky Rocket Equation in Microsoft Excel - Values in S.I. units from [18] and [4]



Three Stage Rocket Model - Solid Rocket

Design Data

| | |
|---|---|
| g | 9,81 |
| Launch Mass | 3709,8 |
| Specific Impulse | 290 |
| Target Delta V | 9300 |
| Stages | 3 |
| Effective Exhaust Velocity | 2844,9 |

Stage 1 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Propellant Mass Ratio | 0,898 |
| Stage Propellant Mass | 2462,091 |
| Stage Empty Mass | 279,6585 |

Stage 2 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 968,0501 |
| Stage Propellant Mass Ratio | 0,898 |
| Stage Propellant Mass | 642,468 |
| Stage Empty Mass | 72,97521 |

Stage 3 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 252,6069 |
| Stage Propellant Mass Ratio | 0,898 |
| Stage Propellant Mass | 167,6482 |
| Stage Empty Mass | 19,04244 |

Payload Data

| | |
|---|---|
| Payload Mass | 65,91625 |
| Payload Ratio | 1,8% |

Three Stage Rocket Model - Hybrid Rocket

Design Data

| | |
|---|---|
| g | 9,81 |
| Launch Mass | 9060,554 |
| Specifc Impulse | 368 |
| Target Delta V | 9300 |
| Stages | 3 |
| Effective Exhaust Velocity | 3610,08 |

Stage 1 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Propellant Mass Ratio | 0,8175 |
| Stage Propellant Mass | 5221,509399 |
| Stage Empty Mass | 1165,658062 |

Stage 2 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 2673,386539 |
| Stage Propellant Mass Ratio | 0,8007 |
| Stage Propellant Mass | 1540,646736 |
| Stage Empty Mass | 383,4780747 |

Stage 3 Data

| | |
|---|---|
| Stage Delta V | 3100 |
| Stage Mass | 749,2617292 |
| Stage Propellant Mass Ratio | 0,737 |
| Stage Propellant Mass | 431,792268 |
| Stage Empty Mass | 154,085979 |

Payload Data

| | |
|---|---|
| Payload Mass | 163,3834822 |
| Payload Ratio | 1,8% |

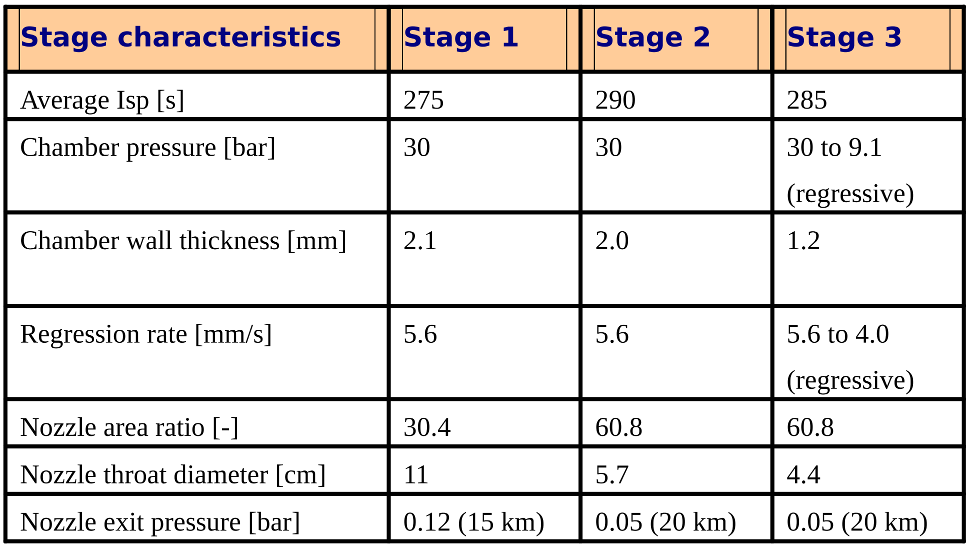
Figure A.2: Simple Rocket Model based on the Tsiolkovsky Rocket Equation in Microsoft Excel - Modified Hybrid Rocket $I_{sp}$ to achieve the same payload ratio. All Values in S.I. units from [18] and [4]

# B

# ALOSS Data

| Stage characteristics | Stage 1 | Stage 2 | Stage 3 |
|---|---|---|---|
| Mass before burn [kg] | 1,379.5 | 388.6 | 87.2 |
| Mass at end burn [kg] | 516.7 | 143.8 | 26.5 |
| Mass propellant [kg] | 862.8 | 244.8 | 60.7 |
| Mass insulation [kg] | 5.8 | 5.4 | 1.4 |
| Mass chamber [kg] | 48.5 | 20.9 | 4.9 |
| Mass nozzle [kg] | 45.9 | 13.5 | 3.5 |
| Mass TVC [kg] | 22.9 | 6.7 | 1.7 |
| Mass interstage/fairing [kg] | 5/none | 5/5 | none/none |
| Mass avionics | none | None | 5 |
| Mass payload | none | None | 10 |
| Dry mass [kg] | 128.1 | 56.5 | 26.5 |
| Propellant mass fraction [-] | 0.87 | 0.81 | 0.70 |
| Mass flow [kg/s] | 17.9 | 5.0 | 3.1 to 0.9 (regressive) |
| Thrust [kN] | 48.3 (constant) | 14.1 (constant) | 8.6 to 2.6 (regressive) |
| Max acc. (at end burn) [g's] | 9.5 | 10 | 10 |
| Thrust duration [s] | 47.4 | 49.4 | 34 |
| $\Delta V$ [km/s] | 2.65 | 2.83 | 3.33 |

Table B.1: Characteristics for the 3-stage launcher for the F-16 Air-launch concept in ALOSS (1/2) - [9]



| Stage characteristics | Stage 1 | Stage 2 | Stage 3 |
|---|---|---|---|
| Average Isp [s] | 275 | 290 | 285 |
| Chamber pressure [bar] | 30 | 30 | 30 to 9.1 (regressive) |
| Chamber wall thickness [mm] | 2.1 | 2.0 | 1.2 |
| Regression rate [mm/s] | 5.6 | 5.6 | 5.6 to 4.0 (regressive) |
| Nozzle area ratio [-] | 30.4 | 60.8 | 60.8 |
| Nozzle throat diameter [cm] | 11 | 5.7 | 4.4 |
| Nozzle exit pressure [bar] | 0.12 (15 km) | 0.05 (20 km) | 0.05 (20 km) |

Table B.2: Characteristics for the 3-stage launcher for the F-16 Air-launch concept in ALOSS (2/2) - [9]

# C

# Original Project Planning

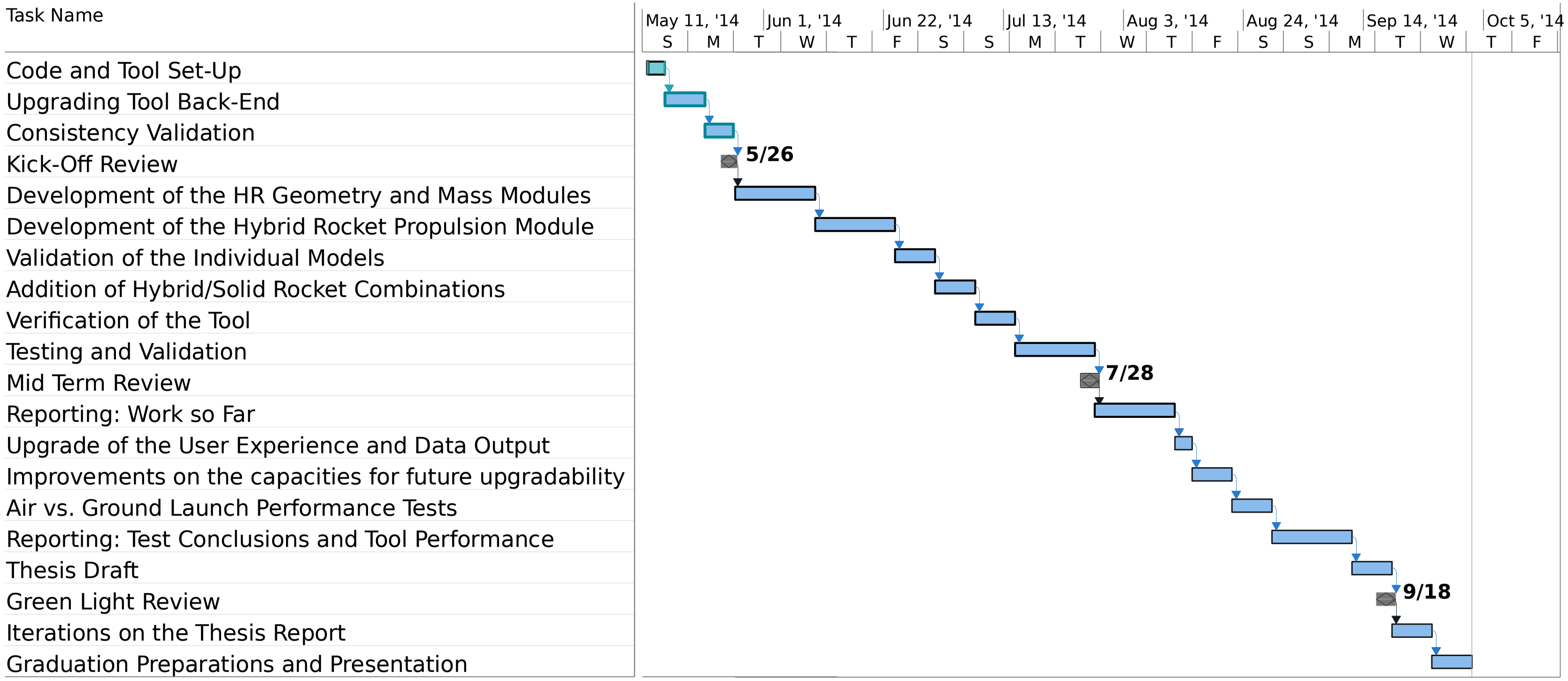


Figure C.1: Gantt Chart for the Thesis Project (graph)

| Task Name | Duration | Start | Finish |
|---|---|---|---|
| Code and Tool Set-Up | 3 days | Mon 5/12/14 | Wed 5/14/14 |
| Upgrading Tool Back-End | 1 wk | Thu 5/15/14 | Wed 5/21/14 |
| Consistency Validation | 3 days | Thu 5/22/14 | Mon 5/26/14 |
| Kick-Off Review | 0 days | Mon 5/26/14 | Mon 5/26/14 |
| Development of the HR Geometry and Mass Modules | 2 wks | Tue 5/27/14 | Mon 6/9/14 |
| Development of the Hybrid Rocket Propulsion Module | 2 wks | Tue 6/10/14 | Mon 6/23/14 |
| Validation of the Individual Models | 1 wk | Tue 6/24/14 | Mon 6/30/14 |
| Addition of Hybrid/Solid Rocket Combinations | 1 wk | Tue 7/1/14 | Mon 7/7/14 |
| Verification of the Tool | 1 wk | Tue 7/8/14 | Mon 7/14/14 |
| Testing and Validation | 2 wks | Tue 7/15/14 | Mon 7/28/14 |
| Mid Term Review | 0 days | Mon 7/28/14 | Mon 7/28/14 |
| Reporting: Work so Far | 2 wks | Tue 7/29/14 | Mon 8/11/14 |
| Upgrade of the User Experience and Data Output | 3 days | Tue 8/12/14 | Thu 8/14/14 |
| Improvements on the capacities for future upgradability | 1 wk | Fri 8/15/14 | Thu 8/21/14 |
| Air vs. Ground Launch Performance Tests | 1 wk | Fri 8/22/14 | Thu 8/28/14 |
| Reporting: Test Conclusions and Tool Performance | 2 wks | Fri 8/29/14 | Thu 9/11/14 |
| Thesis Draft | 1 wk | Fri 9/12/14 | Thu 9/18/14 |
| Green Light Review | 0 days | Thu 9/18/14 | Thu 9/18/14 |
| Iterations on the Thesis Report | 1 wk | Fri 9/19/14 | Thu 9/25/14 |
| Graduation Preparations and Presentation | 1 wk | Fri 9/26/14 | Thu 10/2/14 |

Figure C.2: Gantt Chart for the Thesis Project (dates)

# D

# Test Settings

The following data corresponds to the settings used to perform all the sample tests used throughout the report. They can be used directly with the tool to reproduce any case. Some of the data is redacted.

## Rocket Assembly and Stage Configuration

Following data is to be used to configure the rocket vehicles and the individual stages for each case.

### Nammo Rocket and Stages

```
NumberOfStages: 3

ConfigFileStage1: ConfigurationSettingsStage1.txt
ConfigFileStage2: ConfigurationSettingsStage1.txt
ConfigFileStage3: ConfigurationSettingsStage1.txt

PropellantFileStage1: PropellantDataStage1.txt
PropellantFileStage2: PropellantDataStage1.txt
PropellantFileStage3: PropellantDataStage1.txt

TypeStage1: 2
TypeStage2: 2
TypeStage3: 2

DiameterFairingForLargeVehicles: 2.3
LengthFairingForLargeVehicles: 3.3

DiameterFairingForSmallVehicles: 0.4
LengthFairingForSmallVehicles: 0.6

MassPayloadAdapterPower: 1.0132
MassPayloadAdapterCoefficient: 0.04775

MassVEBAdapterPower: 0.6798
MassVEBAdapterCoefficient: 0.3672

noseToFairingRatio: 1.0
massFairingFactor: 12.2

MassInterStageCoefficient: 7.7165
MassInterstageFirstPower: 3.2808
MassInterstageSecondPower: 0.4856
DistanceBetweenStages: 0.1
MaterialInterStage: 1.0
```

```
BarToPascalConversion: 100000.0
SafetyFactor: 1.4


AvailableSpaceForPropellant: 0.99
SliverFraction: 0.03
ThicknessInsulationLayer: 0.003

SafetyFactor: 1.4
CorrectionInertMass: 1.3
CorrectionThrust: 0.95
CombustionQuality: 0.85
InjectorPressureDropFraction: 0.01

FuelDensity: 953.0
OxidizerDensity: 1376
OxidizerTankPressure_BAR: 4.0

requiresAPressurantTank*: 2
MinimumPumpMass: 10
MinimumTankThickness: 0.001

OxidizerToFuelRatio: 7.0

ExponentK*: 9999
aO_Coefficient*: 0.0035

DensityInsulationMaterial: 850.0
DensityMotorCaseMaterial: 2200.0
DensityOxidizerTankMaterial: 2700.0
StressMotorCaseMaterial_MPA: 800.0
StressOxidizerTankMaterial_MPA: 400.0

NozzleConvergentHalfAngle: 0.523598776
NozzleDivergentHalfAngle: 0.261799388
LongitudinalRadiusThroat: 1.0
BellShapeNozzleFactor: 0.8



////////////////
//If k is set to 9999, regression rate is taken as constant and read from the a0_coefficient for

//////
// 0 - self pressurized tank, 1 - oxi tank + pressurant tank, 2 - pump system
//
```

## DARE Rocket and Stages

```
NumberOfStages: 1

ConfigFileStage1: ConfigurationSettingsStage1.txt

PropellantFileStage1: PropellantDataStage1.txt

TypeStage1: 0


DiameterFairingForLargeVehicles: 2.3
LengthFairingForLargeVehicles: 3.3

DiameterFairingForSmallVehicles: 0.0
```

```
LengthFairingForSmallVehicles: 0.0

MassPayloadAdapterPower: 0
MassPayloadAdapterCoefficient: 0

MassVEBAdapterPower: 0
MassVEBAdapterCoefficient: 0

noseToFairingRatio: 0
massFairingFactor: 0

MassInterStageCoefficient: 0
MassInterstageFirstPower: 0
MassInterstageSecondPower: 0
DistanceBetweenStages: 0
MaterialInterStage: 0

BarToPascalConversion: 100000.0
SafetyFactor: 1.4


filenameForThrustHistory: thrustHistoryStage1.txt
filenameForMassFlowHistory: massFlowHistoryStage1.txt
InertMass: 84.5
PropellanMass: 104.8
lengthStage: 7.5
NozzleConvergentHalfAngle: 0.523598776
NozzleDivergentHalfAngle: 0.261799388
LongitudinalRadiusThroat: 1.0
BellShapeNozzleFactor: 0.8
DiameterThroat: 0.0564
```

### Mixed Rocket

```
NumberOfStages: 3

ConfigFileStage1: ConfigurationSettingsStage1.txt
ConfigFileStage2: ConfigurationSettingsStage1.txt
ConfigFileStage3: ConfigurationSettingsStage1.txt

PropellantFileStage1: PropellantDataStage1.txt
PropellantFileStage2: PropellantDataStage1.txt
PropellantFileStage3: PropellantDataStage1.txt

TypeStage1: 1
TypeStage2: 1
TypeStage3: 2

DiameterFairingForLargeVehicles: 2.3
LengthFairingForLargeVehicles: 3.3

DiameterFairingForSmallVehicles: 0.4
LengthFairingForSmallVehicles: 0.6

MassPayloadAdapterPower: 1.0132
MassPayloadAdapterCoefficient: 0.04775

MassVEBAdapterPower: 0.6798
MassVEBAdapterCoefficient: 0.3672

noseToFairingRatio: 1.0
massFairingFactor: 12.2
```

```
MassInterStageCoefficient: 7.7165
MassInterstageFirstPower: 3.2808
MassInterstageSecondPower: 0.4856
DistanceBetweenStages: 0.1
MaterialInterStage: 1.0

BarToPascalConversion: 100000.0
SafetyFactor: 1.4
```

# Vehicle Optimization and Launch Configuration

Following data is to be used to configure the optimization processes and the launch configuration.

## Nammo Air Launch

```
LaunchVehicleType: 5 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
10 //Chamber Pressure Stage 1 - Start
25 //Chamber Pressure Stage 1 - End
0.05 //Exit Pressure Stage 1 - Start
0.35 //Exit Pressure Stage 1 - End
0.3 //Diameter Motor Case Stage 1 - Start
0.4 //Diameter Motor Case Stage 1 - End
0.1 //Diameter Nozzle End stage 1 - Start
0.4 //Diameter Nozzle End stage 1 - End
50 //Burn Time Stage 1 - Start
60 //Burn Time Stage 1 - End
0 //Coasting Time Stage 1 - Start
20 //Coasting Time Stage 1 - End
10 //Chamber Pressure Stage 2 - Start
30 //Chamber Pressure Stage 2 - End
0.05 //Exit Pressure Stage 2 - Start
0.4 //Exit Pressure Stage 2 - End
0.25 //Diameter Motor Case Stage 2 - Start
0.35 //Diameter Motor Case Stage 2 - End
0.1 //Diameter Nozzle End stage 2 - Start
0.17 //Diameter Nozzle End stage 2 - End
50 //Burn Time Stage 2 - Start
50 //Burn Time Stage 2 - End
100 //Coasting Time Stage 2 - Start
550 //Coasting Time Stage 2 - End
10 //Chamber Pressure Stage 3 - Start
40 //Chamber Pressure Stage 3 - End
0.01 //Exit Pressure Stage 3 - Start
0.2 //Exit Pressure Stage 3 - End
0.1 //Diameter Motor Case Stage 3 - Start
0.4 //Diameter Motor Case Stage 3 - End
0.05 //Diameter Nozzle End stage 3 - Start
0.25 //Diameter Nozzle End stage 3 - End
20 //Burn Time Stage 3 - Start
40 //Burn Time Stage 3 - End

SearchSpaceTrajectory:
30 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
30 //Pitch Angle at 0.03 total action time - Start
```

```
30 //Pitch Angle at 0.03 total action time - End
25 //Pitch Angle at 0.06 total action time - Start
30 //Pitch Angle at 0.06 total action time - End
25 //Pitch Angle at 0.10 total action time - Start
30 //Pitch Angle at 0.10 total action time - End
15 //Pitch Angle at 0.16 total action time - Start
30 //Pitch Angle at 0.16 total action time - End
10 //Pitch Angle at 0.23 total action time - Start
30 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
30 //Pitch Angle at 0.40 total action time - End
0 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

xv:
0.69663526726325
0.640924595517034
0.766673429450105
0.717764147023628
0.689296240235459
0.774719230618676
0.569533237538633
0.331341304319023
0.472343541963231
0.407806081047768
0.561905992292794
0.49517270701986
0.450517842035466
0.480934891751405
0.467471128859065
0.387221843211678
0.916977183456406
0.523130655433941
0.474887070406009
0.620828353924751
0.634345701786762
0.717628624501721
0.548469237457959
0.809364566656752
0.403873216779256
0.203056130522934
0.291536781497606
0.621090376345902

alitude: 15000.0
velocityInitial: 250.0
flightPathAngle: 50.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
```

```
MolarWeightAir: 28.9644


LaunchVehicleType: 5 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
10 //Chamber Pressure Stage 1 - Start
25 //Chamber Pressure Stage 1 – End
0.05 //Exit Pressure Stage 1 – Start
0.35 //Exit Pressure Stage 1 – End
0.3 //Diameter Motor Case Stage 1 – Start
0.4 //Diameter Motor Case Stage 1 – End
0.1 //Diameter Nozzle End stage 1 – Start
0.4 //Diameter Nozzle End stage 1 – End
50 //Burn Time Stage 1 – Start
60 //Burn Time Stage 1 – End
0 //Coasting Time Stage 1 – Start
20 //Coasting Time Stage 1 – End
10 //Chamber Pressure Stage 2 - Start
30 //Chamber Pressure Stage 2 – End
0.05 //Exit Pressure Stage 2 – Start
0.4 //Exit Pressure Stage 2 – End
0.25 //Diameter Motor Case Stage 2 – Start
0.35 //Diameter Motor Case Stage 2 – End
0.1 //Diameter Nozzle End stage 2 – Start
0.17 //Diameter Nozzle End stage 2 – End
50 //Burn Time Stage 2 – Start
50 //Burn Time Stage 2 – End
100 //Coasting Time Stage 2 – Start
550 //Coasting Time Stage 2 – End
10 //Chamber Pressure Stage 3 - Start
40 //Chamber Pressure Stage 3 – End
0.01 //Exit Pressure Stage 3 – Start
0.2 //Exit Pressure Stage 3 – End
0.1 //Diameter Motor Case Stage 3 – Start
0.4 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.25 //Diameter Nozzle End stage 3 – End
20 //Burn Time Stage 3 – Start
40 //Burn Time Stage 3 – End

SearchSpaceTrajectory:
30 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
30 //Pitch Angle at 0.03 total action time - Start
30 //Pitch Angle at 0.03 total action time - End
25 //Pitch Angle at 0.06 total action time - Start
30 //Pitch Angle at 0.06 total action time - End
25 //Pitch Angle at 0.10 total action time - Start
30 //Pitch Angle at 0.10 total action time - End
15 //Pitch Angle at 0.16 total action time - Start
30 //Pitch Angle at 0.16 total action time - End
10 //Pitch Angle at 0.23 total action time - Start
30 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
30 //Pitch Angle at 0.40 total action time - End
0 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
```

```
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

alitude: 15000.0
velocityInitial: 250.0
flightPathAngle: 50.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644
```

## Nammo Ground Launch

```
LaunchVehicleType: 2 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
10 //Chamber Pressure Stage 1 – Start
20 //Chamber Pressure Stage 1 – End
0.05 //Exit Pressure Stage 1 – Start
0.5 //Exit Pressure Stage 1 – End
0.3 //Diameter Motor Case Stage 1 – Start
0.7 //Diameter Motor Case Stage 1 – End
0.1 //Diameter Nozzle End stage 1 – Start
0.6 //Diameter Nozzle End stage 1 – End
50 //Burn Time Stage 1 – Start
80 //Burn Time Stage 1 – End
0 //Coasting Time Stage 1 – Start
20 //Coasting Time Stage 1 – End
10 //Chamber Pressure Stage 2 – Start
30 //Chamber Pressure Stage 2 – End
0.05 //Exit Pressure Stage 2 – Start
0.5 //Exit Pressure Stage 2 – End
0.2 //Diameter Motor Case Stage 2 – Start
0.6 //Diameter Motor Case Stage 2 – End
0.1 //Diameter Nozzle End stage 2 – Start
0.17 //Diameter Nozzle End stage 2 – End
30 //Burn Time Stage 2 – Start
70 //Burn Time Stage 2 – End
100 //Coasting Time Stage 2 – Start
550 //Coasting Time Stage 2 – End
10 //Chamber Pressure Stage 3 – Start
30 //Chamber Pressure Stage 3 – End
0.05 //Exit Pressure Stage 3 – Start
0.25 //Exit Pressure Stage 3 – End
0.75 //Diameter Motor Case Stage 3 – Start
0.2 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.2 //Diameter Nozzle End stage 3 – End
```

```
20 //Burn Time Stage 3 – Start
40 //Burn Time Stage 3 – End
SearchSpaceTrajectory:
70 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
70 //Pitch Angle at 0.03 total action time - Start
25 //Pitch Angle at 0.03 total action time - End
50 //Pitch Angle at 0.06 total action time - Start
25 //Pitch Angle at 0.06 total action time - End
50 //Pitch Angle at 0.10 total action time - Start
25 //Pitch Angle at 0.10 total action time - End
35 //Pitch Angle at 0.16 total action time - Start
25 //Pitch Angle at 0.16 total action time - End
25 //Pitch Angle at 0.23 total action time - Start
25 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
25 //Pitch Angle at 0.40 total action time - End
5 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

xv:
0.820856882386384
0.933211864528596
0.709788937328187
0.764077585762801
0.327172434028812
0.637739380390234
0.617844288081125
0.606734426672162
0.985392430044577
0.879476870958643
0.960524742935035
0.624263136809372
0.407293042091584
0.483870823659715
0.0711570088181355
0.527652266875803
0.972981044326886
0.560230204663113
0.704162179208357
0.60502079958841
0.0924683485556407
0.376034059545148
0.357388649600524
0.378653387320467
0.615108378005108
0.428122891943876
0.556021228890016
0.468193205474555

alitude: 0.0
velocityInitial: 1.0
flightPathAngle: 90.0
longitude: 0.0
latitude: 0.0
```

```
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644


LaunchVehicleType: 2 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
10 //Chamber Pressure Stage 1 - Start
20 //Chamber Pressure Stage 1 – End
0.05 //Exit Pressure Stage 1 – Start
0.5 //Exit Pressure Stage 1 – End
0.3 //Diameter Motor Case Stage 1 – Start
0.7 //Diameter Motor Case Stage 1 – End
0.1 //Diameter Nozzle End stage 1 – Start
0.6 //Diameter Nozzle End stage 1 – End
50 //Burn Time Stage 1 – Start
80 //Burn Time Stage 1 – End
0 //Coasting Time Stage 1 – Start
20 //Coasting Time Stage 1 – End
10 //Chamber Pressure Stage 2 - Start
30 //Chamber Pressure Stage 2 – End
0.05 //Exit Pressure Stage 2 – Start
0.5 //Exit Pressure Stage 2 – End
0.2 //Diameter Motor Case Stage 2 – Start
0.6 //Diameter Motor Case Stage 2 – End
0.1 //Diameter Nozzle End stage 2 – Start
0.17 //Diameter Nozzle End stage 2 – End
30 //Burn Time Stage 2 – Start
70 //Burn Time Stage 2 – End
100 //Coasting Time Stage 2 – Start
550 //Coasting Time Stage 2 – End
10 //Chamber Pressure Stage 3 - Start
30 //Chamber Pressure Stage 3 – End
0.05 //Exit Pressure Stage 3 – Start
0.25 //Exit Pressure Stage 3 – End
0.75 //Diameter Motor Case Stage 3 – Start
0.2 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.2 //Diameter Nozzle End stage 3 – End
20 //Burn Time Stage 3 – Start
40 //Burn Time Stage 3 – End
SearchSpaceTrajectory:
70 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
70 //Pitch Angle at 0.03 total action time - Start
25 //Pitch Angle at 0.03 total action time - End
50 //Pitch Angle at 0.06 total action time - Start
25 //Pitch Angle at 0.06 total action time - End
50 //Pitch Angle at 0.10 total action time - Start
25 //Pitch Angle at 0.10 total action time - End
35 //Pitch Angle at 0.16 total action time - Start
25 //Pitch Angle at 0.16 total action time - End
25 //Pitch Angle at 0.23 total action time - Start
```

```
25 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
25 //Pitch Angle at 0.40 total action time - End
5 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

alitude: 0.0
velocityInitial: 1.0
flightPathAngle: 90.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644
```

## DARE Ground Launch

```
LaunchVehicleType: 2 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
30 //Chamber Pressure Stage 1 - Start
70 //Chamber Pressure Stage 1 - End
0.05 //Exit Pressure Stage 1 - Start
0.95 //Exit Pressure Stage 1 - End
0.3 //Diameter Motor Case Stage 1 - Start
0.373 //Diameter Motor Case Stage 1 - End
0.1 //Diameter Nozzle End stage 1 - Start
0.273 //Diameter Nozzle End stage 1 - End
30 //Burn Time Stage 1 - Start
50 //Burn Time Stage 1 - End
1 //Coasting Time Stage 1 - Start
49 //Coasting Time Stage 1 - End
30 //Chamber Pressure Stage 2 - Start
50 //Chamber Pressure Stage 2 - End
0.05 //Exit Pressure Stage 2 - Start
0.2 //Exit Pressure Stage 2 - End
0.3 //Diameter Motor Case Stage 2 - Start
0.373 //Diameter Motor Case Stage 2 - End
0.1 //Diameter Nozzle End stage 2 - Start
0.17 //Diameter Nozzle End stage 2 - End
20 //Burn Time Stage 2 - Start
60 //Burn Time Stage 2 - End
100 //Coasting Time Stage 2 - Start
550 //Coasting Time Stage 2 - End
30 //Chamber Pressure Stage 3 - Start
50 //Chamber Pressure Stage 3 - End
0.05 //Exit Pressure Stage 3 - Start
```

```
0.1 //Exit Pressure Stage 3 – End
0.1 //Diameter Motor Case Stage 3 – Start
0.2 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.2 //Diameter Nozzle End stage 3 – End
5 //Burn Time Stage 3 – Start
75 //Burn Time Stage 3 – End
SearchSpaceTrajectory:
70 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
70 //Pitch Angle at 0.03 total action time - Start
25 //Pitch Angle at 0.03 total action time - End
50 //Pitch Angle at 0.06 total action time - Start
25 //Pitch Angle at 0.06 total action time - End
50 //Pitch Angle at 0.10 total action time - Start
25 //Pitch Angle at 0.10 total action time - End
35 //Pitch Angle at 0.16 total action time - Start
25 //Pitch Angle at 0.16 total action time - End
25 //Pitch Angle at 0.23 total action time - Start
25 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
25 //Pitch Angle at 0.40 total action time - End
5 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End
xv:
0.49247313553578864
0.57157640283979838
0.33473818040594422
0.59319629545889963
0.90909287200030731
0.33960275778657872
0.084830222411200265
0.36604220062019216
0.0016227553328645415
0.80821207136713036
0.99095224887743849
0.5046982149565985
0.222334006613651l
0.30801859390910585
0.90642885378561888
0.42419023743172024
0.94555247704699286
0.59190937903431262
0.51243921237904821
0.5997466127170672
0.16189501019613226
0.61220757292380668
0.39468718089963989
0.59578321170382886
0.27556381376413303
0.30620379258035313
0.26396166669839338
0.48908287724219091

alitude: 0.0
```

```
velocityInitial: 1.0
flightPathAngle: 90.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644


LaunchVehicleType: 2 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
30 //Chamber Pressure Stage 1 - Start
70 //Chamber Pressure Stage 1 – End
0.05 //Exit Pressure Stage 1 – Start
0.95 //Exit Pressure Stage 1 – End
0.3 //Diameter Motor Case Stage 1 – Start
0.373 //Diameter Motor Case Stage 1 – End
0.1 //Diameter Nozzle End stage 1 – Start
0.273 //Diameter Nozzle End stage 1 – End
30 //Burn Time Stage 1 – Start
50 //Burn Time Stage 1 – End
1 //Coasting Time Stage 1 – Start
49 //Coasting Time Stage 1 – End
30 //Chamber Pressure Stage 2 - Start
50 //Chamber Pressure Stage 2 – End
0.05 //Exit Pressure Stage 2 – Start
0.2 //Exit Pressure Stage 2 – End
0.3 //Diameter Motor Case Stage 2 – Start
0.373 //Diameter Motor Case Stage 2 – End
0.1 //Diameter Nozzle End stage 2 – Start
0.17 //Diameter Nozzle End stage 2 – End
20 //Burn Time Stage 2 – Start
60 //Burn Time Stage 2 – End
100 //Coasting Time Stage 2 – Start
550 //Coasting Time Stage 2 – End
30 //Chamber Pressure Stage 3 - Start
50 //Chamber Pressure Stage 3 – End
0.05 //Exit Pressure Stage 3 – Start
0.1 //Exit Pressure Stage 3 – End
0.1 //Diameter Motor Case Stage 3 – Start
0.2 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.2 //Diameter Nozzle End stage 3 – End
5 //Burn Time Stage 3 – Start
75 //Burn Time Stage 3 – End
SearchSpaceTrajectory:
70 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
70 //Pitch Angle at 0.03 total action time - Start
25 //Pitch Angle at 0.03 total action time - End
50 //Pitch Angle at 0.06 total action time - Start
25 //Pitch Angle at 0.06 total action time - End
50 //Pitch Angle at 0.10 total action time - Start
25 //Pitch Angle at 0.10 total action time - End
```

```
35 //Pitch Angle at 0.16 total action time - Start
25 //Pitch Angle at 0.16 total action time - End
25 //Pitch Angle at 0.23 total action time - Start
25 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
25 //Pitch Angle at 0.40 total action time - End
5 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

alitude: 0.0
velocityInitial: 1.0
flightPathAngle: 90.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644
```

## Mixed Rocket Air Launch

```
LaunchVehicleType: 5 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
10 //Chamber Pressure Stage 1 – Start
25 //Chamber Pressure Stage 1 – End
0.05 //Exit Pressure Stage 1 – Start
0.35 //Exit Pressure Stage 1 – End
0.3 //Diameter Motor Case Stage 1 – Start
0.4 //Diameter Motor Case Stage 1 – End
0.1 //Diameter Nozzle End stage 1 – Start
0.4 //Diameter Nozzle End stage 1 – End
50 //Burn Time Stage 1 – Start
60 //Burn Time Stage 1 – End
0 //Coasting Time Stage 1 – Start
20 //Coasting Time Stage 1 – End
10 //Chamber Pressure Stage 2 – Start
30 //Chamber Pressure Stage 2 – End
0.05 //Exit Pressure Stage 2 – Start
0.4 //Exit Pressure Stage 2 – End
0.25 //Diameter Motor Case Stage 2 – Start
0.35 //Diameter Motor Case Stage 2 – End
0.1 //Diameter Nozzle End stage 2 – Start
0.17 //Diameter Nozzle End stage 2 – End
50 //Burn Time Stage 2 – Start
50 //Burn Time Stage 2 – End
100 //Coasting Time Stage 2 – Start
550 //Coasting Time Stage 2 – End
```

```
10 //Chamber Pressure Stage 3 - Start
40 //Chamber Pressure Stage 3 – End
0.01 //Exit Pressure Stage 3 – Start
0.2 //Exit Pressure Stage 3 – End
0.1 //Diameter Motor Case Stage 3 – Start
0.4 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.25 //Diameter Nozzle End stage 3 – End
20 //Burn Time Stage 3 – Start
40 //Burn Time Stage 3 – End

SearchSpaceTrajectory:
30 //Pitch Angle at 0.01 total action time - Start
25 //Pitch Angle at 0.01 total action time - End
30 //Pitch Angle at 0.03 total action time - Start
30 //Pitch Angle at 0.03 total action time - End
25 //Pitch Angle at 0.06 total action time - Start
30 //Pitch Angle at 0.06 total action time - End
25 //Pitch Angle at 0.10 total action time - Start
30 //Pitch Angle at 0.10 total action time - End
15 //Pitch Angle at 0.16 total action time - Start
30 //Pitch Angle at 0.16 total action time - End
10 //Pitch Angle at 0.23 total action time - Start
30 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
30 //Pitch Angle at 0.40 total action time - End
0 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

xv:
0.510629154497368
0.396002411782195
0.394545919600255
0.647368227437438
0.488467022984982
0.561299999422699
0.522123612070851
0.364742590566094
0.428516458760798
0.288608573526686
0.665624243290972
0.587474790101954
0.444804635925931
0.464827758588067
0.493225599011219
0.3957848715394
0.829071492775792
0.580158092290856
0.522776761077615
0.685708169950084
0.480440069124258
0.540435704266546
0.821907109446991
0.633413353574144
0.725157456296767
```

```
0.363207780177308
0.42467080518854
0.581874636689036

alitude: 15000.0
velocityInitial: 250.0
flightPathAngle: 50.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644



LaunchVehicleType: 5 // [-] 1= ground launched large, 2= ground launched small, 3= air launched la
NumberOfStages: 3
MassPayload: 10
MeanChordLength: 0.0

SearchSpace:
10 //Chamber Pressure Stage 1 - Start
25 //Chamber Pressure Stage 1 – End
0.05 //Exit Pressure Stage 1 – Start
0.35 //Exit Pressure Stage 1 – End
0.3 //Diameter Motor Case Stage 1 – Start
0.4 //Diameter Motor Case Stage 1 – End
0.1 //Diameter Nozzle End stage 1 – Start
0.4 //Diameter Nozzle End stage 1 – End
50 //Burn Time Stage 1 – Start
60 //Burn Time Stage 1 – End
0 //Coasting Time Stage 1 – Start
20 //Coasting Time Stage 1 – End
10 //Chamber Pressure Stage 2 - Start
30 //Chamber Pressure Stage 2 – End
0.05 //Exit Pressure Stage 2 – Start
0.4 //Exit Pressure Stage 2 – End
0.25 //Diameter Motor Case Stage 2 – Start
0.35 //Diameter Motor Case Stage 2 – End
0.1 //Diameter Nozzle End stage 2 – Start
0.17 //Diameter Nozzle End stage 2 – End
50 //Burn Time Stage 2 – Start
50 //Burn Time Stage 2 – End
100 //Coasting Time Stage 2 – Start
550 //Coasting Time Stage 2 – End
10 //Chamber Pressure Stage 3 - Start
40 //Chamber Pressure Stage 3 – End
0.01 //Exit Pressure Stage 3 – Start
0.2 //Exit Pressure Stage 3 – End
0.1 //Diameter Motor Case Stage 3 – Start
0.4 //Diameter Motor Case Stage 3 – End
0.05 //Diameter Nozzle End stage 3 – Start
0.25 //Diameter Nozzle End stage 3 – End
20 //Burn Time Stage 3 – Start
40 //Burn Time Stage 3 – End

SearchSpaceTrajectory:
30 //Pitch Angle at 0.01 total action time - Start
```

```
25 //Pitch Angle at 0.01 total action time - End
30 //Pitch Angle at 0.03 total action time - Start
30 //Pitch Angle at 0.03 total action time - End
25 //Pitch Angle at 0.06 total action time - Start
30 //Pitch Angle at 0.06 total action time - End
25 //Pitch Angle at 0.10 total action time - Start
30 //Pitch Angle at 0.10 total action time - End
15 //Pitch Angle at 0.16 total action time - Start
30 //Pitch Angle at 0.16 total action time - End
10 //Pitch Angle at 0.23 total action time - Start
30 //Pitch Angle at 0.23 total action time - End
10 //Pitch Angle at 0.40 total action time - Start
30 //Pitch Angle at 0.40 total action time - End
0 //Pitch Angle at 0.50 total action time - Start
20 //Pitch Angle at 0.50 total action time - End
0 //Pitch Angle at 0.60 total action time - Start
20 //Pitch Angle at 0.60 total action time - End
0 //Pitch Angle at 0.75 total action time - Start
15 //Pitch Angle at 0.75 total action time - End
0 //Pitch Angle at 0.85 total action time - Start
10 //Pitch Angle at 0.85 total action time - End

alitude: 15000.0
velocityInitial: 250.0
flightPathAngle: 50.0
longitude: 0.0
latitude: 0.0
azimuthAngle: 90.0 //0.0 is Oriented to North-Pole

siderealDay: 86164.09
earthRadius:  6378137.0
MolarGasConstant: 8314.32
ratioOfSpecificHeatsAir: 1.401
MolarWeightAir: 28.9644
```