Sounding Rocket Multidisciplinary Preliminary Design and Trajectory Optimisation

Alexandre Mendes Palaio
alexandre.m.palaio@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Portugal

December 2024

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, studies with up to 10 geometric design variables showed that the latter were capable of achieving the best results, as expected. Then, sensitivity studies of the payload and the minimum altitude confirmed that the rocket sizing is greatly impacted by both. Lastly, comparisons with real rockets, namely, the REXUS 2 and REXUS 10, showed very good agreement, achieving a total mass reduction of 14.5% and 14.9%, respectively. Given the great modularity of the framework, a straightforward extension to other types of rockets, such as multi-stage or liquid-propellant, is expected upon additional development.

Keywords: MDO, Trajectory Optimization, 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 Vehicles (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 44 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.

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. To the C_d profile, a compressible flow correction was applied for better accuracy under compressible flow regimes. Additionally, a recovery system contribution was also integrated in the model to simulate the behaviour of the parachute deployment during the descent phase of the flight profile.

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.

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.

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 [6].

2.4. Structures

After briefly analysing two key structural events, buckling and fin flutter, a model was created to assess the structural integrity of the rocket along the flight profile. First, given a pair of drag and thrust forces, the model calculates the resulting compressive loading at the body tube cross-sectional area. Then, it compares it with the critical buckling stress in order to evaluate if, at any moment in time, the rocket was subject to such a loading condition for buckling to occur.

Additionally, 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.

2.5. Atmosphere

An atmospheric model adapted from the *OpenAeroStruct Python* library was also developed to provide with the main atmospheric properties throughout the trajectory. Arrays with the values of each atmospheric parameter, retrieved from the standard atmosphere convention tables [7, 8], were first created.

By interpolating the altitude value (model input) using the *Akima1DInterpolator* class imported from the *Scipy Python* library [9, 10], 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.

2.6. Flight Dynamics

A set of flight dynamics equations was developed, capable of translating the complex interactions between the rocket, the atmosphere, and any other external factors with active influence on the rocket [11].

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

Thus, the flight dynamics of the rocket, graphically represented in Fig. 1 can be reduced to the following set of equations:

$$\dot{V} = \frac{T}{m}\cos\alpha - \frac{D}{m} - g\sin\gamma, \tag{1}$$

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

$$\dot{x} = V\cos\gamma, \tag{3}$$

$$\dot{h} = V\sin\gamma. \tag{4}$$

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 [11].

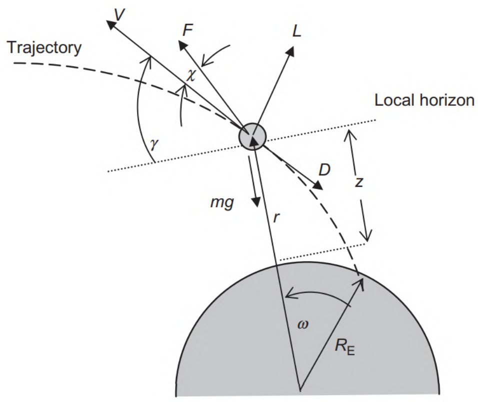

Figure 1: Flight Dynamics state variables and acting forces [11].

2.7. Trajectory

In order to implement a trajectory model capable of reproducing the specific characteristics of the four flight stages aforementioned, a high-level group was created, with 5 coupled models within, namely, Flight Dynamics, Atmospheric, Propulsion, Aerodynamics and Structural.

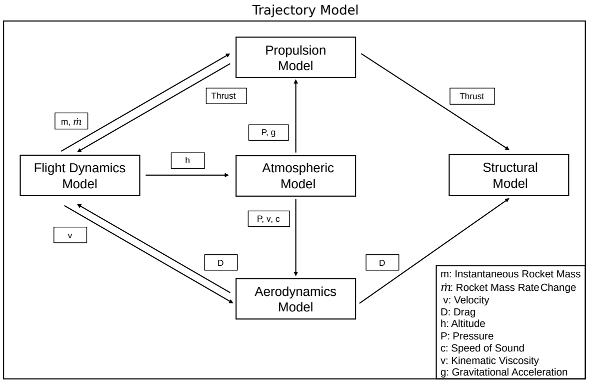

Figure 2: Overview of the trajectory model.

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 advancements 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 to ensure multidisciplinary feasibility [12].

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 [13].

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 which ensure the overall feasibility of the design once the optimisation convergence is achieved [13]. 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 [14], nonetheless.

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).

After a thorough 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 a 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 and, most importantly, with guaranteed feasibility,

3.2. Optimisation Algorithms

Optimisation algorithms are numerical methods designed to systematically search for the variable values which optimise the objective function [15]. 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) [15]. Therefore, these will not be focused.

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

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 on the solution target regions of the design space [15]. 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 [16]. 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; and, 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 [16].

Its formulation can be posed in standard form as

$$\min_{y \in \mathbb{R}^n} \quad f^k(y) \tag{5}$$

$$\text{subject to} \quad g^k(y) \leq 0, \tag{6}$$

where

$$f^k(y) = \frac{1}{2}(y - x_k)^T B_k (y - x_k) + \nabla f(x_k)^T (y - x_k) + f(x_k), \tag{7}$$

$$g_j^k(y) = \nabla g_j(x_k)^T (y - x_k) + g_j(x_k), \quad j = 1, \ldots, m. \tag{8}$$

Then, the Least Squares mathematical method is used to solve iteratively a set of Quadratic Programming subproblems, starting in 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 [16].

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 [16]:

$$x^{k+1} := x^k + \alpha^k d^k \ , \tag{9}$$

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:

$$\text{Optimal Trajectory:} \quad \{x^*(t), u^*(t)\} \tag{10}$$

$$\text{System Dynamics:} \quad \dot{x} = f(t, x, u) \tag{11}$$

$$\text{Constraints:} \quad c_{\text{min}} < c(t, x, u) < c_{\text{max}} \tag{12}$$

$$\text{Boundary Conditions:} \quad b_{\text{min}} < b(t_0, x_0, t_f, x_f) < b_{\text{max}} \tag{13}$$

$$\text{Cost Functional:} \quad J = \phi(t_0, x_0, t_f, x_f) + \int_{t_0}^{t_f} g(t, x, u)\, dt \tag{14}$$

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 [17].

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 [18], as illustrated in Fig. 3.

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). 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

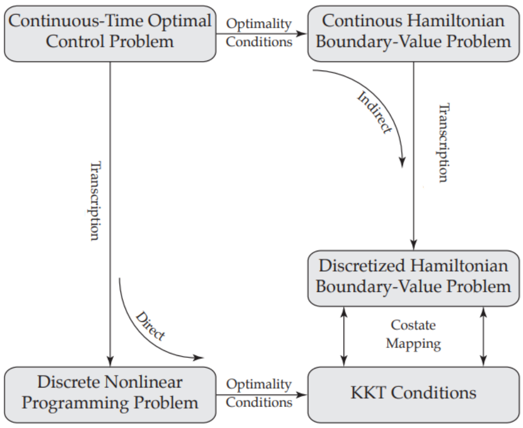

Figure 3: Comparison between direct and indirect collocation methods [19].

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 [20].

In contrast, direct collocation methods are the most used in the context of trajectory optimisation due to their simplicity, robustness, and range of application [20]. 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 [20].

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 [18].

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 [21].

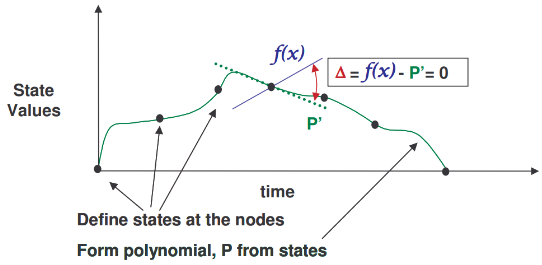

Figure 4: Pseudo-spectral procedure.

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 [20].

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 [20].

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 proof of 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 non database specialised users.

After careful consideration, it was defined that the framework under development was to be implemented using the *OpenMDAO Python* library [22] for the multidisciplinary optimisation integrated with the *Dymos Python* library [23] 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) [24], 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 [25], to a low-order aerostructural wing optimisation [26], to a structural topology optimisation [27], 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 a OpenMDAO optimisation architecture integrated width 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 desig variable values from 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) [23]. A standard architecture of this framework is shown in Fig. 5

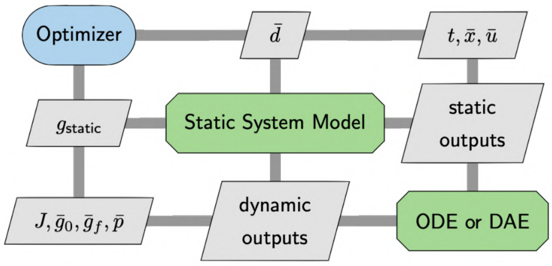

Figure 5: XDSM diagram of a standard coupled co-design problem, i.e., a MDO problem coupled with trajectory optimisation (OpenMDAO base framework integrated with Dymos) [23].

In terms of trajectory optimisation processes, Dymos allows for the implementation of direct transcription methods, particularly, pseudospectral (high-order Gauss-Lobatto and Radau) [23].

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. 2. 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 6 illustrates the XDSM diagram of the

MDO framework implementation. It is possible to observe that the design variables, x, coupled variables, local variables, static variables (constants), d, constraint variables, h and σ, all the six developed models, as well as, the optimiser, SLSQP.

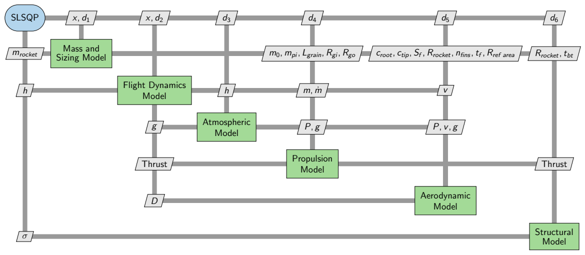

Figure 6: XDSM diagram of the framework highlighting the optimizer SLSQP (blue), the models (green) and design, coupled, local, and static variables (grey).

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.

5.2. Parametric study of optimiser parameters

As the framework was thought for a quick preliminary rocket design application, it is of the utmost importance to use the best setup configuration to obtain the most computationally cost-efficient behaviour from the optimiser. To that end, a parametric study on the impact of the optimiser tolerance, as well as, the step size of the finite-difference gradient approximations was conducted.

As expected, generally speaking, it was observed that lower tolerance levels provide with more accurate results at a higher computational cost, as expected.

Regarding the impact of the step size in the optimisation process results, it was observed that the system convergence times were well under 10 minutes for most cases, across different initial guess points and tolerance levels. In contrast, it was observed that for lower tolerances with a 10^{-3} step size, significantly less function and gradient evaluations were required to achieve slightly better results.

It was concluded that the best optimisation setting is to use a tolerance of 10^{-5} with a step size of 10^{-3}, as this is the one which provided with the lowest objective function evaluation which complied with the imposed altitude constraint.

5.3. Benchmark Case Study

In this first case study, the main parameters of the optimised rocket were compared with several known masses and dimensions of the REXUS 2, using only one design variable, the rocket diameter D_{rocket}. The obtained results are presented in Table 1.

Table 1: Comparison between the REXUS 2 and the optimised rocket configuration [5, 28].

Parameter	Unit	REXUS 2	Optimised Rocket	Deviation
Length	[m]	5.620	5.822	+ 3.4%
Diameter	[m]	0.3560	0.364	+ 2.2%
Total Mass	[kg]	514.000	501.768	- 2.4%
Propellant Mass	[kg]	290.000	282.610	- 2.5%
Structural Mass	[kg]	126.000	121.158	- 3.8%
SRM Length	[m]	2.800	2.895	+ 3.4%
Fin Root Chord	[m]	0.590	0.582	- 1.4%
Fin Tip Chord	[m]	0.400	0.408	+ 2.0%

After analysis, it can be observed that the results align well with the REXUS 2 with an average relative deviation of 2.64% was and a total mass reduction of 2.4%.

In terms of the flight profile, a general agreement between both REXUS 2 and the optimised rocket was noticed, with some discrepancies due to the fact that the flight profile of the REXUS 2 was only a prediction done in preparation for the mission itself and so it did not take into account atmospheric and other real conditions effects. Another contributing factor was the significant differences in the thrust profiles from both rockets. Figure 8 shows both thrust-profiles, highlighting the discrepancy between the engine burnout times.

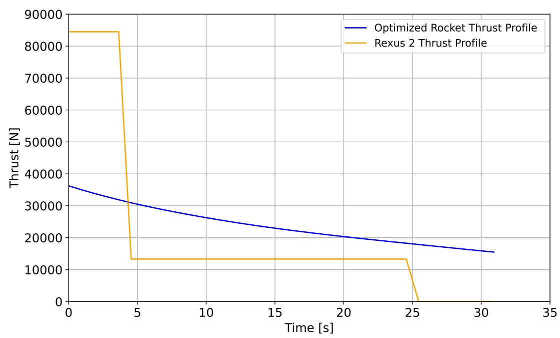

Figure 7: Thrust profile comparison of the Rexus 2 and the optimised rocket.

Finally, a visual comparison of both rockets is presented in Figure 8.

5.4. Multivariable Case Study

A multivariable case study then followed, with significant improvements being observed in the optimisation behaviour of the framework. By setting 10 design variables the optimisation capabilities of the framework improved as the optimizer had control over more variables and a larger design space to work with, thus naturally achieving substantially better results, as expected. Table 2 portrays a compar-

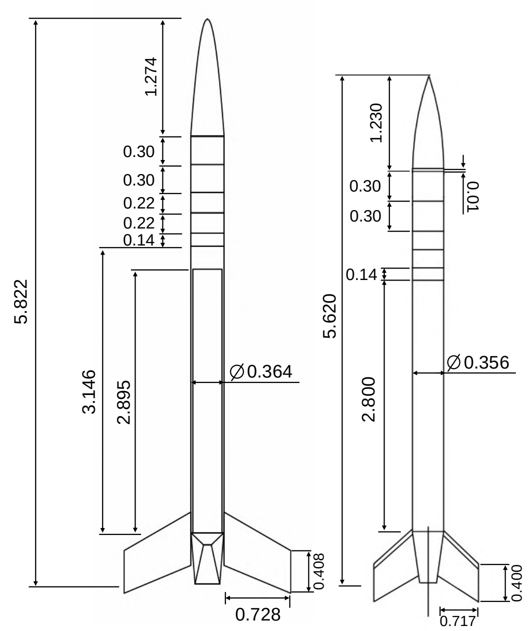

Figure 8: Visual comparison of the Rexus 2 and the optimised rocket comparison.

ison between the REXUS 2 and the new optimised rocket.

Table 2: Comparison between the REXUS 2 and the multivariable optimised rocket configuration [5, 28].

Parameter	Unit	REXUS 2	Optimised Rocket	Deviation
Length	**[m]**	5.620	5.600	-0.4%
Diameter	**[m]**	0.356	0.350	-1.7%
Total Mass	**[kg]**	514.000	439.530	-14.5%
Payload Mass	**[kg]**	98.000	98.000	0.0%
Propellant Mass	**[kg]**	290.000	247.842	-14.5%
Structural Mass	**[kg]**	126.000	105.545	-16.2%
SRM Length	**[m]**	2.800	2.560	-8.6%
Fin Root Chord	**[m]**	0.59	0.56	-5.1%
Fin Tip Chord	**[m]**	0.400	0.392	-2 %

The rocket total lift-off and the propellant masses were reduced 14.5%, as well as the structural mass which was reduced 16.2%, all major improvements from the previous case study

In terms of the flight profile, a good agreement was observed regarding the previous case study and the REXUS 2 flight profiles, as portrayed in Fig. 9.

5.5. Payload and Minimum Altitude Sensitivity Analysis

In previous case studies, the optimisation capabilities of the developed framework were put to test, first for a single design variable optimisation problem, and, afterwards, for a more demanding multivariable optimisation problem with ten geometric design variables. After this initial testing, a sensitivity analysis of two significant optimisation parameters was conducted: minimum altitude and payload.

First, in order to assess the payload sensitivity,

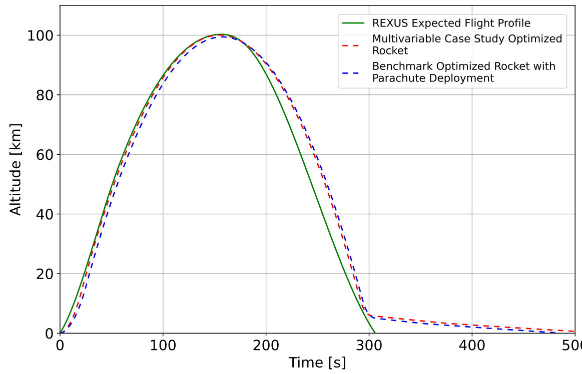

Figure 9: Flight profile comparison between the Rexus 2 and the single and multivariable optimised rockets.

the payload was set to 112.3 kg, matching with the REXUS 10 [28].

It was possible to observe a decrease of 5.7% in the rocket lift-off total mass when compared to the REXUS 2 rocket and a 10.2% increase when compared to the previous multivariable optimised rocket. Moreover, it was possible to observe a general increase in the other compared parameters which aligns well with the expected behaviour of the framework given that more payload was carried.

It then followed a minimum altitude sensitivity analysis by changing the original altitude constraint from 100 km to 82.45 km, the altitude reached by REXUS 10 [28].

In contrast to the payload sensitivity analysis, it was possible to observe a general decrease in the other compared parameters which aligns well with the expected behaviour of the framework given that a lower altitude constraint was imposed, as portrayed in Fig. 10.

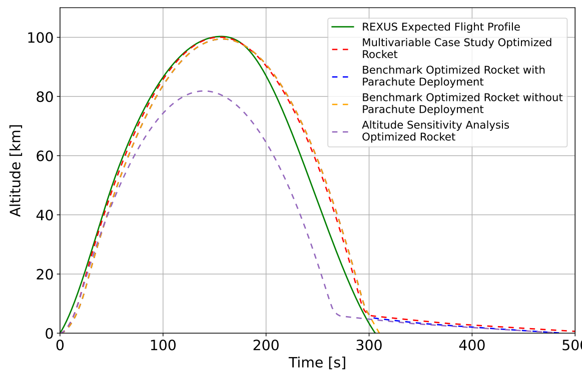

Figure 10: Comparison between the Rexus 2 expected flight profile, retrieved from [5], with the optimised rocket flight profiles of the benchmark with and without parachute deployment, multivariable case study with parachute deployment and altitude sensitivity analysis.

5.5.1 Rexus 10 Case Study

As a final test, the impact of the coupled effect of the payload and minimum altitude in the optimisation results was tested using the REXUS 10 flight mission data for comparison. The payload was set to 112.3 kg and the altitude constraint to 82.45 km.

Results showed that the optimised rocket had a total lift-off mass of 449.7 kg which is an intermediary value between the two previous sensitivity analysis obtained values. This can be explained by the combined effect of the payload and minimum altitude, the payload tends to cause a general increase in rocket sizing parameters, as opposed to, a general decrease caused by a minimum altitude reduction.

From flight data comparison between the REXUS 10 mission and the trajectory simulation a general matching behaviour was observed with a few differences: the speeds were generally higher for the optimised rocket following a simulated trajectory, the trajectory profile was too parabolic and the terminal velocity of the optimised rocket flight profile was manifestly higher than expected which might be related to a poor parachute sizing.

5.6. MDO with High Fidelity Structural Analysis

Additionally, the developed MDO framework was integrated with a high fidelity structural analysis model, developed by [29], and a new optimisation process was created in order to test with high accuracy to what extent the structural mass of the rocket could be minimised.

Using an iterative procedure, it was possible to couple the trajectory simulation with flow and FEM structural analysis ran in SOLIDWORKS, in order to minimise the structural mass of the rocket, given a fixed diameter of the body tube. From the obtained results, it was possible to observe a thickness reduction trend in the majority of the analysed components which translated in a 16.8% reduction in the structural mass. Even though, the observed results are positive, they should be perceived with caution given the fact that only four iterations were conducted due to computational time limitations, which were possibly not enough to reach a final convergence. Consequently, further testing is needed in order to corroborate this results.

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^{-5} and a step size of the finite difference method of 10^{-3}.

An initial benchmark case study was conducted to assess the accuracy of the framework with the results showing great agreement when compared to the benchmark rocket, the REXUS 2. A multivariable case study with 10 design variables then followed, further proving the efficiency and robustness of the developed framework by achieving a 14.5% total lift-off mass reduction.

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 a significant mass reduction between the REXUS 10 and the optimised rockets.

Finally, the framework was integrated in a high fidelity structural analysis, and was able to successfully achieve a 16.8% total mass reduction, although further testing needs to be conducted in other to corroborate these result as the iterative convergence process needed to be interrupted due to major time limitations.

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.

References

[1] G. Genta, “Private space exploration: A new way for starting a spacefaring society?” *Acta Astronautica*, vol. 104, no. 2, p. 480–486, nov 2014, 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, accessed on December 1st, 2023. [Online]. Available: https://www.spacefoundation.org/2023/07/25/the-space-report-2023-q2/

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

[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)*, vol. 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*, Swedish Space Corporation (SSC) & Deutsches Zentrum für Luft- und Raumfahrt (DLR).

Sandefjord, Norway: ESA Publications Division, 30 May - 2 June 2005, pp. 389 – 393.

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

[7] U. S. N. Oceanic, A. Administration, and U. S. C. on Extension to the Standard Atmosphere, *U.S. Standard Atmosphere, 1976*, ser. NOAA - SIT 76-1562. National Oceanic and Amospheric [sic] Administration, 1976.

[8] U. S. C. on Extension to the Standard Atmosphere, U. S. N. Aeronautics, and S. 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.

[9] H. Akima, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures," *Journal of the ACM*, vol. 17, no. 4, p. 589–602, Oct. 1970, DOI:10.1145/321607.321609.

[10] "Akima1DInterpolator — SciPy v1.14.1 Manual," https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.Akima1DInterpolator.html, [Accessed 05-10-2024].

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

[12] 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.

[13] 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*, vol. 4, no. 4, pp. 754–776, Nov. 1994, DOI:10.1137/0804044.

[14] 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, Jun. 2006, DOI:10.2514/6.2006-7080.

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

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

[17] M. P. Kelly, "Transcription Methods for Trajectory Optimization: a beginners tutorial," 2017, DOI:https://doi.org/10.48550/arxiv.1707.00284.

[18] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, no. 1, p. 357–373, Dec. 1992, DOI:10.1007/bf02071065.

[19] 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.

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

[21] I. M. Ross and F. Fahroo, "Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 3, p. 397–405, May 2004, DOI:10.2514/1.3426.

[22] 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*, vol. 59, no. 4, p. 1075–1104, Mar. 2019, DOI:10.1007/s00158-019-02211-z.

[23] R. Falck, J. Gray, K. Ponnapalli, and T. Wright, "Dymos: A Python package for optimal control of multidisciplinary systems," *Journal of Open Source Software*, vol. 6, no. 59, p. 2809, Mar. 2021, DOI:10.21105/joss.02809.

[24] 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, DOI:10.2514/6.2012-1673.

[25] 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*, vol. 51, no. 5, p. 1648–1663, Sep. 2014, DOI:10.2514/1.a32751.

[26] J. P. Jasa, J. T. Hwang, and J. R. R. A. Martins, "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, vol. 57, no. 4, p. 1815–1827, Feb. 2018, DOI:10.1007/s00158-018-1912-8.

[27] 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.

[28] M. S., *Rexus User Manual: Rocket Experiments for University Students*. EuroLaunch, JAN 2014, no. 7.11, Available: https://www.zarm.uni-bremen.de/fileadmin/user_upload/teaching/study_programs/rexus_bexus/RX_UserManual_v7-11_08Jan14.pdf.

[29] H. Fernandes, "Design and Analysis of the Structure of a Sounding Rocket," Master's thesis, Instituto Superior Técnico, Oct. 2024, unpublished.