AeroBest 2025
Lisboa, 22–24 April 2025

DEVELOPMENT OF A SOUNDING ROCKET MULTIDISCIPLINARY PRELIMINARY DESIGN OPTIMISATION FRAMEWORK WITH TRAJECTORY OPTIMISATION

Palaio, A. M.[1*], Marta, A. C.[2] and Gil, P. J. S.[2]

1: Academia da Força Aérea
Força Aérea Portuguesa
Granja do Marquês , 2715-021, Sintra, Portugal
ampalaio@academiafa.edu.pt, https://academiafa.edu.pt/

2: IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001, Lisboa, Portugal
{andre.marta, paulo.gil}@tecnico.ulisboa.pt, https://mdo.tecnico.ulisboa.pt/

Abstract. *The design of rockets is known to be a complex task, not only due to the harsh operating conditions but also the strong coupling among disciplines. A multidisciplinary optimisation (MDO) framework was developed, aimed at providing preliminary designs of a single-stage solid propellant rocket. The choice of the optimiser algorithm, MDO architecture and discipline models, namely, mass and sizing, flight dynamics, aerodynamics, propulsion, structural and atmospheric, were such that the developed numerical tool has a very low computational cost while being able to meet a set of pre-established mission requirements. The resulting design framework solved a co-design optimisation problem, due to the coupling between the trajectory and rocket sizing optimization processes. The capabilities of the design framework were tested for different sets of design variables and multiple missions, with increasing complexity, for an optimisation problem aimed at minimizing the total mass of the rocket while imposing a minimum altitude constraint, with a prescribed payload capacity. First, a case study with 10 geometric design variables achieved a total mass reduction of 27.7 % when compared to a real rocket, REXUS 2. Then, sensitivity studies of the payload and the minimum altitude confirmed that the rocket sizing is significantly impacted by both. The modularity of the framework allows a straightforward extension to other types of rockets, such as multi-stage or liquid-propellant.*

Keywords: MDO, Trajectory, Co-design, Sounding Rocket, Modularity

1 INTRODUCTION

Over the last two decades, a new generation of entrepreneurs has made an unprecedented investment in Space, completely changing the paradigm. Currently, private initiatives play an important role in the future of the space industry, which is no longer controlled by the political agendas of a few superpowers [1]. As a consequence, space exploration, space tourism and space infrastructure are now the main focus of such private and semi-private initiatives, which have completely revived the global space economy. As of 2022, 78% comes from commercial space products, services, infrastructure and support industries and only 22% from government budgets [2].

As Science has always been the major beneficiary from space human endeavours [3], it is expected that this new interest in space affairs will award scientific groups with new lines of investment across a wide range of applications, such as, Research and Development (R&D) on new Launch Vehicle (LV) designs capable of accomplishing their assigned goals in compliance with the most demanding mission requirements.

Currently, the scientific research on modern Multidisciplinary Design Optimization (MDO) methods applied to the design process of LV is a hotspot in the aerospace industry, in an effort to further minimize the material usage, manpower, cost and time, while maximising the reliability, operability and safety of such systems [4].

The main goal of this work is, then, to develop and validate an MDO framework coupled with trajectory optimisation capable of conducting the preliminary design of sounding rockets with a minimum payload capacity of 98 kg and 100 km minimum peak altitude, so that the results may be compared to well known and documented rockets, namely, the Rocket borne Experiments for University Students program (REXUS) [5].

2 ROCKET FUNDAMENTALS

2.1 Mass and Sizing

A model, subdivided into six smaller subcomponents, each of them related to a main rocket part, was created from a set of analytical equations to estimate the masses and component sizing. The targeted rocket parts were: nose cone, modules, fins, nozzle, body tube, and SRM, sorted by the model execution order. Additionally, a final component was also created to calculate a few general properties, namely, the initial mass, empty mass, structural mass, and structural factor of the rocket, as it is illustrated in Fig. 1.

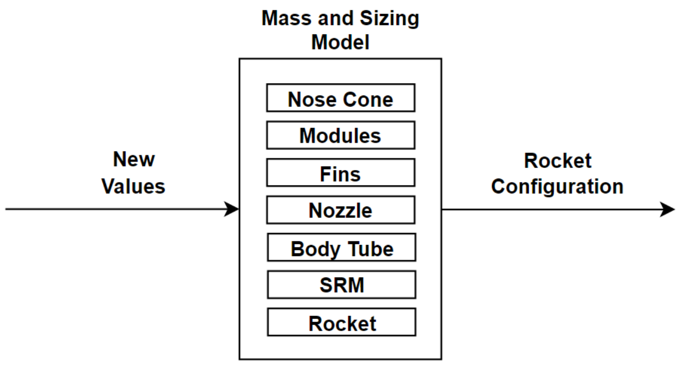


Figure 1: Mass and sizing model simplified schema highlighting the inpust, outputs and inner components.

2.2 Aerodynamics

An aerodynamic model was designed to estimate the aerodynamic behaviour of the rocket at each operating state. The C_d profile of the rocket was calculated based on three main drag sources: nose cone, base, and fins, as stated in the Equation 1.

$$Cd_0 = Cd_{nc} + Cd_b + Cd_f. \tag{1}$$

A compressible flow correction was applied for better accuracy under compressible flow regimes [6].

At subsonic speed ($M_a < 0.8$), the compressible flow correction for the aerodynamic coefficient is defined as

$$C_d = \frac{C_i}{\sqrt{1 - M_a^2}}, \tag{2}$$

where M_a is the free stream Mach number. At the transonic region ($0.8 \leq M_a \leq 1.1$), the corrected aerodynamic coefficient is given by

$$C_d = \frac{C_i}{\sqrt{1 - (0.8)^2}}. \tag{3}$$

At supersonic speed ($M_a > 1.1$), the corrected aerodynamic coefficient is

$$C_d = \frac{C_i}{\sqrt{M_a^2 - 1}}. \tag{4}$$

In addition, a recovery system contribution was also integrated in the model to simulate the behavior of parachute deployment during the descent phase of the flight profile, namely drogue and main parachutes [7].

As the rocket descends and passes the reference altitude of drogue parachute deployment, an additional component of induced drag is added to the overall C_d profile calculated using the following expression:

$$D_{drogue} = \frac{1}{2}\rho v^2 C_{D\,drogue} S_{drogue}\ , \tag{5}$$

where D_{drogue} is the drag component due to the drogue parachute, $C_{D\,drogue}$ is the drag coefficient of the drogue parachute, and S_{drogue} is the drogue parachute area.

Similarly, after the main parachute descent altitude is reached, the additional component of induced drag is calculated as

$$D_{main\,parachute} = \frac{1}{2}\rho v^2 C_{D\,main\,parachute} S_{main\,parachute}\ , \tag{6}$$

where $D_{main\,parachute}$ is the drag component due to the main parachute, $C_{D\,main\,parachute}$ is the drag coefficient of the main parachute, and $S_{main\,parachute}$ is the main parachute area.

2.3 Propulsion

A propulsion model was developed to accurately predict the behaviour of the main physical properties of a Solid Rocket Motor (SRM) under real operating conditions.

Based on several authors [8–12], a set of analytical equations was assembled to model the grain burnback, i.e., the propellant regression rate and respective propellant burning areas over time,

$$m_{pi} = m_{p0} - (m_0 - m_i) \ , \tag{7}$$

$$R_p = \sqrt{\frac{-m_{pi}}{\rho_p \pi L_{grain} + R_i^2}} \ , \tag{8}$$

$$A_b = 2\pi R_p L_{grain} \ , \tag{9}$$

where m_{pi} is the instantaneous propellant mass, m_{p0} is the initial propellant mass, m_0 is the rocket lift-off mass, m_i is instantaneous rocket mass, R_p is the port radius, ρ_p is the propellant density, L_{grain}, is the grain length, R_i is the grain inner radius, and A_b is the instantaneous burning area.

Then, using the propellant burning area as the main input, a second set of equations was assembled to model the internal ballistic behaviour of the motor, namely the combustion chamber pressure and the thrust, using one-dimensional isentropic flow equations [13], as

$$T_e = T_t(1 + \frac{\gamma - 1}{2} M_e^2)^{-1} \ , \tag{10}$$

$$P_t = \left[\frac{a \rho_p A_b}{C_D A_t}\right]^{\frac{1}{1-n}} \ , \tag{11}$$

$$P_e = P_t(1 + \frac{\gamma - 1}{2} M_e^2)^{\frac{-\gamma}{\gamma - 1}} \ , \tag{12}$$

$$V_e = M_e \sqrt{\gamma R T_e} \ , \tag{13}$$

$$\dot{m} = -C_D A_t P_t \ , \tag{14}$$

$$Thrust = \dot{m} V_e + (P_e - P) A_e \ , \tag{15}$$

where a is the burn rate coefficient, C_D is the nozzle discharge coefficient, A_t is the nozzle throat area, n is the propellant ballistic exponent, T_e is the nozzle exit temperature, T_t is the nozzle throat temperature, γ is the gas specific heat ratio, M_e is the nozzle exit mach number, P_e is the nozzle exit pressure, P_t is the internal casing total pressure, V_e is the nozzle exit velocity, R is the universal gas constant, and P is the atmospheric pressure.

2.4 Structures

A model was created to assess the structural integrity of the rocket along the flight profile. First, given the sum of drag, thrust and weight forces, the model calculates the resulting compressive loading at the body tube cross-sectional area. Then, it compares it with the linearised critical buckling stress of a thin elastic cylindrical shell in order to evaluate if, at any moment in time, the rocket was subject to such a loading condition for buckling to occur, through the expression:

$$\sigma_{crit} = \frac{\gamma E}{\sqrt{3(1-v^2)}} \left(\frac{th}{R}\right) \,, \tag{16}$$

where E is the cylindrical shell Young modulus, R its radius, th its thickness, v the material Poisson's ratio and γ a multiplication factor given by

$$\gamma = 1 - 0.901\left(1 - e^{-\phi}\right) \quad \text{with} \quad \phi = \frac{1}{16}\sqrt{\frac{R}{th}} \,. \tag{17}$$

.

Additionally, using the Flutter Boundary Equation [14], the fin flutter velocity is also monitored throughout the entire flight profile in order to evaluate if the structural integrity of the fins remains unharmed, as this is the pivotal rocket component for stability,

$$V_f = a\sqrt{\frac{G_E}{\frac{YAR^3}{(t/c_r)^3(AR+2)}\left(\frac{\lambda+1}{2}\right)\left(\frac{P}{P_0}\right)}} \,. \tag{18}$$

2.5 Atmosphere

An atmospheric model adapted from the *OpenAero-Struct Python* library was also developed to provide with the main atmospheric properties throughout the trajectory. First, arrays with the values of each atmospheric parameter, retrieved from the standard atmosphere convention tables [15, 16], were created.

By interpolating the altitude value (model input) using the *Akima1DInterpolator* class imported from the *Scipy Python* library [17, 18], it was possible to find the respective values of all atmospheric parameters for each altitude, namely, temperature T, pressure P_a, density ρ, speed of sound c, gravitational acceleration g, dynamic viscosity μ and kinematic viscosity k. These atmospheric property values are the outputs of the model, which will be fed to the other models within the trajectory group, as it is ilustrated in Fig. 2.

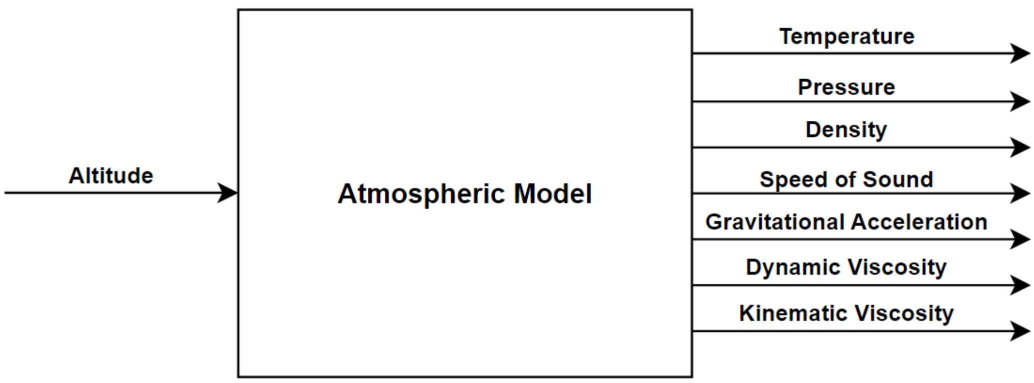


Figure 2: Atmospheric model diagram highlighting inputs and outputs.

2.6 Flight Dynamics

A set of flight dynamics equations was used, capable of translating the complex interactions between the rocket, the atmosphere, and any other external factors with active influence on the rocket [19].

In order to reduce the number of state variables for simplicity and computational cost efficiency, a 2 degrees of freedom (DoF) trajectory defined in a vertical plane was chosen over other more complex dynamics systems, with higher DoF.

The flight dynamics of the rocket can be reduced to the following set of equations:

$$\dot{V} = \frac{T}{m}\cos\alpha - \frac{D}{m} - g\sin\gamma, \tag{19}$$

$$\dot{\gamma} = -\left(\frac{g}{V} - \frac{V}{R_e + h}\right)\cos\gamma + \frac{T}{m}\sin\alpha, \tag{20}$$

$$\dot{x} = V\cos\gamma, \tag{21}$$

$$\dot{h} = V\sin\gamma. \tag{22}$$

where $\dot{v}$ is the rate change of the velocity, $\dot{\gamma}$ is the rate change of the pitch angle, $\dot{h}$ the rate change in altitude, and $\dot{x}$ the rate change in downrange [19].

2.7 Trajectory

In order to implement the trajectory model, a high-level group was created, with 5 coupled models within, namely, Flight Dynamics, Atmospheric, Propulsion, Aerodynamics and Structural, as illustrated in Fig. 3.

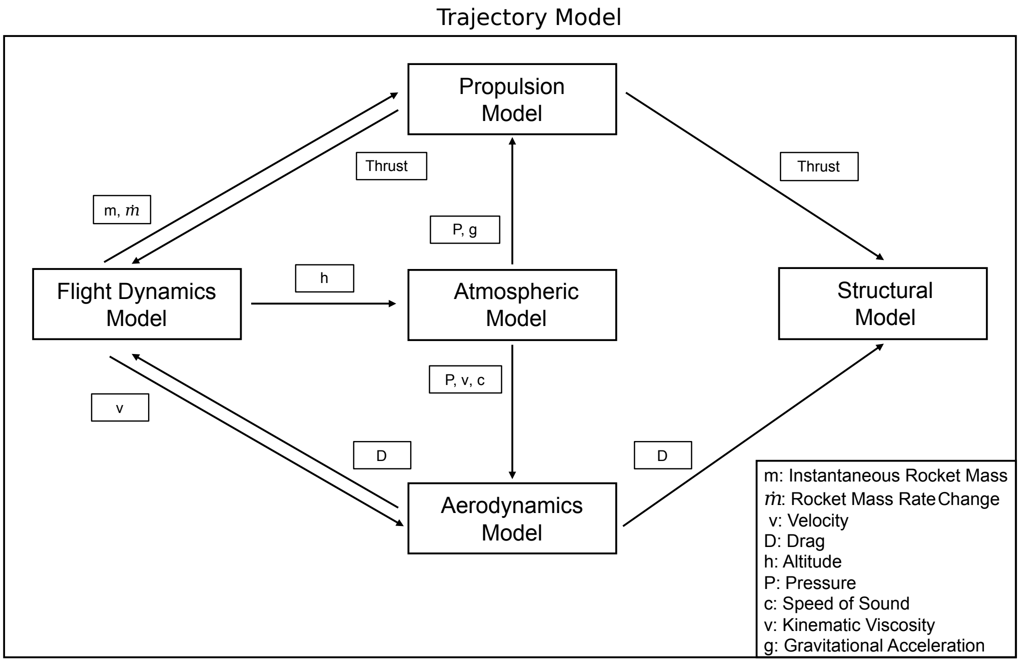


Figure 3: Overview of the trajectory model.

This model was then integrated in a top-level group with the mass and sizing model in order to create a framework capable of conducting an MDO process coupled with trajectory optimisation.

3 MULTIDISCIPLINARY DESIGN OPTIMISATION

Recent advances in technology have improved accessibility to higher computational power at gradually lower costs. Consequently, modern computer-based engineering systems capable of conducting complex MDO processes superseded the traditional concurrent engineering philosophy-based systems, where Disciplinary Design Optimisation (DDO) was conducted.

3.1 MDO Architectures

The main MDO architectures currently in use by the aerospace industry can be classified into two different groups: single-level (or monolithic) and multi-level (or distributed), according to the number of optimisers used in each architecture (single or multiple optimisers, respectively). The monolithic MDO architectures solve a single optimisation problem, while the distributed architectures decompose the original problem into a set of smaller optimisation subproblems which provide the exact same solution.

Single-level architectures are characterised by only using an optimiser in the top level of the multidisciplinary system, which is the governing level responsible for enforcing multidisciplinary feasibility [20].

The multidisciplinary feasible (MDF) architecture, solves the optimisation problem by implementing a system-level optimiser which calls a multidisciplinary analysis (MDA) responsible for solving all governing equations at the subsystem/component level until the coupling variables converge within the specified tolerance limits [21].

As an alternative approach, the individual discipline feasible (IDF) architecture adds additional independent variables to the problem to ensure that each discipline can be solved separately, while interdisciplinary equilibrium is maintained by a set of optimisation constraints that ensure the overall feasibility of the design once the optimisation convergence is achieved [21]. IDF potentially solves the high computational cost opened by the MDF architecture by conducting each discipline feasibility analysis independently and, in parallel, favouring speed and efficiency, at the cost of introducing additional variables and optimisation constraints, which increases the overall complexity of the original problem and might pose scalability issues for larger applications [22].

In contrast to the single-level architectures, multilevel architectures divide the original optimisation problem into a system-level optimisation problem and several sub-system level problems, according to the number of levels. The basic idea is for the system-level optimisation problem to coordinate the smaller sub-level problems, which in turn will be solved locally. The four most common architectures of this sort are: Collaborative Optimisation (CO), Concurrent SubSpace Optimisation (CSSO), Bi-Level Integrated System Synthesis and Analytical Target Cascading (ATC) [22].

After analysis, it was defined that the most suitable architecture for the developed framework was a single-level MDF architecture, as it is capable of solving the optimisation problem using a system-level optimiser that directly handles all the design variables and constraints, relaying on an MDA block to ensure multidisciplinary feasibility at each iteration, balancing simplicity in the hierarchical build of the design, efficiency of the data flows, and computational time.

3.2 Optimisation Algorithms

Optimisation algorithms are numerical methods designed to systematically search for the variable values that optimise the objective function [23]. They can be divided into two major groups: combinatorial (or discrete) or continuous, depending on whether the variables are discrete or continuous quantities, respectively. Discrete optimisation algorithms are hardly suitable for rocket design applications due to the continuous nature of the majority of the design variables involved, which typically represent physical properties (continuous in their essence) [23].

Continuous optimisation algorithms can be further divided into two other groups: linear and nonlinear.

Linear Programming (LP) algorithms are particularly designed for the minimisation (or maximisation) of a linear objective function subject to linear constraints.

Nonlinear Programming algorithms (NLP) are suitable for nonlinear, yet smooth, objective functions with at least continuous first partial derivatives in the target regions of the design space where the optimisation solution might be located [23]. By nature, the objective function, inequality, and equality constraints have a nonlinear behaviour in the rocket design environment with variables having quadratic, cubic, exponential or otherwise nonlinear relationship. Consequently, NLP algorithms need to be used in this work.

One of the most efficient methods for constrained nonlinear optimisation problems is Sequential Quadratic Programming (SQP), regarding function evaluations and computation cost [24]. Some of the most interesting characteristics are: Linear constraints and bounds remain satisfied; For n active constraints, SQP methods can achieve local convergence with quadratic convergence rate; Local convergence speed is superlinear; A large number of constraints can be treated by an active set strategy and the computation of gradients for inactive restrictions can be omitted.

In essence, Sequential Least Squares Quadratic Programming (SLSQP) is an optimisation method within the SQP wider family in which the constraints are linearized about the current point and a quadratic approximation of the objective function is defined [24].

Its formulation can be posed in standard form as

$$\min_{y \in \mathbb{R}^n} \quad f^k(y) \tag{23}$$

$$\text{subject to} \quad g^k(y) \leq 0, \tag{24}$$

where

$$f^k(y) = \frac{1}{2}(y - x_k)^T B_k (y - x_k) + \nabla f(x_k)^T (y - x_k) + f(x_k), \tag{25}$$

$$g_j^k(y) = \nabla g_j(x_k)^T (y - x_k) + g_j(x_k), \quad j = 1, \ldots, m. \tag{26}$$

Then, the Least Squares mathematical method is used to solve iteratively a set of Quadratic Programming subproblems, starting in a given vector of parameters, x^0, until a $(k+1)^{th}$ iterate, x^{k+1}, is reached in which the objective function converges within a specific tolerance condition, in compliance with all equality and inequality constraints [24].

In each iteration k, the optimiser needs to evaluate the function and constraint gradients, Δf and Δg, respectively, to determine a search direction d^k. Then, a line search is performed along that direction to find the step length α^k that minimises the $f(x)$, and a new iteration then follows at [24]:

$$x^{k+1} := x^k + \alpha^k d^k \ , \tag{27}$$

where d^k is the search direction within the k^{th} step and α^k is the step length.

3.3 Trajectory Optimisation

Trajectory optimisation problems are a part of the larger optimal control theory branch of mathematics, which specifically seeks to find the optimal control law of a dynamic system that satisfies a set of constraints while minimising a cost function.

A general mathematical problem definition can be defined as follows:

$$\begin{aligned}
\text{Optimal Trajectory:} \quad & \{x^*(t), u^*(t)\} && (28)\\
\text{System Dynamics:} \quad & \dot{x} = f(t, x, u) && (29)\\
\text{Constraints:} \quad & c_{\min} < c(t, x, u) < c_{\max} && (30)\\
\text{Boundary Conditions:} \quad & b_{\min} < b(t_0, x_0, t_f, x_f) \\
& < b_{\max} && (31)\\
\text{Cost Functional:} \quad & J = \phi(t_0, x_0, t_f, x_f) + \\
& \int_{t_0}^{t_f} g(t, x, u)\, dt && (32)
\end{aligned}$$

where x is for the state variables, u is for control variables, $f(t, x, u)$ are the system dynamics functions, $c_{\min}$, $c_{\max}$ and $c(t, x, u)$ are the lower, upper bounds and boundary function, respectively, $b_{\min}$, $b_{\max}$, $b(t_0, x_0, t_f, x_f)$ are the lower, upper bounds and boundary function, respectively, and, finally, J is the cost function.

3.3.1 Direct vs Indirect Collocation

Generally speaking, collocation methods belong to a broader transcription family of methods, in which differential equations governing the rocket system dynamics are enforced in a grid of points discretised from an initial continuous time interval, called collocation nodes, ensuring that the discretised approximations at these points are faithful to the continuous dynamics [25].

Collocation methods can be formulated in two different approaches: direct or indirect. Direct methods first discretise and then optimise while indirect methods optimise and then discretise [26], as illustrated in Fig. 4.

Indirect collocation methods, first establish the necessary and sufficient conditions for optimality, thus forming a Hamiltonian boundary-value problem (HBVP) which is analytically derived by applying the Pontryagin's Minimum Principle (PMP). Then, the newly created differential equations governing the adjoint variables, the control equation, and the boundary conditions form a new Two Point Boundary Value Problem (TPBVP).

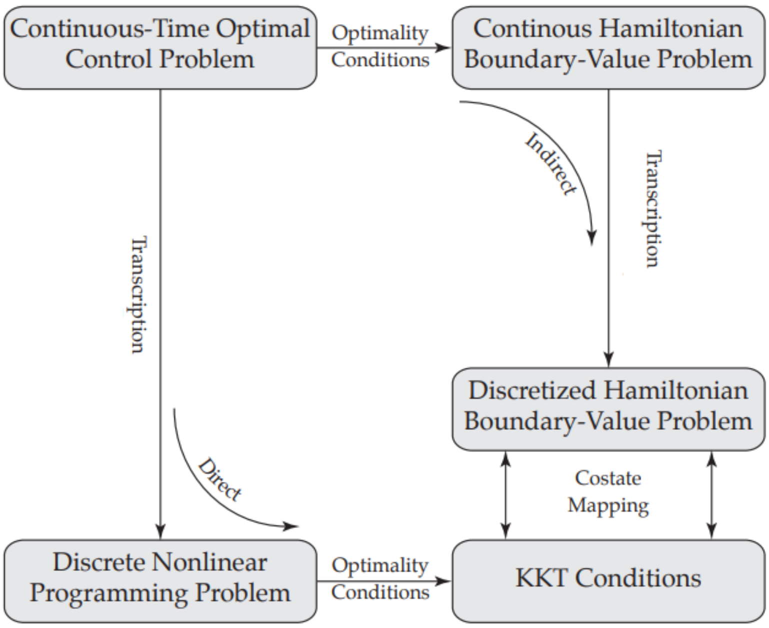


Figure 4: Comparison between direct and indirect collocation methods [27].

Then, TPBVP is discretised using a collocation method, such as, Hermite-Simpson, for example, transforming the continuous-time problem into a finite-dimensional nonlinear programming problem (NLP), which is numerically solved through the application of optimisation solvers, such as, gradient-based methods or sequential quadratic programming (SQP), until the Karush-Kuhn-Tucker (KKT) optimality conditions are met [28].

In contrast, direct collocation methods are the most used in the context of trajectory optimisation due to their simplicity, robustness, and range of application [28]. These methods are characterised by first discretising a continuous time interval into a grid of collocation points. Then, the state and control variables are also discretised at the collocation points, in which dynamics are enforced. Lastly, a nonlinear program is formulated from the discretised points and solved [28].

In comparison with the latter, indirect methods are commonly more accurate, providing stronger solutions with reliable error estimates due to analytically deriving the necessary and sufficient conditions in the early stages of the problem formulation, at the cost of requiring a better initialisation as they tend to have smaller convergence regions [26].

Therefore, at the preliminary design level, for a single-stage suborbital trajectory optimisation process, the direct collocation methods are the better choice because they have proven to be simpler, computationally faster, and accurate enough, while avoiding potential convergence issues for problems with increased complexity.

3.3.2 Pseudo-spectral Methods

Pseudo-spectral methods have gained traction in the trajectory optimisation field in recent years as a powerful, highly efficient alternative for the already well-established direct collocation methods to solve continuous nonlinear constrained optimal control problems with smooth functions, such as single-phase rocket trajectory optimisation problems. Highly complex applications of this method range from low-thrust orbit transfers, impulsive orbit transfers, ascent guidance, reentry trajectory design, spacecraft attitude control, among others [29].

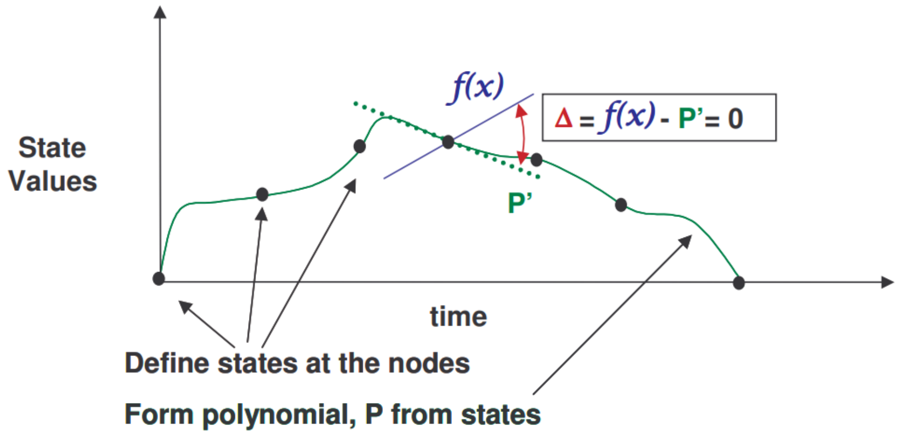


Figure 5: Pseudo-spectral procedure [30].

The basic idea behind a pseudospectral method is to build a high-order polynomial so that its time derivative values match the values of the system dynamics differential equations (state and control variable differential equations) at all collocation points across the entire time interval of the trajectory. By evaluating both the polynomial time derivatives and the physical time derivatives for a well-distributed representative number of discretisation nodes, it is possible to use numerical methods (Legendre-Gauss, Legendre-Gauss-Radau, Legendre-Gauss-Lobatto or Chebyshev-Gauss-Lobatto) to minimise the existing defects until a preset maximum tolerance limit is satisfied [28].

The major difference between direct collocation methods and pseudo-spectral resides in the fact that the first typically divides the trajectory into multiple segments and independently attempts to find a low-order polynomial that suits well with the system dynamics differential equations at the collocation points, facing the necessity of setting continuity constraints between segments and additional interior nodes within segments, whereas the latter is based on building a one segment high-order polynomial whose time derivatives match the system dynamics differential equations for all the collocation nodes, which suits well only for problems with smooth flight dynamics without significant function discontinuities [28].

Given that pseudospectral collocation methods are particularly powerful and highly efficient methods for solving continuous nonlinear constrained optimal control problems when compared to other direct collocation methods, these were the methods selected for the framework to solve the trajectory optimisation problem. Particularly, the high-order Gauss-Lobatto quadrature rules, as higher order polynomials offer an improved accuracy to the collocation method due to the finite precision, and, the number of parameters solved by the NLP problem is potentially lower in comparison to other lower order polynomials.

4 ROCKET DESIGN FRAMEWORK

4.1 MDO *Python* Libraries

In order to implement a multidisciplinary system for the current rocket design optimisation problem, it was necessary to search for an available software framework with the following characteristics:

- ability for handling with a system with multiple coupled disciplines integrated with trajectory optimisation;
- support a wide range of optimisers so that a suitable option can be chosen according to specific optimisation requirements of the problem;
- an open-source framework with proven capabilities to handle the optimisation problem at hand;
- a modular environment for easier model construction;
- and, lastly, a good metadata and data handling capabilities for less advanced and non database specialised users.

We selected the *OpenMDAO Python* library [31] for multidisciplinary optimisation integrated with the *Dymos Python* library [32] for the trajectory optimisation end of the problem, thus producing a coupled approach to the preliminary rocket design.

4.1.1 OpenMDAO

OpenMDAO is an open-source object-oriented software framework crafted for multidisciplinary design, analysis and optimisation applications, programmed mainly in the *Python* language (for scripting convenience) and completely capable of interacting with other compiled languages, such as, SWIG, Cython, C and C++, among others.

Since it was first introduced for NASA's next-generation advanced single-aisle civil transport project in 2008 at the NASA Glenn Research Center (based in Cleveland, USA) [33], it has been under continuous development with several compelling use cases across a wide range of applications: from a Cubesat MDO problem for maximised data download capabilities [34], to a low-order aerostructural wing optimisation [35], to a structural topology optimisation [36], etc.

4.1.2 Dymos

Dymos is an open-source software tool built on top of the OpenMDAO framework designed to solve optimal control problems, such as trajectory optimisation. The combination of a framework built from an OpenMDAO optimisation architecture integrated with a Dymos trajectory optimisation opens the possibility to solve co-design optimisation problems with high computational efficiency even for complex use cases. The proposed framework will allow the implementation of a static system model within each optimisation cycle (a mass and sizing model, for example), which will receive new design variable values from the optimiser, and then sends the rocket sizing as its outputs (for example, the length and mass of the rocket) to a trajectory group capable of conducting all the necessary dynamic calculations through Ordinary Differential Equations (ODE) or Differential-Algebraic Equations (DAE) [32]. A standard architecture of this framework is shown in Fig. 6

In terms of trajectory optimisation processes, Dymos allows for the implementation of direct transcription methods, particularly, pseudospectral (high-order Gauss-Lobatto and Radau) [32].

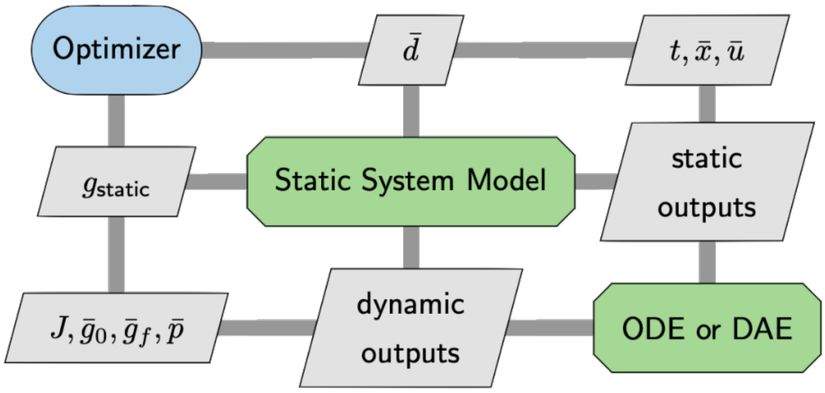


Figure 6: XDSM diagram of a standard coupled co-design problem, i.e., a MDO problem coupled with trajectory optimisation (OpenMDAO base framework integrated with Dymos) [32].

4.2 MDO Framework Implementation

In terms of the hierarchical structure of the framework, it was created a top-level group containing the optimiser and two other main groups: the mass and sizing group, which essentially is the mass and sizing model, and the trajectory group, which essentially is the trajectory model presented in Fig. 3. For each set of design variables x, directly handled by the optimiser, the mass and sizing generates a new rocket configuration, from which a few main parameters are fed within the trajectory model and a new objective function evaluation value is sent back to the optimiser.

At the trajectory level, the flight dynamics model handles four state variables (downrange x, altitude h, velocity v, pitch angle γ, and also their time derivatives, respectively, $\dot{x}$, $\dot{h}$, $\dot{v}$ and $\dot{\gamma}$. A remaining state variable time derivative, $\dot{m}$, is handled by the propulsion model.

These state variables are particularly important in the trajectory integration process because they mark the state values of the trajectory, i.e., the progress of the trajectory at each point in time, as well as the time progress of other models.

At the end of each trajectory simulation, the final altitude and the smallest difference between the critical stress of the body tube and the applied compressive stress are sent back to the optimiser, which does a constraint defect analysis, gradient evaluation, and a new iteration begins after a linesearch process.

Figure 7 illustrates the XDSM diagram of the MDO framework implementation.

5 ROCKET OPTIMAL DESIGN

5.1 Problem Formulation

In order to get a first assessment of the capabilities of the developed MDO framework in the context of a real problem, it is important to formally define it.

The chosen optimisation objective f was to minimise the rocket lift-off total mass subject to a minimum peak altitude constraint of 100 km, using the SLSQP optimisation method.

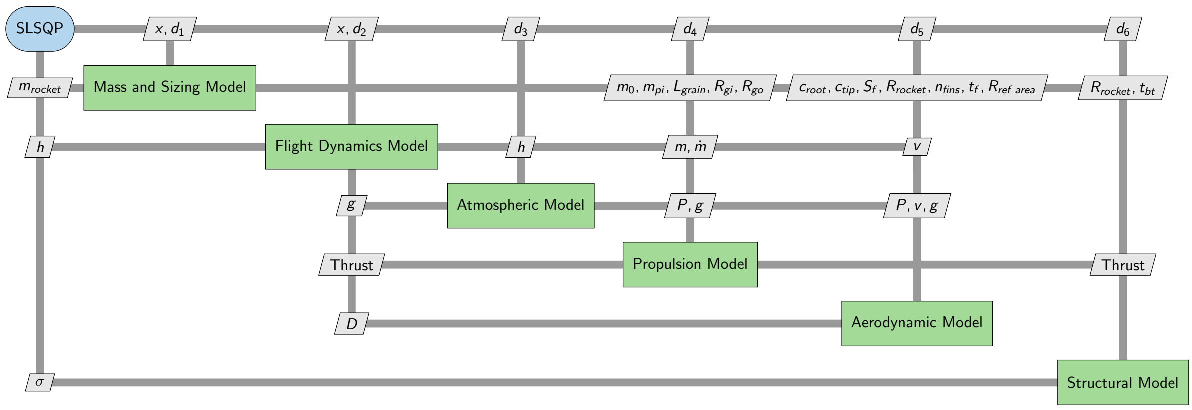


Figure 7: XDSM diagram of the framework highlighting the optimizer SLSQP (blue), the models (green) and design, coupled, local, and static variables (grey).

5.2 Parametric study of optimiser parameters

As the framework was thought for a quick preliminary rocket design application, it is of the greatest importance to use the optimisation setup which provides the most computationally cost-efficient solution. To that end, a parametric study on the impact of the optimiser tolerance level, as well as, the step size of the finite-difference method was conducted.

Generally speaking, it was observed that lower tolerance levels provide with more accurate results in higher computational costs. Reducing the tolerance levels from 10^{-3} to 10^{-6} while maintaining the step size, would require more function and gradient evaluations only to achieve slightly better results. Thus, the benefits from reducing the tolerance levels of the optimisation process were far outweighted by the increase in computational cost.

Regarding the impact of the step size on the optimisation process, it was observed that the system's convergence times were under 30 minutes for most cases although it could take several hours, depending on the quality of the initial design points and the chosen tolerance levels (using a computer operating Windows 10 with a 2.8 GHz, 4 cores processor, 16 GB of RAM and a 128 SSD storage unit).

This parametric study suggests that the best optimisation setting was to use a tolerance level of 10^{-3} combined with a step size of 10^{-3}, as this proved to be the most cost-efficient solution.

5.3 REXUS 2 Case Study

As a first case study, the optimisation goal was set to achieve the minimum lift-off mass for a suborbital flight with a minimum peak altitude of 100 km and a fixed payload of 98 kg, using 10 design variables, namely: the rocket diameter, overall length to diameter ratio, nosecone length to diameter ratio, body tube, SRM casing and propellant grain thicknesses, and the nozzle's expansion ratio, nozzle angle and nozzle convergent section angle. The main parameters of the found solution were then compared with several known masses and dimensions of the REXUS 2, an European suborbital single-staged solid propelled rocket [5], as portrayed in Table 1.

It was observed that these results represent an average percentual deviation of 26.6 % when compared to REXUS 2. Although a 27.7% decrease in total mass was observed,

Table 1: Comparison between the REXUS 2 and the optimised rocket configuration [5, 37].

Parameter	**Unit**	**REXUS 2**	**Optimised Rocket**	**Deviation**
Length	[m]	5.620	4.709	- 19.3%
Diameter	[m]	0.356	0.268	- 24.7%
Total Mass	[kg]	514.000	371.5	- 27.7%
Propellant Mass	[kg]	290.0	196.5	- 32.2%
Structural Mass	[kg]	126.0	77.0	- 38.9%
SRM Length	[m]	2.800	2.295	- 18.0%
Fin Root Chord	[m]	0.590	0.429	- 27.3%
Fin Tip Chord	[m]	0.400	0.300	- 25.0%

which is extremely positive, these results were treated with great caution as it is believed there might have been an oversimplification of the optimisation modules, as well as, underappreciation of several fixed masses (avionics and recovery system, for example). If adjusted to more realistic values, an increase in the propellant mass, structural mass and size of the optimised rocket would be observed, thus diminishing the percentage of total mass reduction.

In terms of the flight profile, a general agreement between both REXUS 2 and the optimised rocket was noticed, (please refer to Fig. 8) although the former sustains slightly higher altitudes in the first 100 seconds of the ascent phase which are evened out in the last 50 seconds. This flight profile discrepancies are the end result of the significant differences observed in terms of overall weight, size, and performance (velocity, drag, and thrust profiles, for example).

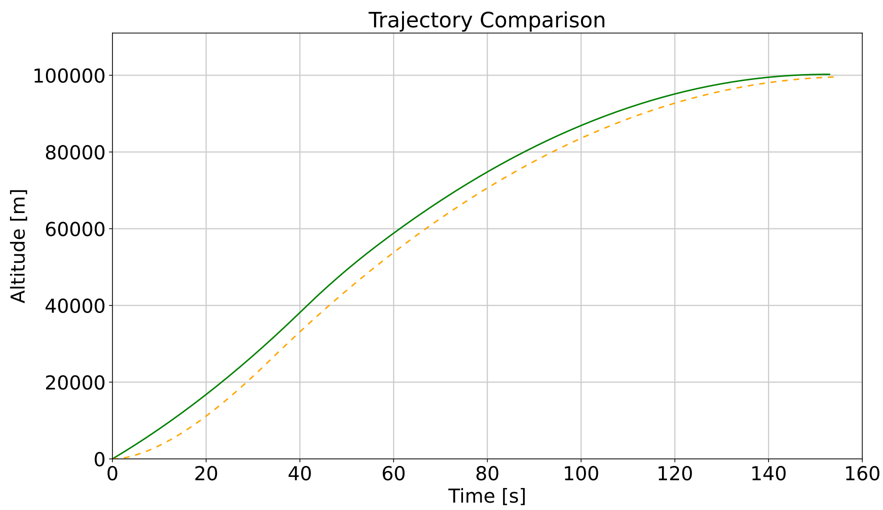


Figure 8: Flight profile comparison between both rockets. The green line represents the Rexus 2 flight profile and the orange dashed line the flight profile of the optimised rocket.

Finally, a 2D visual comparison of both rockets is presented in Fig. 9 where a few known measurements of the Rexus 2 are compared with the optimised rocket.

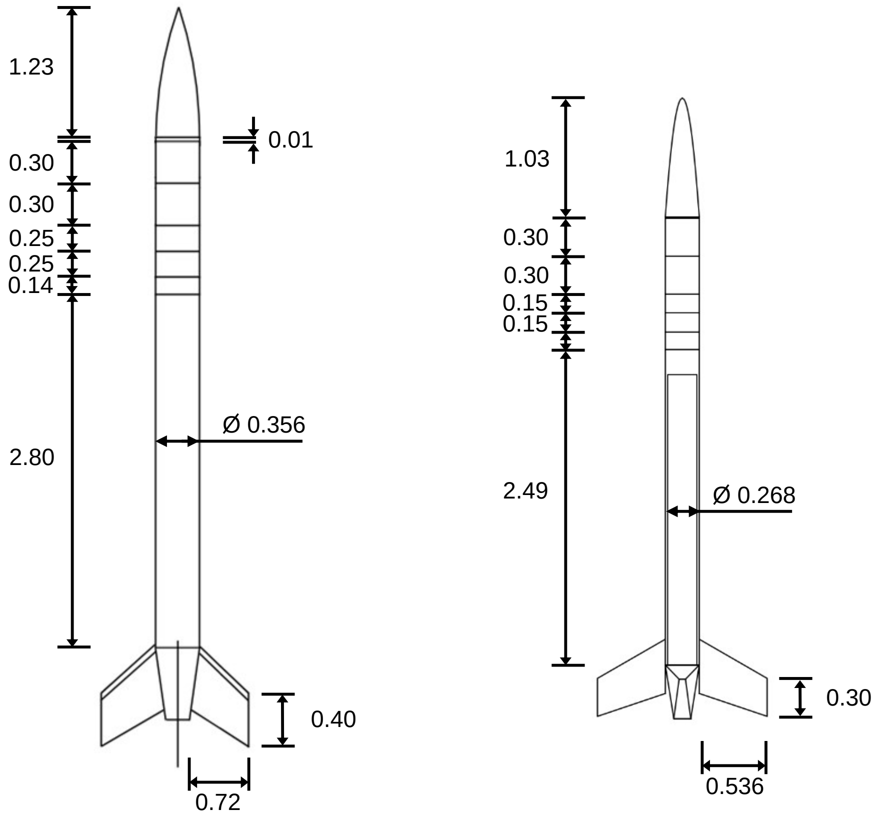


Figure 9: Visual comparison of the Rexus 2 (left) and the optimised rocket (right) highlighting the measurements of a few known parameters.

5.4 Payload and Altitude Sensitivity Analysis

In the Section 5.3, the optimisation behaviour of the framework was tested by comparing the obtained optimisation solution with a real rocket (Rexus 2). Ten design variables, an optimisation goal and specific mission requirements matching with those of the Rexus 2 were defined.

In this section, a sensitivity analysis of two parameters, payload and altitude, will be conducted in order to assess their impact in rocket design. Figure 10 shows how the payload and altitude were varied and four additional optimisation problems were studied.

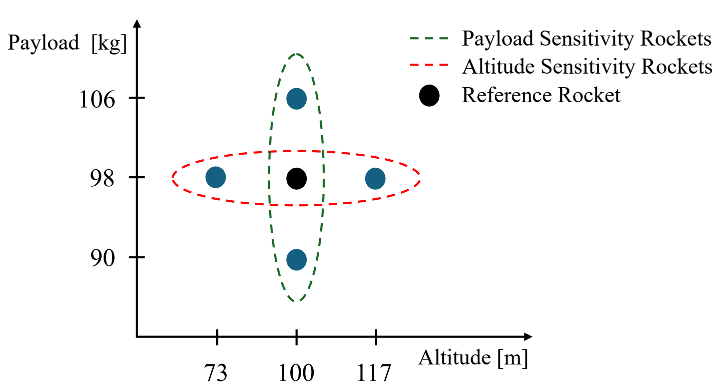


Figure 10: Payload and altitude sensitivity analysis rockets.

5.4.1 Altitude Sensitivity Analysis

In order to assess the altitude sensitivity, two new optimisation problems were created with different minimum peak altitudes, 73 and 117 km while maintaining a constant payload mass of 98 kg. The obtained solutions were then compared with the optimised rocket of the previous case study, which now served as a reference. Table 3 portrays a comparison of a few main parameters of these rockets:

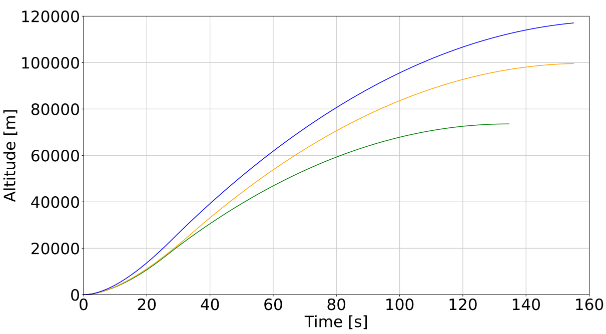
Table 2: Comparison between the altitude sensitivity optimised rockets.

Parameter	Unit	73 km	Deviation	100 km	Deviation	117 km
Length	[m]	4.647	- 1.3 %	4.709	+ 3.3 %	4.868
Diameter	[m]	0.290	+ 8.2 %	0.268	- 2.6 %	0.261
Total Mass	[kg]	353.2	- 4.9 %	371.5	+ 3.6 %	384.8
Payload Mass	[kg]	98.0	0.0 %	98.0	0.0 %	98.0
Propellant Mass	[kg]	178.2	- 9.3 %	196.5	+ 5.0 %	207.1
Structural Mass	[kg]	76.9	- 7.7 %	77.0	+ 3.4 %	79.6
SRM Length	[m]	2.030	-11.6 %	2.295	+ 12.6 %	2.584

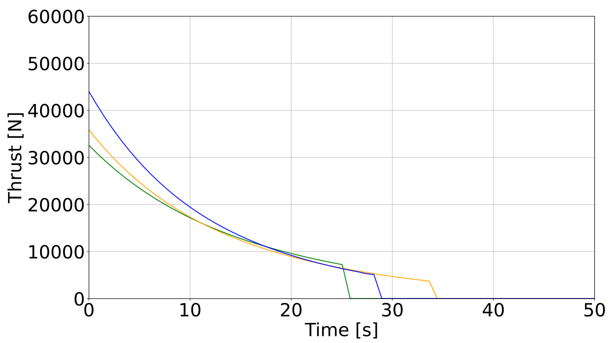
From Table 3, it can be seen that by reducing the minimum peak altitude from 100 km to 73 km while maintaining the same payload, it was possible to reduce the overall mass of the rocket in 5.0 %, the propellant mass in 9.3 %, and also, the structural mass in 7.7 %. On the other hand, increasing the target altitude to 117 km, demanded increasing the propellant mass in 5 %, the structural mass in 3.4 %, and consequently, the total mass in 3.6 %.

Furthermore, this data confirms the patterns expected from a purely theoretical analysis. As higher target altitudes are set, the propellant mass needed increases, thus increasing the SRM length and the overall rocket length. Additionally, the loading on the structure also tends to increase which leads to an increase in the structural mass of the rocket and, ultimately, an increase in the total lift-off mass.

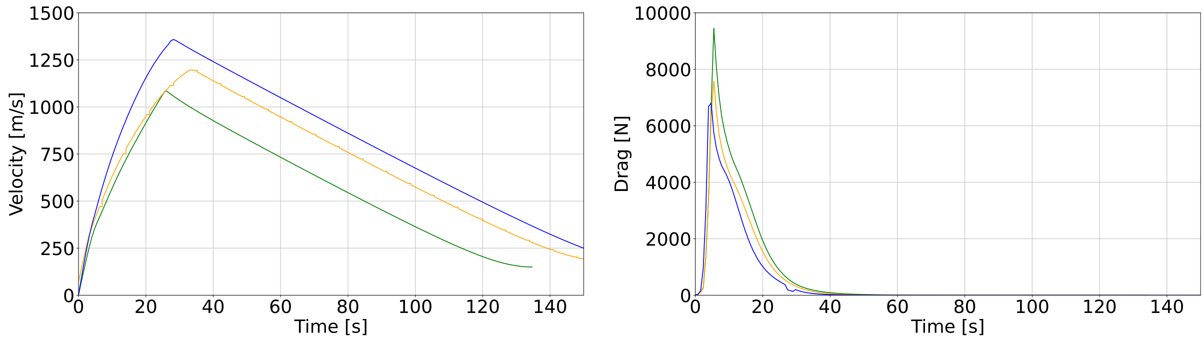
In terms of the flight profile comparison, it was possible to observe that in order to achieve higher altitudes the thrust profile increases in magnitude, as higher velocities are required.



(a) Altitude profile comparison.



(b) Thrust profile comparison.



(c) Velocity profile comparison. (d) Drag profile comparison.

Figure 11: Flight data analysis of four main parameters: altitude, thrust, velocity and drag. The green, orange and blue lines represent the rockets carrying 98 kg of payload aimed at 73, 100 and 117 km minimum peak altitudes, respectively

On the other hand, under increased velocities, the effects of the aerodynamic loading on the structure, namely, the drag become ever more prominent, for which the optimiser tends to reduce the cross-sectional area of the rocket as much as possible by reducing its diameter.

5.4.2 Payload Sensitivity Analysis

In order to assess the payload sensitivity, another two new optimisation problems were created with different payloads, 90 and 106 kg while maintaining a constant minimum peak altitude, fixed at 100 km. Similarly to the altitude sensitivity analysis, the obtained solutions were then compared with the optimised rocket from the first case study, which served as a reference. Table 3 presents a comparison of a few main parameters of these rockets.

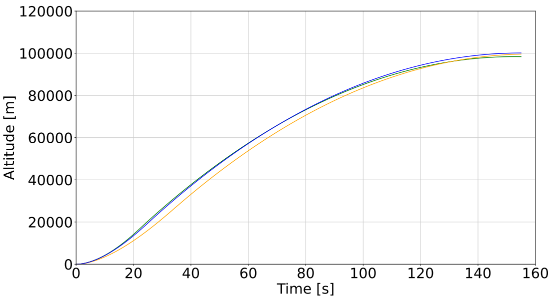
Table 3: Comparison between the payload sensitivity optimised rockets.

Parameter	**Unit**	**90 kg**	**Deviation**	**98 kg**	**Deviation**	**106 kg**
Length	[m]	4.689	- 0.4 %	4.709	+ 2.709 %	4.846
Diameter	[m]	0.261	-2.6 %	0.268	- 2.2 %	0.262
Total Mass	[kg]	347.6	- 6.4 %	371.5	+ 4.8 %	389.5
Payload Mass	[kg]	90.0	0.0 %	98.0	0.0 %	106.0
Propellant Mass	[kg]	180.7	- 8.1 %	196.5	+ 3.3 %	203.1
Structural Mass	[kg]	76.9	- 0.1 %	77.0	+ 4.4 %	80.4
SRM Length	[m]	2.423	+ 5.6 %	2.295	+ 11.7 %	2.563

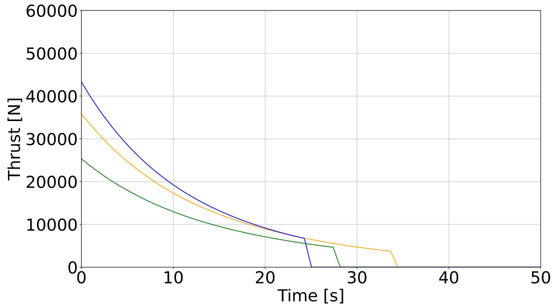
This results show that a payload reduction of 8 kg translates in 6.4 % decrease in total mass while a payload increase in the same amount translates in an increase of 4.8 %. The observed variations in total mass are the end result of the partial variations of propellant mass and structural mass.

Looking at the flight data, presented in Fig. 12, it is possible to observe significant differences in the thrust and velocity profiles. As higher payload is carried, more propellant is needed in order to achieve high enough velocities in the early stages of the ascent phase to free the rocket from the atmosphere and reach the desired target altitude.

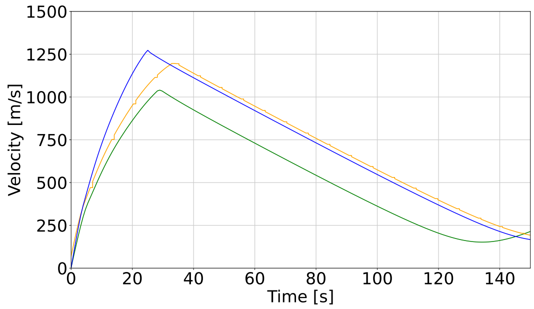
Furthermore, higher velocities involve higher accelerations, higher propellant burn rates and, consequently, higher thrusts which lead to the design of rockets with increased thicknesses and, thus, increased structural mass in order to withstand higher aerodynamic loading conditions.



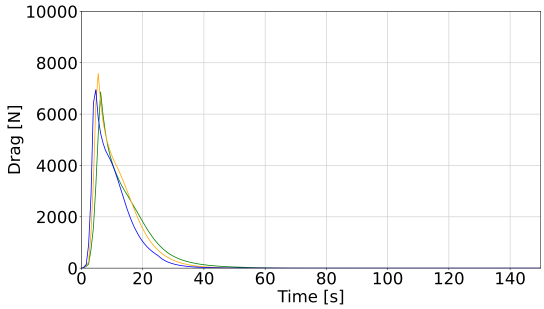
(a) Altitude profile comparison.



(b) Thrust profile comparison.



(c) Velocity profile comparison.



(d) Drag profile comparison.

Figure 12: Flight data analysis of four main parameters: altitude, thrust, velocity and drag. The green, orange and blue lines represent the rockets aimed at 100 km minimum peak altitude carrying payload of 90, 98 and 106 kgs, respectively.

Finally, a 2D visual comparison of all the optimised rockets obtained in this sensitivity analysis is presented in Fig. 13.

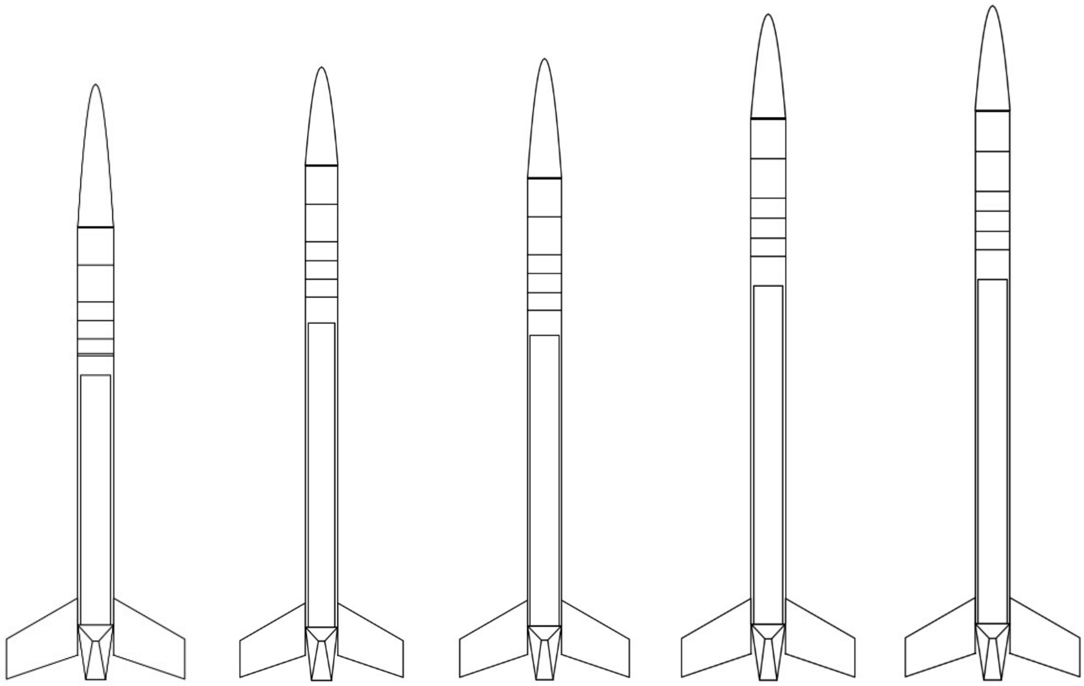


Figure 13: Rocket 2D final comparison.

6 CONCLUSIONS

In this work, a low computational cost multidisciplinary optimisation (MDO) framework capable of solving co-design optimization problems in the context of preliminary design of single-stage solid propellant rockets was developed. Six disciplinary models were successfully developed and integrated within an MDF architecture. As for the optimizer, a gradient-based SLSQP optimization algorithm was selected and successfully integrated in the framework. In addition, the Gauss-Lobatto pseudospectral method was selected to solve the trajectory optimisation problem.

The developed framework underwent several tests in order to assess its optimisation capabilities. First, parametric studies of two main optimisation parameters, the tolerance level of the SLSQP method and the step size of the finite difference, were conducted. From this tests, it was concluded that the best setup for optimisation was using a tolerance level of 10^{-3} and a step size of the finite difference method of 10^{-3}.

An initial case study was conducted to assess the accuracy of the framework against a real rocket. The results showed a 27.7 % total mass reduction which might indicate that the optimisation modules might have been oversimplified and some fixed masses, such as, avionics, recovery system, among others, might have been underappreciated. If adjusted a smaller, more realistic total mass reduction is expected.

Afterwards, a sensitivity analysis allowed to conclude that the payload and minimum altitude greatly influence the behaviour of the optimisation process, with the results showing that reducing the minimum peak altitude from 100 km to 73 km allows for a 4.9 % total mass reduction and increasing it from 100 km to 117 km leads to a 3.6 % increase.

Furthermore, it was observed that varying the payload mass from 98 kg to 90 kg allowed for a 6.4 % total mass reduction against a 4.8 % increase when varied from 98 kg to 106 kg.

Overall, the developed framework shows good signs of being capable of performing the design optimisation of a single stage sounding rocket at a preliminary level. Given its great modularity, a straightforward extension to a larger spectrum of applications is expected, such as multi-stage or liquid-propellant, upon additional development.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the full support of IDMEC and also Academia da Força Aérea throughout the development of this work. A. C. Marta and P. J. S. Gil acknowledge Fundação para a Ciência e a Tecnologia (FCT) for its financial support via the project LAETA Base Funding (DOI: 10.5449/UIDB/50022/2020).

REFERENCES

[1] G. Genta. Private space exploration: A new way for starting a spacefaring society? *Acta Astronautica*, 104(2):480–486, nov 2014. ISSN 0094-5765. DOI:10.1016/j.actaastro.2014.04.008.

[2] S. F. E. Team. Space Foundation Releases the space report 2023 Q2, showing annual growth of global space economy to $546B, 2023. Available: `https://www.spacefoundation.org/2023/07/25/the-space-report-2023-q2/`.

[3] G. Gurney. *Space Technology Spinoffs.* An Impact book. Watts, 1979. ISBN 9780531022900.

[4] A. Papageorgiou and J. Olvander. The role of multidisciplinary design optimization (MDO) in the development process of complex engineering products. In *Proceedings of the 21st International Conference on Engineering Design (ICED17)*, volume 4, 2017.

[5] M. Persson, O. & Hörschgen. REXUS 2 - the first EuroLaunch project. In *17th ESA Symposium on European Rocket and Balloon Programmes and Related Research*, pages 389 – 393, Sandefjord, Norway, 30 May - 2 June 2005. Swedish Space Corporation (SSC) & Deutsches Zentrum für Luft- und Raumfahrt (DLR), ESA Publications Division. ISBN 92-9092-901-4.

[6] B. C. M. H. H. Box, S. Estimating the dynamic and aerodynamic parameters of passively controlled high power rockets for flight simulation. *Cambridge Rocketry*, 2009. Available: `https://cambridgerocket.sourceforge.net/AerodynamicCoefficients.pdf`.

[7] L. Pepermans, E. Menting, M. Rozemeijer, B. Koops, N. S. Dahl, N. Suard, S. khurana, F. V. Marion, F. Kuhnert, and M. Serman. Comparison of Various Parachute Deployment Systems for Full Rocket Recovery of Sounding Rockets. In *Proceedings of the 8th European Conference for Aeronautics and Space Sciences. Madrid, Spain, 1-4 july 2019*, 2019. DOI:10.13009/EUCASS2019-411.

[8] J. M. Tizó. Burnback Analysis of Solid Propellant Rocket Motors. *arXiv preprint*, 2023. DOI:10.48550/ARXIV.2301.04559.

[9] R. Hartfield, R. Jenkins, J. Burkhalter, and W. Foster. A Review of Analytical Methods for Solid Rocket Motor Grain Analysis. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2003. DOI:10.2514/6.2003-4506.

[10] W. Miller. *Solid Rocket Motor Performance Analysis and Prediction.* NASA SP 8039. National Aeronautics and Space Administration, May 1971.

[11] Brooks, W.T. *Solid Propellant Grain Design and Internal Ballistics.* NASA SP 8076. National Aeronautics and Space Administration, March 1972.

[12] G. Henry, R. Humble, W. Larson, U. S. D. of Defense, U. S. N. Aeronautics, and S. Administration. *Space Propulsion Analysis and Design.* College custom series. McGraw-Hill, 1995. ISBN 9780070313293.

[13] K. Kuo and M. Summerfield. *Fundamentals of Solid-propellant Combustion.* Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, 1984. ISBN 9780915928842.

[14] D. J. Martin. Summary Of Flutter Experiences As A Guide To The Preliminary Design of Lifting Surfaces On Missiles. Report No. 685, NACA, Langley Aeronautical Laboratory Langley Field, Va, 1940.

[15] United States. National Oceanic and Atmospheric Administration and United States Committee on Extension to the Standard Atmosphere. *U.S. Standard Atmosphere, 1976.* NOAA - SIT 76-1562. National Oceanic and Amospheric Administration, 1976.

[16] United States Committee on Extension to the Standard Atmosphere and United States. National Aeronautics and Space Administration. *U.S. Standard Atmosphere, 1962: ICAO Standard Atmosphere to 20 Kilometers; Proposed ICAO Extension to 32 Kilometers; Tables and Data to 700 Kilometers.* U.S. Government Printing Office, 1962.

[17] H. Akima. A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *Journal of the ACM*, 17(4):589–602, Oct. 1970. ISSN 1557-735X. DOI:10.1145/321607.321609.

[18] Akima1DInterpolator - SciPy v1.14.1 Manual. Available: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.Akima1DInterpolator.html`.

[19] P. Sforza. *Theory of Aerospace Propulsion.* Aerospace Engineering. Elsevier Science, 2016. ISBN 9780128096017. DOI:10.1016/c2009-0-61051-5.

[20] P. Dépincé, B. Guédas, and J. Picard. Multidisciplinary and multiobjective optimization: Comparison of several methodss. In *7th World Congress on Structural and Multidisciplinary Optimization)*, 2007.

[21] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization*, 4 (4):754–776, Nov. 1994. DOI:10.1137/0804044.

[22] N. Tedford and J. R. R. A. Martins. On the Common Structure of MDO Problems: A Comparison of Architectures. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, June 2006. DOI:10.2514/6.2006-7080.

[23] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN 9780387400655.

[24] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.

[25] M. P. Kelly. Transcription Methods for Trajectory Optimization: a beginners tutorial, 2017. DOI:10.48550/arxiv.1707.00284.

[26] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373, Dec. 1992. ISSN 1572-9338. DOI:10.1007/bf02071065.

[27] A. Rao, D. Benson, G. Huntington, B. Origin, L. Seattle, C. Wa, C. Francolin, M. Darby, and M. Patterson. User's Manual for GPOPS Version 2.1: A MATLAB R Package for Dynamic Optimization Using the Gauss Pseudospectral Method. *Gainesville, FL, USA*, 2009. Available: `https://gpops2.com/resources/gpops2UsersGuide.pdf`.

[28] F. Topputo and C. Zhang. Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications. *Abstract and Applied Analysis*, 2014: 1–15, 2014. ISSN 1687-0409. DOI:10.1155/2014/851720.

[29] I. M. Ross and F. Fahroo. Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems. *Journal of Guidance, Control, and Dynamics*, 27(3): 397–405, May 2004. ISSN 1533-3884. DOI:10.2514/1.3426.

[30] S. Paris, J. Riehl, and W. Sjauw. Enhanced procedures for direct trajectory optimization using nonlinear programming and implicit integration. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2006. DOI:10.2514/6.2006-6309.

[31] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, Mar. 2019. ISSN 1615-1488. DOI:10.1007/s00158-019-02211-z.

[32] R. Falck, J. Gray, K. Ponnapalli, and T. Wright. Dymos: A Python package for optimal control of multidisciplinary systems. *Journal of Open Source Software*, 6 (59):2809, Mar. 2021. ISSN 2475-9066. DOI:10.21105/joss.02809.

[33] C. Heath and J. Gray. OpenMDAO: Framework for Flexible Multidisciplinary Design, Analysis and Optimization Methods. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 2012*, 04 2012. ISBN 978-1-60086-937-2. DOI:10.2514/6.2012-1673.

[34] J. T. Hwang, D. Y. Lee, J. W. Cutler, and J. R. R. A. Martins. Large-Scale Multidisciplinary Optimization of a Small Satellite's Design and Operation. *Journal of Spacecraft and Rockets*, 51(5):1648–1663, Sept. 2014. ISSN 1533-6794. DOI:10.2514/1.a32751.

[35] J. P. Jasa, J. T. Hwang, and J. R. R. A. Martins. Open-source coupled aerostructural optimization using Python. *Structural and Multidisciplinary Optimization*, 57(4): 1815–1827, Feb. 2018. ISSN 1615-1488. DOI:10.1007/s00158-018-1912-8.

[36] H. Chung, J. T. Hwang, J. S. Gray, and H. A. Kim. Implementation of topology optimization using openMDAO. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, Jan. 2018. DOI:10.2514/6.2018-0653.

[37] M. S. *Rexus User Manual: Rocket Experiments for University Students*. Number 7.11. EuroLaunch, JAN 2014. Available: `https://www.zarm.uni-bremen.de/fileadmin/user_upload/teaching/study_programs/rexus_bexus/RX_UserManual_v7-11_08Jan14.pdf`.