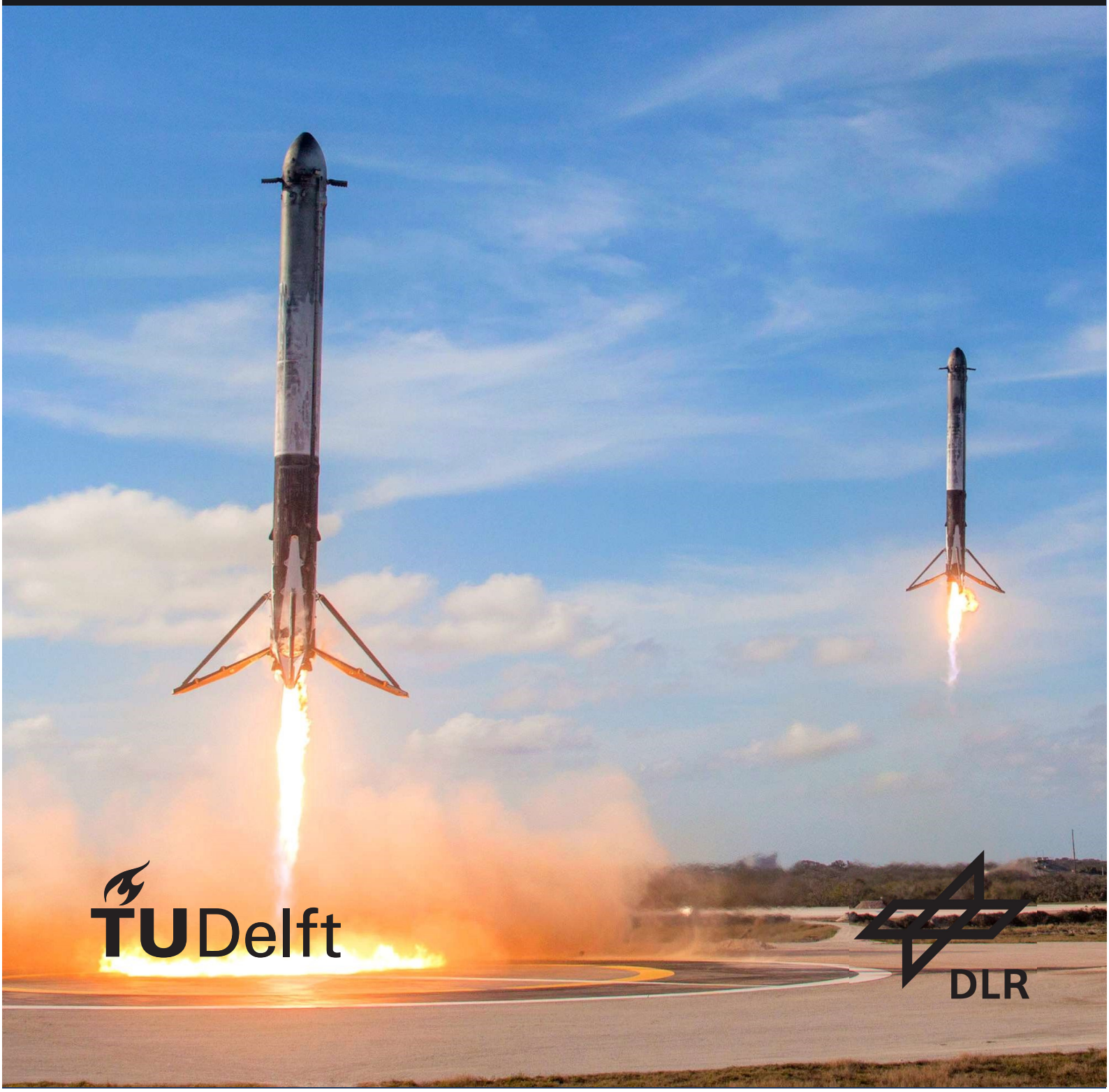
Fault-Tolerant Control Allocation Development for Clustered-Engine Reusable Launch Vehicles

Guilherme Saavedra Santos



Use of AI

According to the TU Delft OPEN Publishing Policies the use of AI tools must be disclosed.

For this thesis, an image is generated using ChatGPT (GPT-5 mini with the DALL·E image generation) with the following prompt: ”Produce a generic five engine rocket which I can use in a diagram (so profile view) with transparent background”. This image is used as the background of the diagram presented in Figure 2 of the scientific article.

Furthermore, in some sections of the thesis, ChatGPT (GPT-3.5) is used for reviewing parts of the text written by the author with prompts of the sort “Review the text in terms of grammar, clarity and style without adding any new information”. From the AI suggestions that were accepted, all were reviewed and approved by the author.

Contents

Nomenclature

List of Abbreviations

ARE	Algebraic Riccati Equation
CAP	Critical Available Power
CG	Centre of Gravity
CI	Confidence Interval
CP	Centre of Pressure
CREs	Clustered Rocket Engines
DCPs	Disciplined Convex Programs
DoF	Degrees of Freedom
ECEF	Earth-Centred Earth-Fixed
ECI	Earth-Centred Inertial
FD	Fault Detection
FDI	Fault Detection and Isolation
FI	Fault Isolation
FTC	Fault-Tolerant Control
GNC	Guidance, Navigation and Control
GP	Gimbal Point
GPO	Gauss Pseudospectral Optimisation
LOX	Liquid Oxygen
LP	Linear Programming
LQR	Linear Quadratic Regulator
LS	Least Squares
MC	Monte Carlo
MIMO	Multiple-Input, Multiple-Output
MoI	Moment of Inertia
NED	North-East-Down
NLP	Nonlinear Programming
OCP	Optimal Control Problem
ODEs	Ordinary Differential Equations
PACT	Passive-Active Control
PDG	Powered Descent Guidance
PSMs	Pseudospectral Methods
QP	Quadratic Programming
RLV	Reusable Launch Vehicle
RP	Recovery Pad
SQP	Sequential Quadratic Programming
SVD	Singular Value Decomposition
TVC	Thrust Vector Control

Subscripts or superscripts

*	Estimate
0	Initial Value
an	Analytical Method
avail	Available
B	Body Frame
cmd	Command
e	Error
f	Final Value
faulty	Faulty Case
grav	Gravity
I	Inertial Frame
i	Specific Element
jam	Jammed
max	Maximum
min	Minimum
n	Nominal Case
num	Numerical Method

opt	Optimal
opt	Propagated
p	Preferred Value
PL	Payload
prop	Propellant
ref	Reference
RP	Recovery Pad Frame
sim	Simulated

List of Symbols

α	Angle of Attack
β	TVC Angle
$\boldsymbol{\omega}$	Angular Rate Vector
Γ	Set of System's Possible Constraints
γ	Flight Path Angle
κ	Fault's Influence
$\mathcal{A}$	Aerodynamic Axial Force
$\mathcal{K}$	Feasible Control Region
$\mathcal{L}$	Control algorithms that can be implemented
$\mathcal{M}$	Aerodynamic Pitch Moment
$\mathcal{N}$	Aerodynamic Normal Force
$\mathcal{U}$	Uniform Distribution
Φ	Mayer Term
ϕ	Roll Angle
$\boldsymbol{\tau}_c$	Virtual Inputs
Ψ	Lagrange Term
ψ	Yaw Angle
ρ	Atmospheric Density
A	State Matrix
a	Acceleration Vector
B	Input Matrix
b	Bound Vector
C	Output Matrix
D	Feedthrough Matrix
d	Decision Variable Vector
E	Control Effectiveness Matrix
F	Force Vector
G	Generalised Inverse
H	Angular Momentum Vector
I	Inertia Tensor
K	Controller Gain Matrix
M	External Moment Vector
N	Constraint Matrix
P	Riccati Matrix
Q	State Weighting Matrix (LQR)
R	Control Weighting Matrix(LQR)
r	Position Vector
T	Rotation Matrix
u	Control Input Vector
v	Velocity Vector
W	Weighting Matrix
x	State Vector
y	Output Vector
Θ	Γ Corresponding Set of Parameters
θ	Pitch Angle
C	System's Constraints
c	Speed of Sound
$C_{\mathcal{A}}$	Axial Aerodynamic Force Coefficients
$C_{\mathcal{N}}$	Normal Aerodynamic Force Coefficients
C_l	Roll Aerodynamic Moment Coefficient
C_m	Pitch Aerodynamic Moment Coefficient
C_n	Yaw Aerodynamic Moment Coefficient
d	Diameter
G	Universal Gravitational Constant

g_0	Gravity Acceleration at Earth's Surface
h	Height
I	Inertia Tensor Element
I_{sp}	Specific Impulse
J	Objective/Cost
l	Control Inputs
M	Pitch Moment
m	Mass
O	System's Objectives
p	Roll Rate
Q	Dynamic Pressure
q	Quaternion/Pitch Rate/Diagonal Element of Q
r	Yaw Rate/Diagonal Element of R/Radius
S	Reference Aerodynamic Area
s	Degrees of Freedom
T	Thrust Magnitude
t	Time
u	Longitudinal Velocity/Control Input
V	Total Velocity
v_x	Lateral Velocity
v_z	Vertical Velocity
w	Normal Velocity
x	x-Axis Position
z	z-Axis Position
He	Helium
K	Kerosene

List of Figures

List of Tables

1 Introduction

In this chapter, the introduction is organised in the following way. The main topics of this research are presented (Section 1.1, Section 1.2, Section 1.3), and relations between them highlighted, leaving the more theoretical details to be explained in the literature review in Chapter 3. Furthermore, in Section 1.4, the contributions of this thesis are discussed. Subsequently, the research objective and research questions that drive the research project (Section 1.5), as well as the employed methodology (Section 1.6) and the expected results (Section 1.7), are presented. Finally, the report structure is explained in Section 1.8.

1.1. Reusable Launch Vehicles

In this section, the field of Reusable Launch Vehicles (RLVs) is described, providing a historical overview in Section 1.1.1. When building a vehicle, it is convenient to develop it so that it can be reused since this provides several advantages, namely, economic and environmental. The reuse of Launch Vehicles (LVs) allows for the more frequent use of this type of vehicle without increasing the costs as much, as well as reducing the need for new material. However, for rockets, achieving reusability is not that simple, as the intact landing of a LV presents challenges, as mentioned in Section 1.1.2. Therefore, traditionally designed rockets are meant for single use.

Recently, the paradigm of space engineering has begun to change with the development and higher investment in RLVs, as described in the following subsection.

1.1.1. Historical Overview

Thanks to [1], where the state of the art in space system reusability is described, it is possible to provide a condensed historical background of RLV systems. RLVs, as the name suggests, are designed to allow some or all components of the LV to be recovered and reused after flight. These systems began to appear in the mid-20th century, with early foundations between 1935 and 1945 through the German Silbervogel project. From there, a fast evolution of the space systems took place with the space race during the Cold War.

A shift toward reusability happened in the 1970s with the development of the Space Shuttle program, which was developed with an effort to reuse space vehicles. While the shuttle succeeded technologically, it did not achieve its economic goals, particularly the promise of significantly reducing the cost of access to low Earth orbit. As a result, confidence in reusable systems decreased, and many programs went back to focusing on single-use launch vehicles.

In recent years, an effort has been made to enhance the reusability of rockets, particularly with the transition from government-led programs to commercial initiatives. These developments often led to more cost-effective solutions, with several successful demonstrations proving that this path can be followed.

One of these private initiatives that has been investing in the development and improvement of these RLVs is SpaceX, with the objective of lowering launch costs and enabling broader access to space. SpaceX achieved a milestone by recovering the first stage of its Falcon 9 rocket [2] in 2015, followed by the first successful re-flight in 2017.

Nevertheless, there are other competitor companies that are achieving the same global goal. Blue Origin has also contributed significantly to the development of reusable systems. In 2015, the company

successfully landed its suborbital vehicle built to transport as many as six astronauts and/or a stack of payloads [3], another achievement in the advancement of RLV technology. More recently, in November 2025, Blue Origin's New Glenn rocket completed its second flight, placing NASA's ESCAPADE twin spacecraft into their planned orbit and recovering the first stage intact [4].

Europe and the rest of the world are also investing in this technology with several projects, which can be used as examples: CALLISTO [5], a cooperative effort between the French, German, and Japanese space agencies to develop a reusable-launcher technology demonstrator; and THEMIS [6], which aims to be a low-cost solution for, again, a technology demonstrator. Naturally, several other RLV projects have existed or exist nowadays that are not listed here, but the previously mentioned ones are chosen to provide a set of examples.

1.1.2. RLV Powered Descent and Landing Phase

Any mission of an LV involves several phases, each with its own performance requirements to satisfy to successfully achieve the mission goals. However, due to the lack of necessity of a soft landing for single-use launch vehicles, the descent and landing phase is basically absent. On the other hand, for RLVs, this phase of descent and landing is, of course, of great importance in order to recover the rocket in good condition.

This so-called soft landing ensures the necessary conditions for a successful recovery. It must comply with predefined tolerance values, on the maximum velocities at which the vehicle can land, on the vehicle's attitude at touchdown, and final position error to ensure the vehicle lands on the Recovery Pad (RP). As can be seen in [7], a study that analyses the landing dynamics of an RLV and derives design parameters for landing leg structures to ensure safe touch-down configurations, the final landing states influence the reach of a safe parking position of the LV.

Given the importance of this phase for RLVs, the present thesis focuses on the powered descent and landing of this type of vehicle. This includes the challenges of trajectory planning, its tracking and achieving satisfactory landing performance.

1.2. Fault Tolerant Control

In this section, Fault-Tolerant Control (FTC) is introduced, and its importance for RLVs is explained in Section 1.2.1, as well as the reasoning behind its particular importance for use in the propulsive systems in Section 1.2.2.

FTC is a study field with numerous engineering applications, as it guarantees a more reliable response for any system that has to be controlled in the case of faults. The specifics of FTC are going to be described in the literature review in Section 3.4. However, in general, after a fault in the system has been identified, FTC provides a way of mitigating the effects of the fault, for example, by reallocating the efforts to the remaining healthy actuators in order to try to achieve the set of objectives the best way possible.

1.2.1. FTC for RLVs

As in the development of any vehicle, it is important that it is reliable and can complete its mission or provide the best response possible, even in the case of the occurrence of a fault. This fact is even more crucial in the case of a rocket, since the non-completion of the mission can result in a crash, and so if unmanned, the loss of the investment made, or even worse, if with a crew, the potential loss of human lives.

The use of FTC is supposed to reduce the risk of mission failure. Even in cases where the primary mission objectives cannot be fully achieved, FTC can still increase the probability of safely recovering the vehicle. Moreover, in contrast to single-use LVs, where the components are new, RLVs are supposed to perform several flights, causing the materials to undergo wear. Even with constant maintenance, the fault probability increases. Nevertheless, the system should be prepared for that.

Practically, FTC can be categorised into active and passive [8]. In the passive case, the control law remains unchanged when a fault occurs, but the fault-tolerant behaviour is guaranteed to a certain extent by design. For example, a system can be considered fault-tolerant if the rocket contains hardware redundancy [9]. In the active case, the control law is adapted in accordance with the identified fault in real time, which can lead to modifications in the control algorithm as adjustments to tuning parameters, or even a change

in the control strategy. Furthermore, the control law may involve control allocation, which can also be reconfigured, as explained in more detail in Section 1.3.

1.2.2. FTC Use Motivation for Propulsion Systems of LVs

This thesis focuses on faults related to the propulsion system. Therefore, in this subsection, it is shown why these faults are worth studying, based on rocket launches in recent years.

By analysing failure statistics, propulsion systems have historically been the most subject to failure in space launch vehicles. This fact can be justified by examining data from two nations: the United States and Russia. Out of 47 documented launch failures, between 2000 and 2022, involving U.S. and Russian launch vehicles, 72.3% were attributed to propulsion subsystem malfunctions. Looking at each country individually, of the 18 U.S. launch failures, 14 were caused by propulsion issues, which is approximately 78% of the total. Similarly, Russia experienced 19 propulsion-related failures out of 29 launches, which represents about 66% [10].

A separate study analysing 57 launch failures worldwide between 2006 and 2021 found that 54% of them were related to problems in the propulsion system [11]. Although this global percentage is lower than that observed in the U.S. and Russia, it still highlights the point that the propulsion system is consistently a mission-critical subsystem in launch vehicle design.

1.3. Control Allocation

The central topic of this thesis, which is introduced in this section, is control allocation. Therefore, the concept of control allocation and how it relates to LVs is presented in Section 1.3.1, and to FTC, in Section 1.3.2.

Control allocation strategies are used so that, instead of the controller directly commanding each of the actuators of a system, signals acting as a virtual control command are computed. These signals, which usually represent the cumulative external forces and external torques in the case of a vehicle, are then applied via the available actuators. This allows not only the achievement of the intended response from the system but also the distribution of the effort among the actuators in predefined ways [12].

1.3.1. Control Allocation for LVs

LVs are vehicles that depend on their control system, and that usually have a large number of actuators, from several Thrust Vector Control (TVC) actuators used to control each of the engines, to reaction control system thrusters and aerodynamic control surfaces. This high number of actuators justifies the use of a control allocation algorithm, which receives a request from the controller, such as a set of forces and moments the group of actuators should apply to the vehicle, and consequently distributes it across the available actuators. This also allows the system to replace multiple controls with a reduced number of virtual controls [13] still being able to influence the usage level of the actuators, which may offer advantages in terms of managing actuator degradation, preventing operation near saturation limits, and enabling reconfiguration, as demonstrated in the next subsection.

1.3.2. Control Allocation in FTC

In the case of a fault, the nominal control allocation would behave as if the rocket were in a healthy condition, resulting in allocating control efforts to damaged actuators, which would degrade the response of the system.

As mentioned in the previous subsection, control allocation allows for reconfiguration, meaning that the control allocation algorithm can be changed throughout the mission. This becomes useful for achieving FTC allocation. For example, if a fault is identified, this reconfiguration can be used to transform the allocation into a fault-aware algorithm, better distributing the control efforts to the healthy actuators [14].

1.4. Research Gaps and Project Scope

In Section 1.2, the concept of FTC is introduced. FTC has become a well-established field, with a solid theoretical background. A book that gives a theoretical basis for this topic is [8], which provides an introduction to fault diagnosis and fault-tolerant control. It explains how faults in sensors, actuators, or other components can be detected, identified, and compensated for to ensure reliable system operation,

showing that a theoretical basis exists for these types of systems and that the fault-tolerance problem can be approached by different methods, but usually requiring parallel systems capable of diagnosing the faults.

FTC is applied in several contexts, aerospace systems being one of them. In the book [15], the authors examine the design, analysis, and implementation of Fault Detection, Identification, and Recovery technologies for aircraft and spacecraft, presenting examples of real-world applications of these types of systems. Several examples of FTC applied to aerospace can be found in the literature, from which one including an LV is chosen to serve as an example: in [16] an active FTC approach for LV attitude control is developed, which integrates Kalman filter–based fault diagnosis and control reconfiguration to accurately detect and isolate sensor and actuator faults, reconstruct faulty signals, and maintain attitude tracking even with system faults.

More specifically, LVs equipped with clustered engines and the analysis of potential engine faults are detailed in the study from N. Paulino et al. [17]. In that work, the authors developed guidance and control reconfiguration strategies that enable an LV with a clustered propulsion system to maintain stability and performance in the presence of propulsion faults. This paper focuses on the ascent phase of the mission. However, it is not possible to understand whether the vehicle can still descend and land safely under the same fault conditions. Additionally, not all fault cases are discussed in detail: while various faults are considered, their specific effects are not examined. This article is part of the results obtained by the ESA-financed project “Fault-Tolerant Control of Clusters of Rocket Engines”. Also part of this project, another paper exists [18] which is a summary of all the results obtained in the project. In this summary, the descent is covered but not in a detailed manner, which makes it difficult to isolate the contribution of each FTC strategy and to evaluate the impact of different fault types and fault characteristics on landing success. Furthermore, although this project makes use of allocation it does not go into detail on how it is applied, only stating that the allocation uses a high-efficiency quasi-linear algorithm based on [19]. On the other hand, control allocation techniques comparison for LVs has been covered in the study from Diego Navarro-Tapia, Pedro Simplício and and Andrés Marcos [20] for an ascent flight with fault occurrences, but only applied to a specific instant of the ascent. This reveals a gap in the existing research, which the present thesis aims to address. Therefore, one of the goals of this thesis is to deeply analyse how a fault-aware reconfiguration algorithm can compensate or mitigate the faults affecting the propulsion system in a powered descent and landing scenario.

Ultimately, the scope of this project is to develop a fault-aware control allocation for RLVs applied to a cluster of liquid rocket engines during the powered descent and landing phase. The work includes testing under multiple fault conditions to evaluate their effects on vehicle performance and recovery capability, as better explained in Section 1.6. Specifically, the study aims to quantify how different fault types, magnitudes, and time occurrences influence the vehicle throughout descent and landing, and to determine the extent to which fault-awareness in the allocation algorithm improves landing success in each of these faulty scenarios. In doing so, this thesis advances the understanding of fault effects in RLV operations.

1.5. Research Questions

Considering the research gaps and project scope presented in Section 1.4, it is necessary to translate this information into a clear research objective which defines the overall aims of the study and the corresponding research questions that specify the particular issues the study seeks to investigate in order to achieve the objective.

Research Objective

Develop and evaluate a fault-tolerant control allocation strategy for reusable launch vehicles with a cluster of throttleable rocket engines, and analyse the effects of the control technique and fault conditions on vehicle performance and recovery capability, including quantifying the impact and severity of faults across various flight instants of the descent to enhance system resilience.

Research Question 1

How do different fault conditions and the application of a fault-aware control allocation strategy influence the recovery performance, behaviour, and overall mission success of an RLV?

Subquestions:

1. How do different types of faults affect the vehicle's dynamic behaviour and its potential for successful post-fault recovery?
2. How do the magnitude, timing, and affected engine of a fault influence performance degradation and recoverability?
3. To what extent does a fault-aware control allocation technique improve recovery performance under various fault scenarios?

Research Question 2

To what extent can a gain-scheduled Linear Quadratic Regulator synthesised considering a linearised rocket dynamics across the descent trajectory, maintain stability and acceptable performance under various fault scenarios during a descent and landing mission?

1.6. Methodology

Taking into account the research objectives and questions presented in Section 1.5, the following methodology is proposed to obtain the desired outcomes:

1. Vehicle Model

   Since this research is not based on an existing or in-development launch vehicle, the first step involves defining a rocket to work with. This vehicle is designed based on reasonable engineering assumptions to ensure that its characteristics are representative of a real-world system, by using a benchmark that was developed to test a guidance algorithm named DESCENDO, based on convex optimisation [21]. For this thesis, the single-engine configuration of the benchmark is replaced with a cluster of liquid rocket engines.

2. Rocket Dynamics Modelling

   Once the vehicle parameters are established, the next phase consists of developing a mathematical model of the rocket. This includes formulating the equations of motion and implementing a nonlinear simulation environment to accurately represent the system's dynamics.

3. Optimised Descent Trajectory Planning

   Subsequently, the mission profile is defined. Given that the focus of this study is on the powered descent and landing phase, an optimised descent trajectory is designed to achieve the desired performance and landing conditions.

4. Linearisation

   After the nonlinear model and trajectory are established, the system is linearised around several operating points along the descent trajectory. These linearised models serve as a basis for controller design and analysis.

5. Controller

   With the base framework in place, the control system can be developed. An optimal controller, specifically a Linear Quadratic Regulator (LQR) implemented with gain scheduling, is designed, allowing the rocket to follow the defined trajectory and so addressing Research Question 2. This design choice relies on the fact that this control technique is well established, having been used in projects in different fields. For launch vehicles in particular, it has also been used (as is the case of [22]), but it is not a very commonly used method, it is more commonly used as a benchmark, as in [23]. Additionally, in this problem, some physical effects such as wind gusts, sensor noise, actuator disturbances, propellant slosh, and structural flexibility are not included. This justifies the use of an

LQR controller, as most of the system dynamics are modelled, in contrast to some of the problems explored in the state of the art (for example, [17]), which use robust control approaches. Furthermore, the properties and gains of an LQR controller are relatively easy to interpret, and consequently, its tuning is facilitated, which contributes to a simple but effective control solution which allows for the primary focus of the thesis on the control allocation and fault influence.

6. Control Allocation

   Once the controller design is completed, a control allocation algorithm is implemented. Initially, an allocation algorithm is developed to distribute control efforts among all actuators in a way that best satisfies the controller's commands. Subsequently, the algorithm is augmented considering the fault information to ensure that, in the presence of actuator faults, the control effort is effectively redistributed among the remaining healthy actuators. The implementation of this algorithm enables the investigation of Research Question 1.

7. Monte Carlo Campaign

   Finally, a Monte Carlo (MC) simulation framework is developed to perform extensive testing. This framework enables systematic variation of several parameters, including initial conditions, aerodynamic coefficients, vehicle characteristics, and fault scenarios. The fault cases under investigation are limited to propulsion system faults and are classified into three categories:

   - Partial engine thrust loss – the affected engine reduces its overall thrust capability.
   - Total engine thrust loss – the affected engine produces zero thrust.
   - Thrust vector control (TVC) jamming – the affected thruster engine becomes locked at a fixed gimbal angle.

   The resulting simulation campaign generates the data required to evaluate system performance and address the research questions.

1.7. Expected Results

In this section, the results that are expected to be obtained from the previously described methodology are presented. In general, as already mentioned, the main expectation of this thesis is to demonstrate that a fault-aware allocation algorithm improves landing performance in faulty scenarios.

Regarding fault severity, it is anticipated that a complete loss of thrust in one engine often exceeds the vehicle's remaining control authority, resulting in an unrecoverable situation. For a partial thrust fault, a threshold level of residual thrust is expected to exist, below which recovery becomes impossible. This research is meant to identify that threshold by evaluating partial-loss scenarios and quantifying the recovery success rate. For TVC jamming cases, where the affected engine continues to produce thrust at a fixed gimbal angle, the FTC algorithm is expected to compensate more effectively than in total engine loss scenarios. This expectation derives from the jammed engine's full contribution to propulsive force, while the remaining engines redistribute moment generation. However, a critical jamming angle may exist beyond which compensation is no longer feasible, and this limit is investigated. The precise boundaries of compensation for TVC jamming are established through simulation.

These expectations can be organised into a set of hypotheses:

1. **Hypothesis H1** - Fault-aware allocation improves landing success
   - *Prediction*: Fault-aware allocation results in a higher landing success probability than a nominal controller under actuator fault scenarios.
   - *Metrics*: landing success rate comparison between nominal allocation and fault-aware allocation.
   - *Evaluation*: Monte-Carlo trials across a range of initial conditions and faults.
2. **Hypothesis H2** - Total engine loss is often unrecoverable
   - *Prediction*: Complete loss of one engine frequently leads to mission failure. This happens if the remaining thrust cannot counter gravity and maintain attitude control.
   - *Metrics*: landing success rate of total engine loss faults.
   - *Evaluation*: Monte-Carlo trials with total engine loss across a range of initial conditions.

3. **Hypothesis H3** - Partial thrust loss has a recoverable threshold
    - *Prediction*: There probably exists a value of remaining thrust for a partially failing engine below which recovery is not possible.
    - *Metrics*: Correlation between success rate and fraction of remaining thrust.
    - *Evaluation*: Monte-Carlo trials including partial thrust loss across a range of initial conditions.
4. **Hypothesis H4** - TVC jamming is more manageable than thrust loss
    - *Prediction*: TVC jamming can be more readily compensated by the allocation algorithm than a total thrust loss because the jammed engine still contributes with axial force.
    - *Metrics*: landing success rate under various gimbal jammings.
    - *Evaluation*: Monte-Carlo trials including TVC jamming across a range of initial conditions and comparison with the results obtained for thrust loss.

1.8. Report Structure

This thesis report is structured to provide a presentation of the research conducted, organised into distinct parts that build upon one another.

Firstly, Part I presents a scientific article that provides the main procedures and results of this research. This is supposed to exist as a standalone document, and so, the article necessarily includes a small amount of content that might appear elsewhere in the thesis, though it displays the core of the research. Additionally, the literature review is presented in Chapter 3, establishing the theoretical foundation for the topics addressed in this study.

Secondly, Part II provides supplementary material through two supporting chapters. Additional analysis and results are presented in Chapter 4, while important for supporting the overall work, these findings are supplementary to the core results presented in the scientific article. Moreover, Validation and Verification of the methodologies employed throughout the thesis are performed in Chapter 5, ensuring the reliability and trustworthiness of the results obtained.

Finally, Part III concludes the thesis. The project conclusions are presented in Chapter 6, addressing each research question based on the results obtained. Furthermore, an overview of the recommendations for future continuation of this research is provided in Chapter 7.

Part I

Scientific Article

Fault-Tolerant Control Allocation Development for Clustered-Engine Reusable Launch Vehicles

G. Saavedra Santos*
Delft University of Technology, Faculty of Aerospace Engineering, 2629 HS Delft, The Netherlands
German Aerospace Center, Institute of Space Systems, 28359 Bremen, Germany

This paper presents the development and testing of a Fault-Tolerant Control (FTC) allocation algorithm aimed at decreasing the negative influence of engine faults during the powered descent and landing phase of a reusable rocket. The faults considered include total or partial loss of engine thrust and jamming of the Thrust Vector Control (TVC) angle. The objective is to assess the feasibility and operational limits of such strategies during this flight phase and to evaluate how different fault scenarios influence landing performance. To this end, the rocket control is designed using an LQR controller integrated with a control allocation algorithm that can be reconfigured in case of a fault occurrence. The results indicate that engine power-loss faults have the greatest impact on landing success in comparison with jamming faults, and that the fault-aware allocation approach significantly improves system probability of mission success, while still exhibiting a limit to the severity of faults it can compensate for.

I. Introduction

Rocket reusability has been regaining a lot of interest in the past years. Nowadays, there still exist a lot of Launch Vehicles (LV) that are expendable, with only some reusable systems being operational. Nevertheless, demonstration programs and development initiatives exist, confirming the renewed interest in reusability that is largely fueled by SpaceX's and Blue Origin's achievements of successful landings. Namely, some successeful cases can be given as examples, SpaceX recovered the first stage of its Falcon 9 rocket [1] in 2015, followed by the first successful re-flight in 2017, while Blue Origin successfully landed its suborbital vehicle in 2015 [2], and in 2025 New Glenn rocket completed its second flight, recovering the first stage intact [3]. Furthermore, the rest of the world is also developing projects of this sort, as is the case of a joint effort between the French, German and Japanese space agencies with CALLISTO [4], a reusable-launcher technology demonstrator. Other examples exist, such as another technology demonstrator named THEMIS [5], but these are chosen to illustrate the effort being made towards the reusability of rockets.

Reusability offers several advantages, including economic and environmental benefits, as well as a reduced risk of debris falling at sea or on land [6]. For the reuse of rockets, the landing system should guarantee a soft, controlled touchdown. Moreover, these rockets being as fault-tolerant as possible is an advantage, having the best chance of landing even in faulty cases.

A variety of faults can occur in a rocket, but an examination of the statistics shows a clear trend: of 47 documented launch failures involving U.S. and Russian launch vehicles from 2000 to 2022, 72.3% were caused by propulsion subsystem malfunctions. The country-level breakdown is similar: of 18 U.S. failures, 14 (about 78%) were caused by propulsion issues, while Russia experienced 19 propulsion-related failures out of 29 launches (about 66%) [7]. Therefore, the present study focuses on propulsion faults, the faults that have a greater influence on the failures of rocket missions.

Due to this set of faults that can occur, it is helpful for the LVs to exhibit a certain level of fault-tolerance in order to maximise the probability of success of the mission even in the case of some malfunctioning.

*MSc. Student, Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology

Focusing on the state of the art about FTC, it is a well-studied domain where the book [8] can serve as an example by providing a solid theoretical base not only for this topic but also for other systems that work side by side with FTC, as is the case with Fault Detection and Isolation (FDI) systems. A classic framework can be viewed as consisting of two stages, the FDI and the FTC: fault detection indicates that the problem at hand has shifted from the original formulation, and fault isolation specifies the subset of constraints that remain valid, corresponding to the components that continue to function correctly. With this information, the FTC can be applied. The FTC can be passive, where control uses alternatives to the faulty components indicated by the FDI to tolerate faults without changing the control law or, on the other hand, active, where the FTC uses FDI information to know specifics about a fault and then the Guidance, Navigation and Control (GNC) solution is reconfigured, switching controllers, trajectory or allocation strategy.

The use of FTC in aerospace vehicles has also been studied, as demonstrated by the book [9], where FDI and FTC application for aircraft and spacecraft, with a focus on real-world implementation, is presented. There are several examples of FTC application in the aerospace context covering different types of FTC startegies. Nowadays, the most common fault-tolerant applications for launch vehicles are on hardware redundancy [10]. An example is the Ariane 5 launch vehicle, which illustrates a passive FTC approach designed to tolerate expected failures through pre-planned redundancies and fixed recovery mechanisms [11]. However, FTC can also be applied in an active way for LVs, meaning the control law is adapted in case of a fault [8]. An example is given with [12], where an active FTC approach for launch vehicle attitude control that combines Kalman filter–based fault diagnosis with control reconfiguration to maintain accurate attitude tracking under sensor and actuator faults is used.

Focusing on the application of FTC solutions to propulsion faults in LVs, particularly in over-actuated configurations such as LVs with clustered engines, the use of active FTC strategies can help attenuate the effects of these types of faults, as discussed in the paper [10]. In this paper, guidance and control reconfiguration strategies in case of propulsion faults are presented for an LV with clustered engines, and it is part of the results obtained in an ESA-financed project “Fault-Tolerant Control of Clusters of Rocket Engines”. This paper does not go into detail about certain topics, such as the influence of different types of faults and their characteristics and how control allocation is applied and its direct influence on the success of the mission. Furthermore, this paper analyses the results in an ascent scenario, as is the case with the great majority of studies about LVs, since a big part of the rockets is not reusable, and so the descent phase is neglected. Another paper included in the same project [13] presents a summary of all the results obtained in the project. In this paper, the descent is analysed, but not in detail, not allowing for understanding how each FTC strategy influences the landing success or even how different faults and characteristics of each fault influence the success of the landing.

Both the works [10] and [13] mention that an allocation algorithm is used and can be applied to redistribute thrust levels and optimise deflections within the cluster, but neither goes into detail on how it is done, only stating that the allocation uses a high-efficiency quasi-linear algorithm based on [14]. Another paper [15] compares different allocation techniques applied in an ascent scenario of an LV with fault occurrences, but only applied to a specific instant of the ascent.

These last 2 paragraphs reveal a gap in the current state of the art, where there exists no study that goes into detail about how an FTC allocation algorithm in the descent and landing of a Reusable Launch Vehicle (RLV) can improve the landing success, and also, there is no study revealing how different faults and the related fault characteristics influence the RLV performance in this descent phase of the trajectory.

The current study aims to develop a fault-aware allocation strategy to distribute thrust levels and TVC angles for Clustered Rocket Engines (CREs), as well as the RLV descent mission and result analysis. For this fault-aware allocation algorithm to work, it needs to do so based on information provided by an FDI. The FDI system itself is out of the scope of this study, and so it is not developed within this work. It is assumed to provide the control system with real-time information regarding fault type, magnitude, and location.

In general, with this paper, it is possible to understand the impact of each of the different propulsion faults,

as jamming and engine thrust-loss, as well as the influence of the fault characteristics: magnitude, timing and affected engine. Furthermore, the objective is also to provide results that provide understanding in an isolated way about the functioning of a fault-aware allocation algorithm and how it affects the behaviour of an RLV in a powered descent and landing scenario, also comparing it to the use of a non-fault-aware allocation algorithm in the same situation, in order to have an idea about the advantages of using FTC solutions.

The paper is organised as follows. In section II, the launch vehicle under study, its mission profile, the equations of motion, their linearisation, and the considered fault scenarios are described. section III and section IV present the development of the optimal controller and the control allocation strategy, respectively. Finally, section V explains the simulator setup and the structure of the Monte Carlo (MC) campaign, and presents the resulting data together with the corresponding analysis.

II. Problem Statement

A. Vehicle

This paper focuses on a Reusable Launch Vehicle (RLV) based on a benchmark that was developed to test a guidance algorithm named DESCENDO, based on convex optimisation [16]. For this work, the single-engine configuration of the benchmark is replaced with a cluster of liquid rocket engines. The cluster design allows for FTC, due to over-actuation: if one engine fails, the remaining engines can compensate and try to maintain performance. The configuration has five engines arranged in a cross pattern. Moreover, the central engine is meant not to be deflected, producing only a longitudinal force on the vehicle, while the other four engines can normally operate the TVC, as illustrated in Figure 1. In terms of thrust, the rocket has a throttle range that goes from 30% to 110% of its nominal maximum thrust and operates with a combination of Kerosene and Liquid Oxygen (LOX) propellant. A summary of the vehicle's principal characteristics is provided in Table 1.

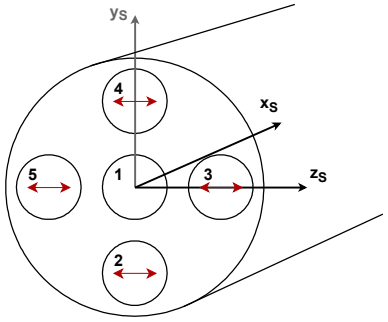


Fig. 1 Thrusters configuration and TVC degrees of freedom (red)

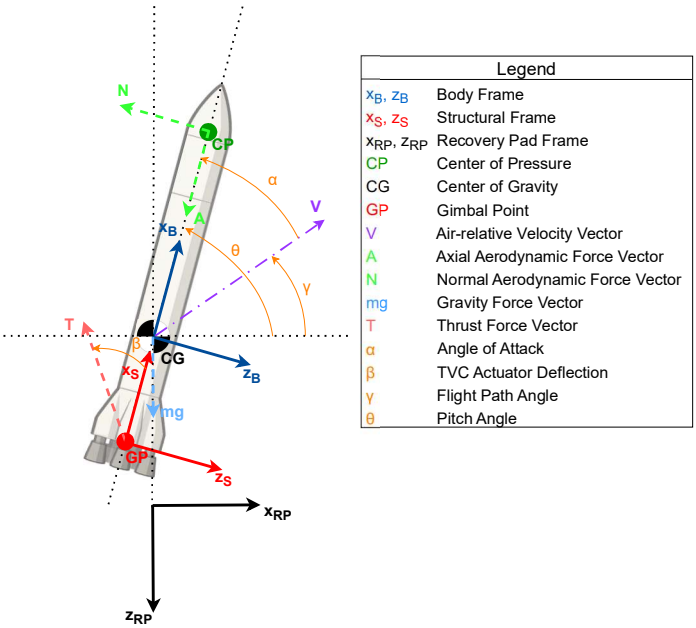


Fig. 2 Planar flight dynamics

Table 1 Rocket parameters

Name	Symbol	Value	Unit
Dry mass	m_{dry}	2750	kg
Initial wet mass	m_{wet}	4300	kg
Height	h	11.70	m
Diameter	d	3	m
Lateral engine deviation	$\lvert z_{\text{lat engine},S} \rvert$	0.45	m
Base LOX tank position	$x_{\text{LOX},S}$	2.90	m
Base He tank position	$x_{\text{He},S}$	0	m
Base K tank position	$x_{\text{K},S}$	5.80	m
Dry rocket CG position	$x_{\text{CG},S}$	4.60	m
Dry rocket lateral moment of Inertia	$I_{yy,\text{dry}}$	40267	kg m^2
Maximum thrust (individual thruster at 100%)	T_{max}	11.61	kN
Thrust rate limit	$\dot{T}_{\text{max}}$	23.22	kN/s
Maximum TVC deflection angle	β_{max}	10	deg
Maximum TVC deflection rate	$\dot{\beta}_{\text{max}}$	10	deg/s
Specific impulse	I_{sp}	282	s

B. Three Degrees of Freedom (DoF) Equations of Motion

The rocket is modelled as a variable-mass rigid body operating in a planar configuration with three DoF: two translational coordinates (horizontal position x, vertical position z) and one rotational coordinate (pitch angle, θ). Structural flexibility and propellant sloshing effects are neglected. A flat, non-rotating Earth reference frame is used for this analysis. Therefore, due to the short trajectory distances characteristic of powered descent and landing, the effects of planetary curvature and rotation are neglected [17]. The dynamics are formulated in an Earth-Fixed North–East–Down (EFNED) coordinate system, fixed at the Recovery Pad (RP). By assuming a non-rotating Earth, this frame is treated as the inertial reference frame. Furthermore, a body-fixed reference frame is also used as shown in Figure 2, utilising the rotation matrix presented in Equation 1, and applying it to the inertial frame, this body frame can be obtained. The figure also includes the resultant thrust vector of the propulsion system, along with the aerodynamic forces, which are expressed in the body frame. This thrust vector represents the combined force produced by the set of engines, accounting for both their individual thrust magnitude and gimbal angles. The aerodynamic forces are divided into axial and normal forces as represented in Equation 2 and Equation 4, respectively. Furthermore, the aerodynamic moment is also represented in Equation 3. The powered 3-DoF motion of a reusable rocket is represented by the following set of nonlinear differential equations in Equation 5.

$$\mathbf{T}_{RP\to B} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{1}$$

$$\mathcal{A} = QSC_{\mathcal{A}} \tag{2}$$

$$\mathcal{M} = QSdC_m \tag{3}$$

$$\mathcal{N} = QSC_{\mathcal{N}} \tag{4}$$

$$\begin{aligned} \dot{\mathbf{r}}_{RP} &= \mathbf{v}_{RP} \\ \dot{\mathbf{v}}_{RP} &= \mathbf{a}_{RP}^{\text{grav}} + \mathbf{a}_{RP}^{\text{thrust}} + \mathbf{a}_{RP}^{\text{aero}} \\ \dot{\theta} &= q \\ \dot{q} &= I_{yy}^{-1}[M_B^{\text{thrust}} + M_B^{\text{aero}}] \end{aligned} \tag{5}$$

In Equation 5, $\mathbf{r}_{RP}$ and $\mathbf{v}_{RP}$ represent the Centre of Gravity (CG) position and velocity in the RP frame, respectively. The vehicle attitude is described by the pitch angle θ, with $\theta = 90°$ being the upright orientation of the rocket. The translational dynamics are governed by $\dot{\mathbf{v}}_{RP}$, which is the sum of the gravitational, propulsive (thrust) and aerodynamic acceleration contributions (see Equation 6). The rotational dynamics are given by $\dot{q}$, which equals the net pitching moment with contributions from the thrust and the aerodynamics (see Equation 7) multiplied by the inverse of the inertia matrix. In this case, the only relevant element of the inertia matrix is the I_{yy} due to the 2D construct of the problem. Finally, the derivatives of the position and pitch are equal to the velocity and the pitch rate, respectively.

$$\begin{aligned} \mathbf{a}_{RP}^{\text{grav}} &= \frac{1}{m}\begin{bmatrix} 0 & mg \end{bmatrix}^\top \\ \mathbf{a}_{RP}^{\text{thrust}} &= \mathbf{T}_{RP\to B}^\top \begin{bmatrix} T\cos\beta & -T\sin\beta \end{bmatrix}^\top \\ \mathbf{a}_{RP}^{\text{aero}} &= \mathbf{T}_{RP\to B}^\top \begin{bmatrix} -\mathcal{A} & -\mathcal{N} \end{bmatrix}^\top \end{aligned} \tag{6}$$

$$\begin{aligned} M_B^{\text{thrust}} &= T x_B^{TVC} \sin\beta \\ M_B^{\text{aero}} &= -\mathcal{M} \end{aligned} \tag{7}$$

The final state-space system can be defined with a state vector containing horizontal position and velocity, vertical position and velocity, pitch and pitch rate, and the inputs being thrust and the TVC angle, as represented in Equation 8. The condensed expression of the nonlinear state-space system is presented in Equation 9.

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_{RP} & \dot{x}_{RP} & z_{RP} & \dot{z}_{RP} & \theta & q \end{bmatrix}^\top \\ \mathbf{u} &= \begin{bmatrix} T & \beta \end{bmatrix}^\top \end{aligned} \tag{8}$$

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{9}$$

C. Mission Description

To generate the mission's powered descent and landing trajectory, a general-purpose optimal control software [18] is used to solve the Nonlinear Programming (NLP) problem that represents this descent.

In short, the mission goal is to model a 3-DoF descent, accounting for vertical and lateral position and vehicle pitch. The vehicle begins at an altitude of 5000 m with an initial lateral offset from the RP, and it ends at touchdown on the recovery pad. For this study to establish the NLP problem, the system states, initial conditions, constraints, and cost function to obtain an optimal powered descent trajectory are defined.

The cost function used is shown in Equation 11 with the respective weights presented in Table 2. The decisions behind this cost function are taken following the work of [19]. Minimising fuel consumption is defined as the primary objective of the function, which translates to minimising the integral of the thrust magnitude along the trajectory. Moreover, to avoid any sudden excitations in the TVC angle and the thrust, the corresponding rates are also penalised. Therefore, in order to be able to define a cost function which allowed controlling the smoothness of the trajectory and the rocket mass evolution, an augmentation of the states is performed in relation to the ones presented in Equation 8, as observable in Equation 10, 3 states are added: mass, thrust and TVC angle. Consequently, 2 new inputs are used: thrust rate and TVC angle rate.

The problem definition data in Table 3 presents the general trajectory bounds, and in Table 4, the imposed initial and final conditions are also presented. In some cases, these conditions are given as intervals, which

$$\mathbf{x}_{\text{aug}} = \begin{bmatrix} x_{RP} & \dot{x}_{RP} & z_{RP} & \dot{z}_{RP} & \theta & \dot{\theta} & T & \beta & m \end{bmatrix}^\top$$
$$\mathbf{u}_{\text{aug}} = \begin{bmatrix} \dot{T} & \dot{\beta} \end{bmatrix}^\top \tag{10}$$

$$\text{minimise } J = \int_{t_0}^{t_f} T + \omega_{\dot{T}} \dot{T} + \omega_{\dot{\beta}} \dot{\beta} \tag{11}$$

Table 2 Cost function weights

Weight	Value
$\omega_{\dot{T}}$	10^{-3}
$\omega_{\dot{\beta}}$	10^{-4}

allows the solver to select values within the predefined bounds.

Table 3 NLP problem bounds

Name	Symbol	Bounds	Unit
Time	t	[0, 100]	s
Lateral position	x_{RP}	[-500, 500]	m
Vertical position	z_{RP}	[-5000, 0]	m
Lateral velocity	$v_{x,RP}$	[-500, 500]	m/s
Vertical velocity	$v_{z,RP}$	[-500, 500]	m/s
Pitch	θ	[83, 97]	deg
Pitch rate	q	[-7, 7]	deg/s
Mass	m	[2750, 4300]	kg
Thrust	T	[23.22, 58.05]	kN
TVC angle	β	[-5, 5]	deg
Thrust rate	$\dot{T}$	[-23.22,23.22]	kN/s
TVC angle rate	$\dot{\beta}$	[-10, 10]	deg/s
Angle of attack	α	[170, 190]	deg

Comparing the optimiser bounds with the rocket's physical limits shows that some actuation parameters, such as thrust and TVC angle, are constrained more tightly in the optimisation problem than in the actual hardware. For instance, although the engine can operate between 30% and 110% of its nominal maximum thrust, the optimiser is restricted to a narrower range of 40% to 100%. A similar approach is taken for the thrust vector control angle: while the physical system allows up to $\pm$ 10 deg, the optimisation problem limited it to $\pm$5 deg. This conservative choice moves the nominal trajectory away from constraint limits, providing the controller with additional margin to correct errors that occur when the nonlinear system is tested under varying initial conditions and uncertain aerodynamic and vehicle parameters. Since the study's primary objective is to evaluate responses to engine faults, which inevitably produce deviations from the nominal trajectory, the bounds used create a safety interval that facilitates fault recovery by the controller and does not allow the actuators to reach near saturation cases in a nominal scenario.

The NLP solution, shown in Figure 3 and Figure 4, satisfies all imposed constraints and the previously mentioned initial and final conditions with all the state and control variables remaining within their allowable limits throughout the manoeuvre. A global inspection of the solution allows the trajectory to be divided into two distinct phases. In the first phase, the thrust magnitude is held at its minimum and in the final phase, the thrust is driven to its maximum, occurring roughly during the last 10 seconds of the trajectory. This behaviour is consistent with the known fuel-optimal, bang–bang thrust profile, in which the thrust magnitude switches instantaneously between its bounds [20]. Only one transition from minimum to maximum thrust is observed here, which is in accordance with the demonstrated theoretical limit of at most two boundary transitions [17].

Table 4 NLP problem initial and final values/bounds

Name	Symbol	Initial Value/Bounds	Symbol	Final Values/Bounds	Unit
Time	t_0	0	t_f	[10, 100]	s
Lateral position	$x_{RP,0}$	300	$x_{RP,f}$	0	m
Vertical position	$z_{RP,0}$	-5000	$z_{RP,f}$	0	m
Lateral velocity	$v_{x,RP,0}$	[-50, 50]	$v_{x,RP,f}$	0	m/s
Vertical velocity	$v_{z,RP,0}$	150	$v_{z,RP,f}$	0.5	m/s
Pitch	θ_0	[87, 93]	θ_f	90	deg
Pitch rate	q_0	[-2, 2]	q_f	0	deg/s
Mass	m_0	4300	m_f	[2750, 4300]	kg
Thrust	T_0	[23.22,58.05]	T_f	[23.22,58.05]	kN
Thrust rate	$\dot{T}_0$	[-23.22,23.22]	$\dot{T}_f$	[-23.22,23.22]	kN/s
TVC angle	β_0	[-5, 5]	β_f	[-5, 5]	deg
TVC angle rate	$\dot{\beta}_0$	[-10, 10]	$\dot{\beta}_f$	[-10, 10]	deg/s

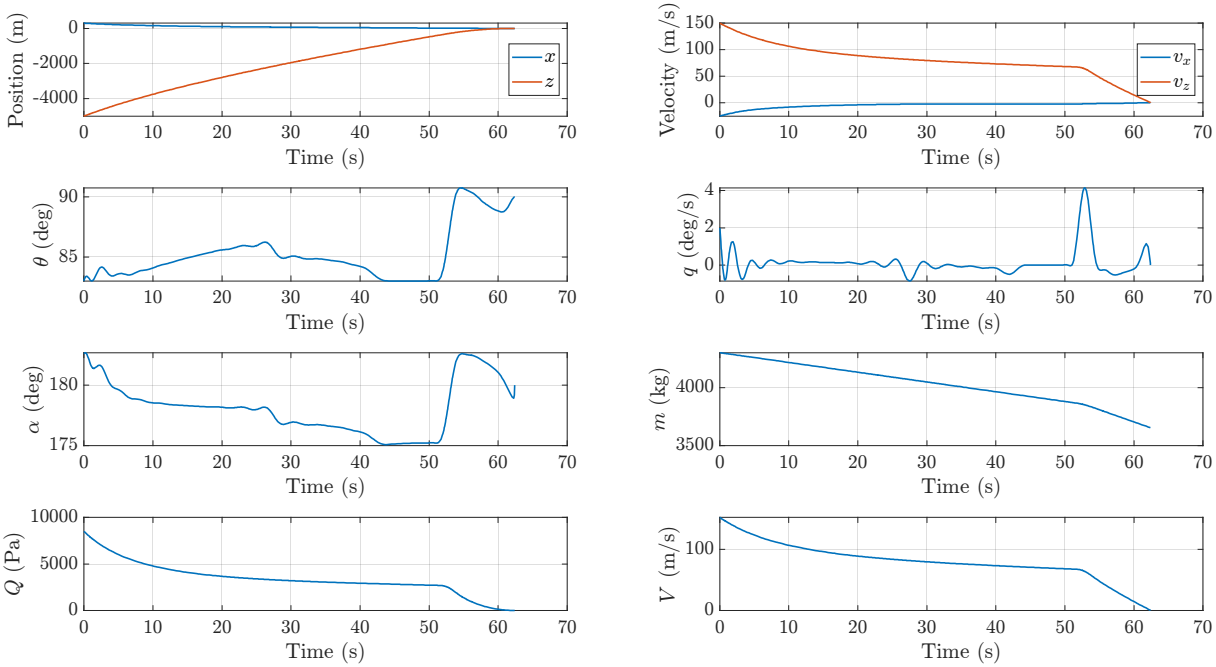


Fig. 3 Time evolution of trajectory parameters

The pitch angle also reaches a bound, going until its lower limit. The fuel-optimal trajectory therefore exploits the allowable pitch angle extreme to align the vehicle's horizontal position with the RP (as shown in Figure 4) while minimising propellant usage. During approximately the final 10 seconds, the vehicle tilts in the opposite direction to recover an upright attitude for touchdown. This interval coincides with the second phase (the thrust-maximum period), so the terminal portion of the trajectory serves two purposes: reducing velocity to ensure a soft landing and correcting the attitude so the vehicle contacts the ground in an upright orientation.

Examining the time evolution of the forces and moments in Figure 5, it can be noted that at the beginning

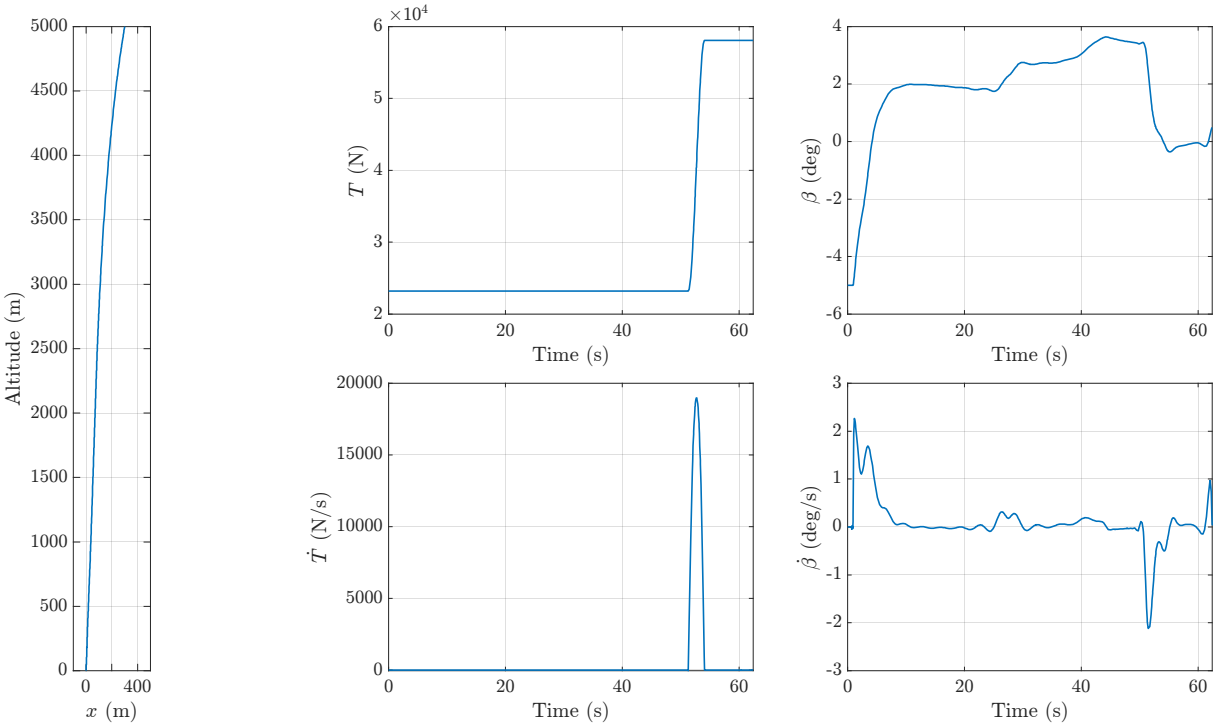


Fig. 4 Time evolution of trajectory position, control variables and derivatives

of the trajectory, there is an effort to adjust the Launch Vehicle (LV) from its initial position towards the horizontal position of the RP. This initial phase is characterised by higher force interactions, which gradually decrease as the vehicle approaches the desired alignment.

At the start of the trajectory, the main vertical force counteracting gravity is the aerodynamic lift. This aerodynamic contribution decreases rapidly at first and then more gradually, eventually becoming negligible in the final phase of the manoeuvre. This succession of events can be explained with the velocity profile shown in Figure 3, as it reflects the direct influence of velocity on the magnitude of aerodynamic forces. The vertical component of the thrust force is, as expected, related to the thrust magnitude profile in Figure 4, since most of the thrust action is perpendicular to the ground, contributing to overcoming gravity and controlling the descent.

Finally, an examination of the moments reveals that the aerodynamic and thrust-generated moments act in opposition. The sign of the aerodynamic moment depends on the vehicle's angle of attack, which can be verified by noting that each time α passes through 180 deg, the aerodynamic moment undergoes a change in sign. The thrust vector produces a moment about the CG with the opposite sign, which is used to counteract the aerodynamic pitch and to command attitude changes. The balance between these aerodynamic and thrust-generated moments determines the vehicle's tilt evolution and enables the sequence of movements for the fuel-optimal trajectory.

D. Linear State-Space Representation

In this section, a linearised state-space model is constructed to provide a starting point for controller development. For this process, it is necessary to compute the state space of the model for several points of the trajectory since these are needed to perform gain scheduling when applying the controller, allowing for the tuning of the controller at different moments of the mission. By showing the dynamics of the rocket in a linear state space form, the model is directly usable for controller synthesis and state space analysis.

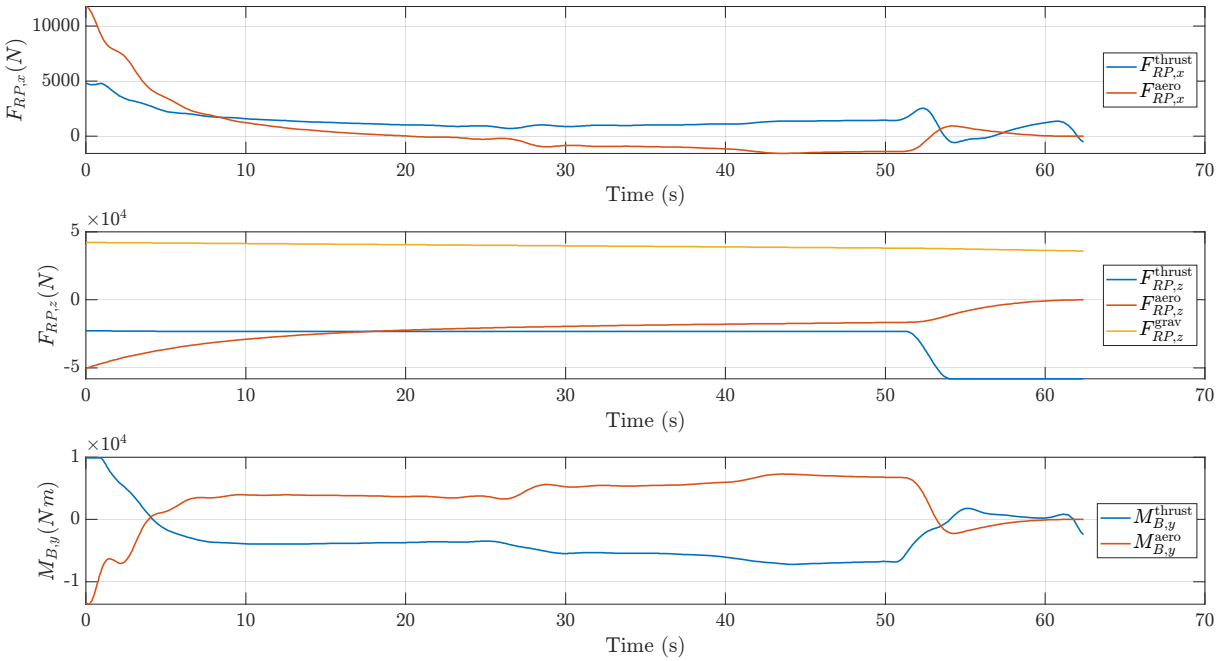


Fig. 5 Time evolution of forces and moments

This linearised system is obtained by expanding all quantities about the nominal flight path, obtained from the NLP, using small perturbations. More specifically, each state is written as the trim component plus a small time-varying perturbation ($f(t) = f_0 + \Delta f(t)$). With this linearised representation, apart from depending on the trimmed states, propulsion terms become functions of the trim control deflections ΔT_0 and $\Delta \beta_0$ while aerodynamic forces and moments, on the other hand, also depend on the trimmed aerodynamic coefficients.

To carry out the linearisation process, it is first necessary to linearise certain variables, such as the aerodynamic coefficients. Since these coefficients depend on both the angle of attack and the Mach number, these quantities must first be expressed in terms of the system states, as shown in Equation 12 and Equation 13. Additionally, to linearise the aerodynamic forces and moments, it is also necessary to express the dynamic pressure as a function of the system states, as done in Equation 14. The aerodynamic coefficients are then linearised about the equilibrium values of angle of attack and Mach number, also using the corresponding force gradient coefficients (e.g., for axial aerodynamic coefficient with respect to α: $\frac{\partial C_{\mathcal{A}}}{\partial \alpha} = C_{\mathcal{A}_\alpha}$). This procedure is illustrated in Equation 15 for the axial aerodynamic force coefficient, but the same approach can be applied to any of the aerodynamic coefficients.

$$\alpha = \theta - \gamma = \theta - \arctan\left(\frac{-v_{z,RP}}{v_{x,RP}}\right) \tag{12}$$

$$Mach = \frac{V}{c} = \frac{\sqrt{v_{x,RP}^2 + v_{z,RP}^2}}{c} \tag{13}$$

$$Q = \frac{1}{2}\rho V^2 = \frac{1}{2}\rho(v_{x,RP}^2 + v_{z,RP}^2) \tag{14}$$

$$C_{\mathcal{A}}(\alpha_0 + \Delta\alpha, Mach + \Delta Mach) \approx C_{\mathcal{A}}(\alpha_0) + C_{\mathcal{A}_\alpha}(\Delta\alpha) + C_{\mathcal{A}}(Mach_0) + C_{\mathcal{A}_{Mach}}(\Delta\alpha) \tag{15}$$

By applying the state and input partial derivatives to the nonlinear equations of motion present in Equation 5, the results shown in the appendix, subsection VII.A, are obtained. Thereafter, these partial derivatives

obtained in the appendix allow for the computation of Equation 16 in order to reach the state-space construction with the form exhibited in Equation 17. Since the outputs are defined to be the full state vector, **C** is taken as an identity matrix so that $\Delta\mathbf{y} = \Delta\mathbf{x}$. Moreover, there is no direct instantaneous dependence of the outputs on the inputs in the chosen measurement configuration, so **D** is set to be a zero matrix.

$$\underbrace{\begin{bmatrix} \Delta\dot{x}_{RP} \\ \Delta\dot{v}_{x,RP} \\ \Delta\dot{z}_{RP} \\ \Delta\dot{v}_{z,RP} \\ \Delta\dot{\theta} \\ \Delta\dot{q} \end{bmatrix}}_{\Delta\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial\dot{v}_{x,RP}}{\partial v_{x,RP}} & 0 & \frac{\partial\dot{v}_{x,RP}}{\partial v_{z,RP}} & \frac{\partial\dot{v}_{x,RP}}{\partial\theta} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{\partial\dot{v}_{z,RP}}{\partial v_{x,RP}} & 0 & \frac{\partial\dot{v}_{z,RP}}{\partial v_{z,RP}} & \frac{\partial\dot{v}_{z,RP}}{\partial\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\partial\dot{q}}{\partial v_{x,RP}} & 0 & \frac{\partial\dot{q}}{\partial v_{z,RP}} & \frac{\partial\dot{q}}{\partial\theta} & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \Delta x \\ \Delta v_x \\ \Delta z \\ \Delta v_z \\ \Delta\theta \\ \Delta q \end{bmatrix}}_{\Delta\mathbf{x}} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{\partial\dot{v}_{x,RP}}{\partial T} & \frac{\partial\dot{v}_{x,RP}}{\partial\beta} \\ 0 & 0 \\ \frac{\partial\dot{v}_{z,RP}}{\partial T} & \frac{\partial\dot{v}_{z,RP}}{\partial\beta} \\ 0 & 0 \\ \frac{\partial\dot{q}}{\partial T} & \frac{\partial\dot{q}}{\partial\beta} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} \Delta T \\ \Delta\beta \end{bmatrix}}_{\Delta\mathbf{u}} \tag{16}$$

$$\begin{aligned} \Delta\dot{\mathbf{x}} &= \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \\ \Delta\mathbf{y} &= \mathbf{C}\Delta\mathbf{x} + \mathbf{D}\Delta\mathbf{u} \end{aligned} \tag{17}$$

Open-Loop Pole Analysis

With the linear state-space computed, it is now possible to perform a linear analysis of the open-loop system in the nominal conditions. In Figure 6, the poles evolution during the previously computed trajectory is shown. Furthermore, these poles can be divided into 6 modes as represented in Figure 7.

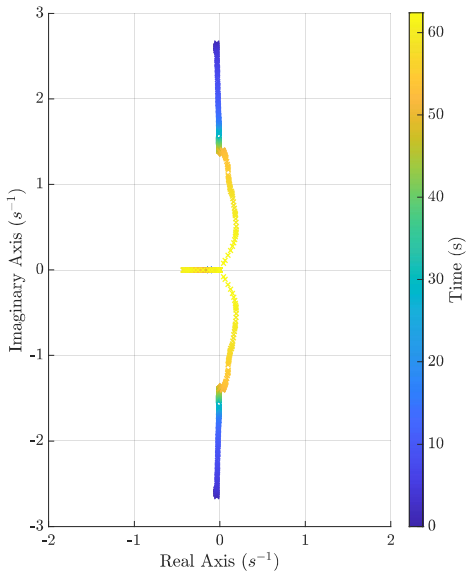


Fig. 6 Powered descent open loop pole evolution diagram

Firstly, it is possible to observe that modes 1 and 2 are always at the origin of the Gauss plane, the so-called integrator poles. Furthermore, Modes 3 and 4 are a pair of complex conjugates. These poles are unstable during approximately the final 10 seconds in the open-loop configuration since they are on the right side of the complex plane. Finally, modes 5 and 6 for every point of the trajectory are located on the left-hand side of the complex plane, so these modes are stable. Both modes are located on the real axis,

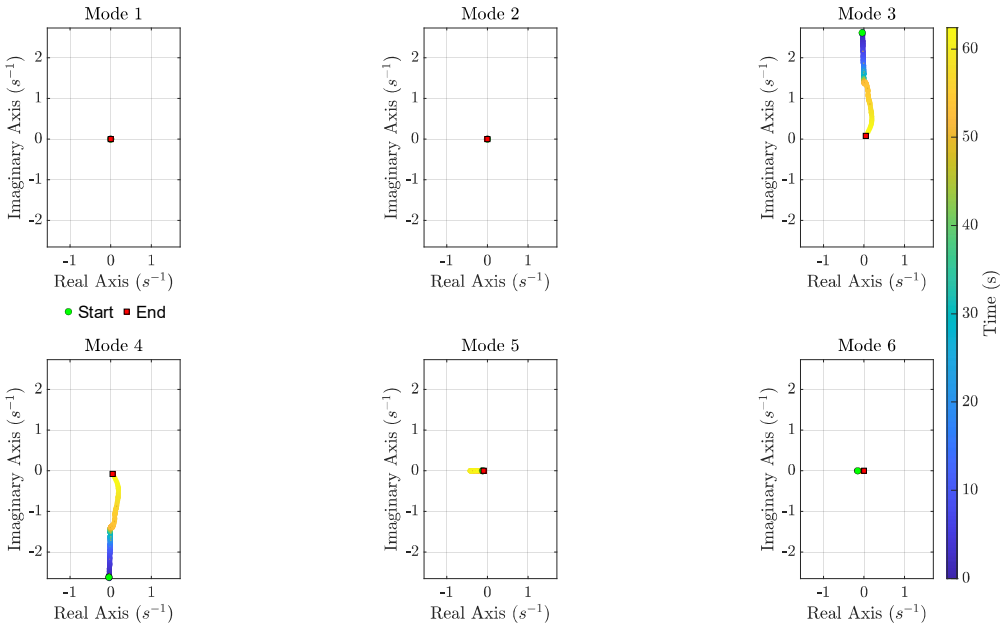


Fig. 7 Powered descent open loop pole modes evolution

corresponding to a damping ratio of 1.

E. Faults Description

Finally, to terminate this problem statement, the faults to which the rocket is subjected during the simulations must be defined:

1) TVC jamming – the affected thruster becomes locked at a fixed gimbal angle and can no longer change its TVC angle.
2) Partial engine thrust loss – the affected engine has a reduction in available thrust, where the magnitude is given by a percentage that indicates the faulty engine's available power. At a partial-fault level of 0%, the engine can still deliver its minimum thrust, while at 100% it behaves as a healthy engine.
3) Total engine thrust loss – the affected engine does not generate any thrust. It is considered completely inoperative.

III. Controller

The closed-loop architecture used in this work is illustrated in Figure 8. A desired state trajectory is given to the vehicle as guidance based on the results obtained in subsection II.C. The architecture is one of a classic control loop, where the measured vehicle state is subtracted from the reference to form an error signal, which is presented to the controller. The controller itself commands thrust level and a TVC deflection correction. These commands are added to the nominal feed-forward commands provided by the guidance and applied to the vehicle dynamics.

Furthermore, to apply the controller throughout the trajectory, gain scheduling is used. By tuning a set of controllers at different trajectory points, it is possible to take into account variations in the system behaviour according to the trajectory instant. In this case, five points are selected, and each of these controllers is tuned to provide the desired response at the corresponding point of the trajectory. In this system, all the states are assumed to be available, and so there is no need for an observer. Finally, interpolation between these points is used to guarantee continuous gain scheduling of the controller during the trajectory.

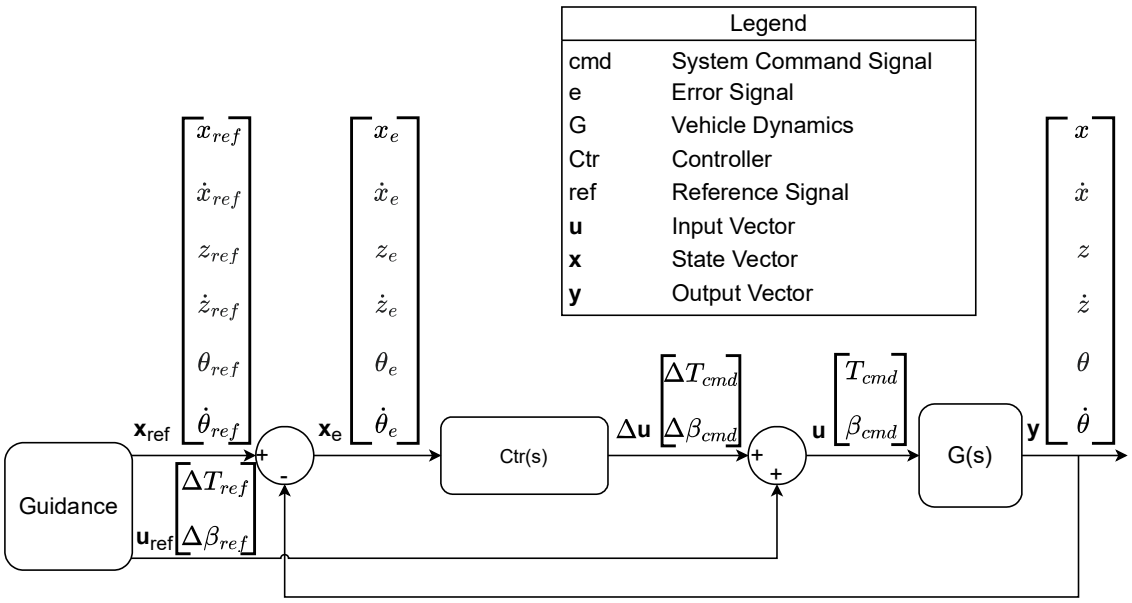


Fig. 8 Closed loop structure for LTI plant

A. Linear Quadratic Regulator

Firstly, in order to use LQR, a system needs to be stabilisable [21]. If a system is controllable, then it is by definition stabilisable, so for this work, it is verified that throughout the entire trajectory, the system is controllable. Moreover, this means that a feedback control always exists to place the poles anywhere desired. The objective of the LQR is to minimise the infinite-horizon quadratic cost function Equation 18, which includes **Q** and **R** positive semi-definite and positive definite weighting matrices of appropriate dimensions, respectively.

$$J = \int_0^\infty (\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u})\, dt \tag{18}$$

In order to keep the tuning not too complex for easy interpretation, the values of the **Q** and **R** matrices are selected to be diagonal, not including cross-coupling terms. The initial values of the weighting factors are determined using Bryson's rule, which provides a heuristic for selecting diagonal entries as $\frac{1}{\text{range}^2}$ where the range corresponds to the expected variation of each state or control input when tracking the reference trajectory, meaning the expected maximum errors. This approach offers a reasonable first approximation but still requires tuning. Consequently, the values of **Q** and **R** are subsequently adjusted by analysing both the linear system response at the five gain-scheduling points and the nonlinear simulation results. The final tuned matrix values are presented in Table 6.

Table 5 Bryson's rule denominators

Q						R	
den_x	$\text{den}_{\dot{x}}$	den_z	$\text{den}_{\dot{z}}$	den_θ	den_q	den_T	den_β
100^2	10^2	100^2	10^2	$\left(5\frac{\pi}{180}\right)^2$	$\left(2\frac{\pi}{180}\right)^2$	5000^2	$\left(10\frac{\pi}{180}\right)^2$

As shown in Table 5, the diagonal denominator entries of **Q** and **R** are selected in accordance with Bryson's rule. For the state-weighting matrix **Q** elements, the denominators are defined as the square of the maximum admissible state error. This maximum value is defined as the maximum initial condition error used in the Monte Carlo campaign (presented in subsection V.C), which is a conservative estimate of the largest state deviation expected along the trajectory. Similarly, for the input-weighting matrix **R**, the maximum predicted deviation of each control input from its nominal trajectory value is used and squared to form the denominator. To use these denominators in the LQR weighting matrices, they are grouped into vectors: $\mathbf{den_x} = \text{diag}\begin{bmatrix}\text{den}_x^{-1} & \text{den}_{\dot{x}}^{-1} & \text{den}_z^{-1} & \text{den}_{\dot{z}}^{-1} & \text{den}_\theta^{-1} & \text{den}_q^{-1}\end{bmatrix}^\top$ and $\mathbf{den_u} = \text{diag}\begin{bmatrix}\text{den}_T^{-1} & \text{den}_\beta^{-1}\end{bmatrix}^\top$.

Table 6 LQR weighting matrices

Gain Scheduling Point	Time (s)	Q	R
1	0	$\text{diag}[100, 100, 100, 100, 5, 20] \cdot \mathbf{den_x}$	$\text{diag}[10, 10] \cdot \mathbf{den_u}$
2	15.7	$\text{diag}[100, 100, 100, 100, 5, 20] \cdot \mathbf{den_x}$	$\text{diag}[10, 10] \cdot \mathbf{den_u}$
3	31.2	$\text{diag}[100, 100, 100, 100, 1, 1] \cdot \mathbf{den_x}$	$\text{diag}[1, 10] \cdot \mathbf{den_u}$
4	46.7	$\text{diag}[100, 100, 1000, 1000, 1, 1] \cdot \mathbf{den_x}$	$\text{diag}[1, 1] \cdot \mathbf{den_u}$
5	62.4	$\text{diag}[1000, 100, 1000, 1000, 1, 1] \cdot \mathbf{den_x}$	$\text{diag}[0.1, 1] \cdot \mathbf{den_u}$

These error values evolve throughout the trajectory as the controller works to decrease them. However, since this study addresses faulty cases, the evolution of maximum errors and control variables deviation from nominal cannot be predicted reliably. Therefore, the conservative approach of maintaining the maximum admissible error for the entire trajectory is adopted.

The choice of the numerator resulted from an iterative process, analysing the linear response at each gain-scheduling point as well as the performance of the controller in the nonlinear system. Generally, it can be observed that at the beginning of the trajectory, greater emphasis is placed on the attitude states (θ and q) to align the rocket with the trajectory and correct any initial attitude errors as quickly as possible. The nonlinear analysis showed that these errors are corrected relatively rapidly and that the majority of the considered faults do not critically affect attitude correction. Consequently, the relative importance of the attitude states given by the controller is reduced during the second part of the trajectory.

The states related to the translational movement, such as position and velocity, are always important to the controller. However, during the final phase, an emphasis is given to maximise the chances of landing in the RP and with acceptable velocity, in accordance with the safety criteria that are presented in subsection V.A. In addition, vertical position and velocity are prioritised earlier in the trajectory in comparison with the lateral components to help the rocket reach the ground with appropriate vertical velocity, a critical aspect of a rocket landing, as the timing must be precise to avoid excessive impact velocity or an ascent before the target landing point.

The reasoning used to decide the numerators of the **Q** matrix also influences the numerators for the **R** matrix. In particular, the **R** numerator values decrease towards the end of the trajectory to provide greater control authority, enabling a faster response while placing less emphasis on actuator limits, in order to guarantee the best chances for a soft landing.

IV. TVC Angle and Thrust Allocation Strategy

As Figure 8 shows, the controller's input is the error state vector, and it outputs commanded actuator deflections. These deflections consist of a vector containing the deflections for the single-engine configuration thrust magnitude and TVC angle. In this way, a desired net force and moment are given in a form that is independent of the actuator configuration, decoupling the high-level control objectives from the low-level

actuator implementation. Nevertheless, the control effort must be distributed among the actuators, in this case, the five available thrusters.

In this section, it is demonstrated how the nominal control allocation solution assuming healthy actuators is developed. This baseline allocator has two purposes: it performs the allocation whenever no faults are present, and it provides a benchmark for evaluating the performance improvements given by a fault-aware allocation algorithm. Thereafter, the fault-aware allocation is also described. This second allocation consists of a reconfiguration of the nominal case that makes the allocator better redistribute control effort among the remaining healthy engines when faults occur.

A. Nominal Allocation Algorithm

Problem 1: Nominal TVC angle and thrust allocation

Decision variables:

$$\mathbf{d} = \begin{bmatrix} \mathbf{f}_{x,B} \\ \mathbf{f}_{z,B} \end{bmatrix}, \quad \mathbf{f}_{x,B} = [f_{x,1,B}, \ldots, f_{x,n,B}]^\top, \; \mathbf{f}_{z,B} = [f_{z,1,B}, \ldots, f_{z,n,B}]^\top.$$

Objective:

$$J(\mathbf{d}) \; = \; \Big(F_{x,B} - \sum_{i=1}^{n} f_{x,i,B}\Big)^2 + \Big(F_{z,B} - \sum_{i=1}^{n} f_{z,i,B}\Big)^2 + \Big(M_{y,B} - \sum_{i=1}^{n} \big(-p_{x,i,B} f_{z,i,B} + p_{z,i,B} f_{x,i,B}\big)\Big)^2.$$

Subject to:

- *Box bounds:*

$$T_{\min} \cos(\beta_{\max,i}) \le f_{x,i,B} \le T_{\max},$$
$$-T_{\max} \sin(\beta_{\max,i}) \le f_{z,i,B} \le T_{\max} \sin(\beta_{\max,i})$$

- *Magnitude:*

$$f_{x,i,B}^2 + f_{z,i,B}^2 \le T_{\max}^2, \qquad i = 1, \ldots, n.$$

- *Gimbal cone:*

$$|f_{z,i,B}| \le f_{x,i,B} \tan(\beta_{\max,i}), \qquad i \in \{\text{outer thrusters}\}.$$

- *Central thruster special-case:*

$$\beta_{\max,1} = 0.$$

Initialisation: pick an initial guess $\mathbf{d}_0$ by distributing $F_{x,B}, F_{z,B}$ evenly and clipping to allowable bounds.
Solver: use a constrained nonlinear optimiser (`fmincon`).
Post-processing:

$$T_i = \sqrt{f_{x,i,B}^2 + f_{z,i,B}^2}, \qquad \beta_i = -\operatorname{atan2}(f_{z,i,B}, f_{x,i,B}), \quad i = 1, \ldots, n.$$

For the nominal allocation algorithm, the first step is to transform the input total thrust magnitude and TVC angle into forces and moments in the body frame: $F_{x,B}$, $F_{z,B}$, and $M_{y,B}$. This transformation is necessary because the input is defined as if the rocket had a single central engine with the combined power of the five engines used in the configuration, hence the need for allocation.

Once this computation is completed, the objective function can be defined. It consists of minimising the squared sum of the differences between the desired forces and moments and those generated by the set of rocket engines. Furthermore, the lower and upper bounds of the decision variables must be defined. Thereafter, the associated constraints must also be specified. These include the total thrust magnitude and

the maximum gimbal angle. In this case, all engines are assumed to be identical, so the constraints are the same for all n engines, except for engine 1, which is identical but is constrained to a fixed TVC angle of 0 deg.

To employ an allocation method, initial guesses are usually required. To provide this guess, the desired forces among the thrusters are uniformly distributed and then constrained to the component's operational limits.

After determining the final forces produced by each component that minimise the cost function, the corresponding thrust magnitudes and TVC angles for each element are computed. These are then used to command the individual actuators in the nonlinear simulator. A simplified view of the entire algorithm can be seen in Problem 1, which is obtained based on the work of [15].

B. Fault Aware Allocation

In the case of a fault, the objective is to reconfigure the control allocation algorithm so that the healthy components of the launch vehicle can be used to compensate for the faulty ones. The goal is to satisfy the controller's command as closely as possible to approximate its behaviour to the desired one, as much as the remaining healthy actuators allow.

This would require that the available thrust of each engine be known, as well as any jamming occurrence. Therefore, there is a need for a fault detection and diagnosis system that isolates the fault, the FDI system. For this project, the system is assumed to operate perfectly, providing information without delay, as the focus is on fault-tolerant control rather than fault detection. The system is therefore responsible for identifying the type, magnitude, and location of any fault in real time.

Problem 2: Fault-aware TVC angle and thrust allocation (reconfiguration of Problem 1)

For each thruster i, define the available maximum thrust $T_{\text{max avail,i}}$ as:

$$T_{\text{max avail,i}} = \begin{cases} T_{\text{max}}, & \text{healthy,} \\ T_{\text{max avail,i}} < T_{\text{max}}, & \text{partial loss,} \\ 0, & \text{total loss.} \end{cases}$$

Box bounds:

$$T_{\text{min}} \cos(\beta_{\text{max,i}}) \leq f_{x,i,B} \leq T_{\text{max avail,i}},$$

$$-T_{\text{max avail,i}} \sin(\beta_{\text{max,i}}) \leq f_{z,i,B} \leq T_{\text{max avail,i}} \sin(\beta_{\text{max,i}})$$

Magnitude:

$$f_{x,i,B}^2 + f_{z,i,B}^2 \ \leq \ \left(T_{\text{max avail,i}}\right)^2.$$

Gimbal cone:

If TVC free: $|f_{z,i,B}| \leq f_{x,i,B} \tan(\beta_{\text{max,i}})$.

If TVC jammed at $\beta_{\text{jam,i}}$: $f_{z,i,B} + f_{x,i,B} \tan(\beta_{\text{jam,i}}) = 0.$

With this information, the control allocator can rewrite the box constraints and thrust limits of the affected engines, or fix the TVC angle of a jammed actuator to its stuck position, as can be seen in the reconfiguration of the nominal problem in 2.

To verify and understand the behaviour of the allocation algorithms, the system responses have been plotted for the nominal trajectory, as shown in Figure 9. The nominal allocation algorithm is applied to a healthy case, while the fault-aware allocation algorithm is applied to a faulty scenario.

Beginning with the healthy case, it is to be noticed that the thrust behaviour closely resembles the nominal trajectory behaviour, but at a reduced magnitude because of the distribution of thrust across all five engines. Similarly, the TVC angles exhibit comparable patterns. However, they reach higher values than the one of the optimised trajectory due to the constraint imposed on thruster 1, which remains fixed. This restriction forces the remaining thrusters to compensate for the inactivity of thruster 1.

The faulty case considered here included each of the three faults described in subsection II.E:
- Jamming of engine 3 at 1 degree at 20 seconds.
- Total loss of engine 2 at 40 seconds.
- Partial loss of 80% of engine 4 at 55 seconds.

It is possible to understand how the fault-aware allocation deals with the fault occurrences by analysing the actuators' reaction. The jamming fault does not produce a significant change in the thrust levels, but it does affect the remaining TVC angles. This shows that the allocator is attempting to correct the moment while maintaining the commanded overall forces.

The total loss of engine 2 at 40 seconds produces a change in both the thrust levels and the TVC angles, with a sudden increase in both at the instant of the fault occurrence.

Finally, engine 4 experiences a partial thrust loss: from 55 seconds onward, its output falls from the original maximum thrust to the fault-constrained maximum thrust. The actuator's response doesn't change except for the one of the fault-affected engine since the remaining engines are already operating at their maximum thrust.

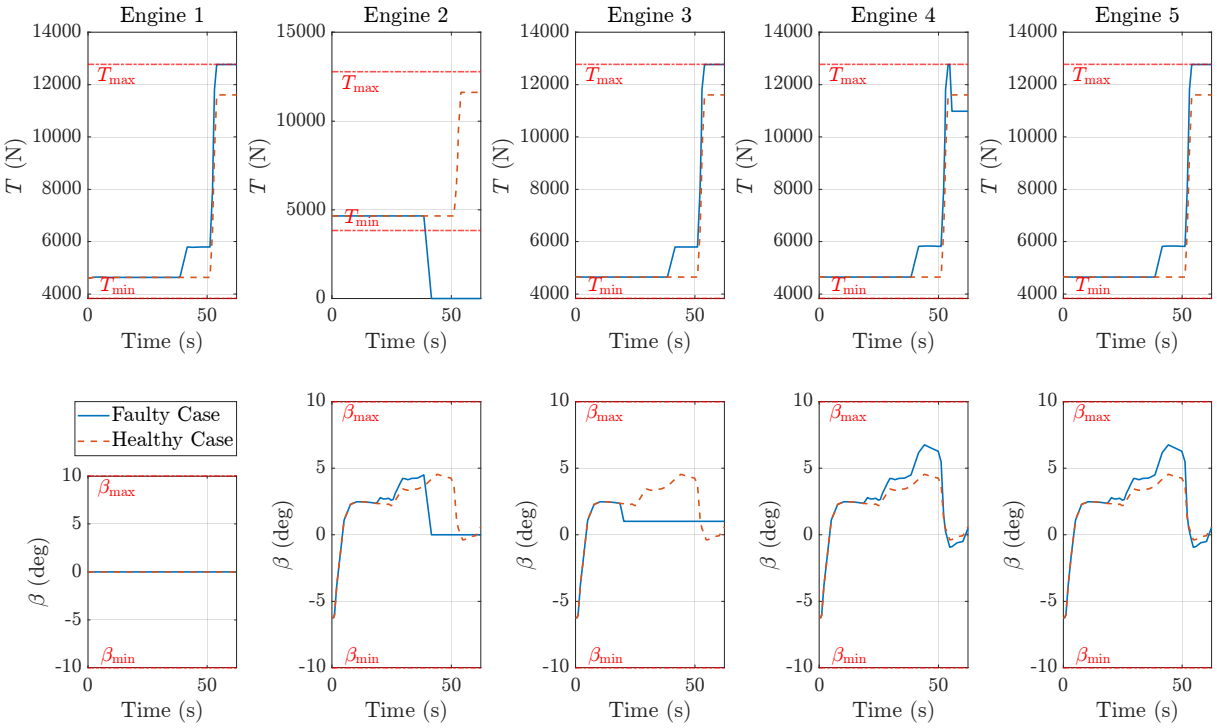


Fig. 9 Healthy and faulty cases allocation comparison for nominal trajectory

V. Monte Carlo Campaign

To evaluate the entire framework on a nonlinear model subjected to varying conditions, a Monte Carlo campaign is developed. In this section, the structure of the campaign is described, and the results obtained are presented and analysed.

A. Success and Simulation Stop Conditions

To prepare the MC campaign, the conditions that define the success or failure of a run of the rocket are defined, as well as the conditions that terminate the simulation. With these criteria defined, it is possible to guarantee the possibility to analyse and compare the success rate across different scenarios and make sure the simulations do not continue in irrelevant scenarios, stopping the simulation before that happens.

The stop conditions are defined as follows:

- Ground contact - If the LV reaches an altitude of 0 m, it is considered to have landed, and the simulation is terminated.
- Time limit - If the simulation has not been concluded by the end time of the nominal trajectory plus seven seconds, it is terminated to avoid unnecessary computation in divergent scenarios.
- Attitude limit - If the pitch angle deviates by more than 90 deg from the upright position (i.e., becomes parallel to the ground), the simulation is stopped, as such a configuration indicates loss of attitude authority.
- Velocity constraint - If the vehicle exhibits positive vertical velocity (i.e., begins going upward), the simulation is terminated. This behaviour is not expected during descent, and it indicates that the controller is not as ”focused” as it should be in the vertical states. This may happen if other errors are too big, which can be the case in a faulty situation.

As aforementioned, the other criteria that have to be defined are the success parameters. For each run, these parameters, which consist of six conditions, are utilised, and all must be satisfied so a run can be considered successful. These conditions, based on the rocket states at landing, are summarised in Table 7.

Table 7 Rocket landing success conditions

Condition	Success Description	Success Criterion
Ground contact	Rocket reaches the ground.	$\lvert z_{RP,f}\rvert = 0$ m
RP landing	Rocket lands within the designated RP radius.	$\lvert x_{RP,f}\rvert \leq 10$ m
Horizontal velocity	Horizontal velocity at touchdown is within safe limits.	$\lvert v_{x,RP,f}\rvert \leq 0.75$ m/s
Vertical velocity	Vertical velocity at touchdown is within safe limits.	$v_{z,RP,f} \leq 1.5$ m/s
Pitch angle	Rocket is upright within 1 deg of vertical at landing.	$\lvert 90 - \theta_f\rvert \leq 1$ deg
Pitch rate	Angular rotation rate around pitch axis is within safe limits.	$\lvert q_f\rvert \leq 1.5$ deg/s

B. Touchdown Safety Mechanism

To ensure that the rocket does not approach an excessively low vertical velocity during the landing phase, guaranteeing that the ground is reached within an acceptable time frame, a touchdown safety mechanism is implemented. The threshold values that activate this mechanism are derived from an analysis of multiple flight simulations and from identifying the scenarios in which such intervention is beneficial.

Two trigger conditions are defined. The safety mechanism activates if the vertical velocity becomes too small, specifically if it drops below 0.1 ms^{-1} at any point during the trajectory, or if, at a height of one meter

above the ground, the vertical velocity falls below 1 ms^{-1}. These safety mechanism exists for cases where the faults may cause the controller to command unrealistically low descent rates.

If either trigger condition is met, the engines generate a total thrust equal to 99% of the gravitational force, or if the capability of the remaining healthy engines is not enough, the closest value possible guaranteeing that the rocket maintains a controlled descent and achieves a smooth and safe touchdown.

C. Monte Carlo Campaign Organisation

With the objective of evaluating the proper functioning of the system and the impact of a fault-aware control allocation across several scenarios, with and without faults and subjected to different fault types, a Monte Carlo campaign is performed. This subsection describes how that campaign is organised. Each Monte Carlo batch of runs included one thousand runs, each executing the proposed powered descent and landing mission. The batches that are run are the following:

- MC 1 - Fault-free scenario.
- MC 2 - TVC jamming scenario, with a non-fault-aware control allocation.
- MC 3 - TVC jamming scenario, with a fault-aware control allocation.
- MC 4 - Partial engine thrust-loss scenario, with a non-fault-aware control allocation.
- MC 5 - Partial engine thrust-loss scenario, with a fault-aware control allocation.
- MC 6 - Total engine thrust-loss scenario, with a non-fault-aware control allocation.
- MC 7 - Total engine thrust-loss scenario, with a fault-aware control allocation.

Each scenario is either fault-free or represents a specific fault type. For faulted scenarios, the fault magnitude, the time of occurrence, and the engine where the fault occurs vary randomly for each run. The time of occurrence is selected within the nominal mission time interval, and the magnitude is between the allowed values. Furthermore, the initial conditions and some LV parameters are randomised, as are the fault occurrences, as shown in Table 8.

Table 8 Initial conditions, parameter and fault uncertainties

Property	Dispersion	Units
$\Delta r_{x,z,RP,0}$	$\mathcal{U}(-100, 100)$	m
$\Delta v_{x,z,RP,0}$	$\mathcal{U}(-10, 10)$	m/s
$\Delta\theta_0$	$\mathcal{U}(-5, 5)$	deg
Δq_0	$\mathcal{U}(-2, 2)$	deg/s
Δm_{dry}	$\mathcal{U}(-2\%, 2\%)$	kg
$\Delta I_{yy,\text{dry}}$	$\mathcal{U}(-5\%, 5\%)$	kg m^2
$\Delta C_{\mathcal{A},m,\mathcal{N}}$	$\mathcal{U}(-10\%, 10\%)$	-
$\Delta\rho$	$\mathcal{U}(-10\%, 10\%)$	kg/m^3
T_{avail}	$\mathcal{U}(0\%, 100\%)$	N
β_{jam}	$\mathcal{U}(-10, 10)$	deg
$t_{\text{fault occurence}}$	$\mathcal{U}(0, t_f)$	s
$\text{Engine}_{\text{fault}}$	$\mathcal{U}\{1, 2, 3, 4, 5\}$	-

In this work, all properties are subjected to a uniform distribution in order not to understudy any region of the chosen envelope. The primary purpose of this Monte Carlo campaign is to examine system behaviour across the entire range defined for each parameter. A uniform distribution ensures similar sampling density throughout that envelope, allowing model responses to be explored equally throughout the envelope.

D. Results and Analysis

A small note about the analysis: it is important to make clear that some of the images mentioned in the text, the ones related to the landing conditions and the comparison of the influence of the fault's magnitude and timing on the success of the mission, are presented in the appendix (subsection VII.B) in order to achieve a better organisation of the article.

In Figure 10, it is possible to observe a general analysis of the MC campaign, which shows the MC success rates for the powered descent and landing mission of the seven batches. These results lead to four principal conclusions.

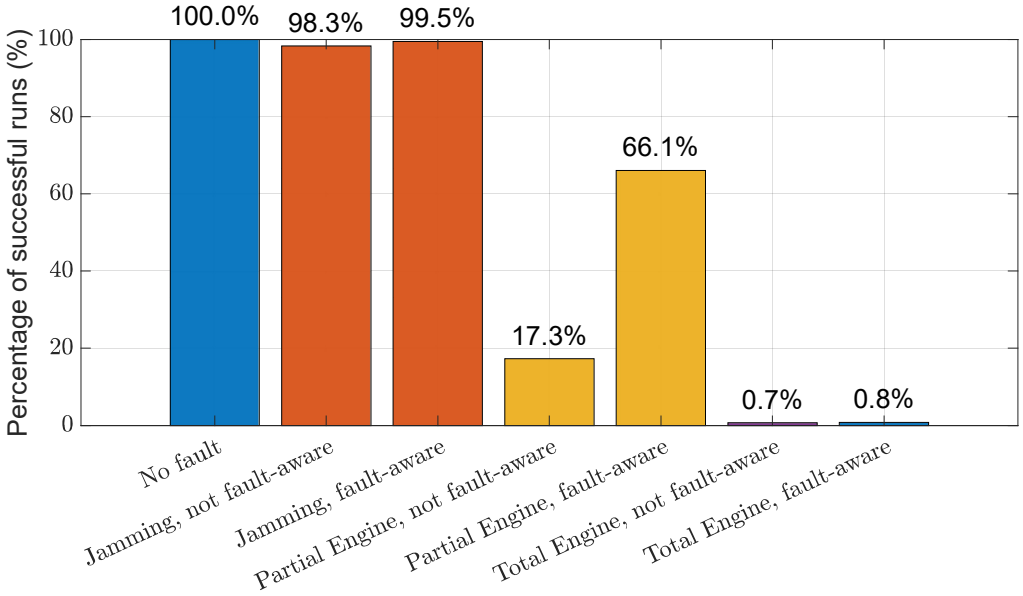


Fig. 10 Monte Carlo campaign general comparison

Firstly, it is possible to conclude that the healthy system performance is acceptable since, in the absence of faults, every simulated run succeeds, demonstrating that the baseline guidance, control, and allocation models produce reliable descent and landing behaviour across the entire varying conditions envelope.

Secondly, the TVC jamming fault is well tolerated. Both the not-fault-aware and fault-aware cases achieve very high success rates (98.3–99.5%), with the fault-aware allocation having a small improvement in comparison. This suggests that, for jamming faults, the applied control/allocation solution is adequate.

Thirdly, the partial engine-loss regime is where the fault-aware strategy shows the biggest improvement: success rises from $\sim$17% without fault awareness to $\sim$66% with it. This success improvement demonstrates that reconfiguration of the allocation substantially increases the recoverability probability. Nevertheless, looking at the case where the fault-awareness algorithm is used, approximately one-third of the cases still fail. This reveals that even this technique does not solve all the problems.

Lastly, in the case of total engine loss, the success rate is close to zero ($\sim$0.7–0.8%) in both the allocation cases. This fault case is studied in detail in subsubsection V.D.4, but this indicates that it is probably going to be considered unrecoverable, since the healthy actuators do not seem to be enough to land the rocket successfully.

1. Fault-Free Scenario

As already mentioned, the success rate in the fault-free scenario is 100%. The landing states of this batch of runs are illustrated in Figure 16, together with their acceptable boundaries. These results confirm that every landing is well within the limits. In particular, the final lateral position is always at approximately 1 m radius from the centre of the RP, comfortably inside the 10 m pad tolerance. Regarding velocities, the vertical component is never too close to zero, due to the touchdown-safety mechanism (as described in subsection V.B).

To understand the system behaviour, the evolution of the states can be analysed, or, as shown in Figure 11, the evolution of the state errors. A global inspection indicates that the control system makes all the state errors converge to zero along the trajectory.

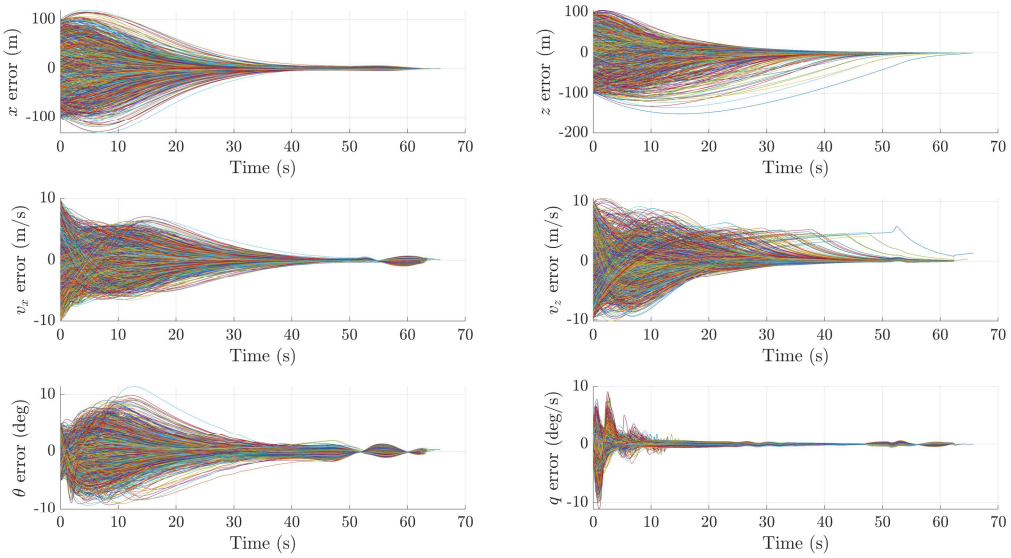


Fig. 11 Fault-free scenario state error evolution

A more detailed examination confirms the expected behaviour of the LQR controller, according to the weighting choices and analysis provided in subsection III.A. The attitude-related states converge to near-zero error faster than the position and velocity states, particularly their vertical components. Moreover, it is possible to observe slight non-smooth behaviour around zero error, especially in the attitude states and the horizontal velocity component. One contribution to these small oscillations comes from the controller changing priorities as the vehicle approaches touchdown, giving more importance to the translational and, specifically, vertical translational components. The LQR increasingly penalises deviations in vertical states, leading to more aggressive corrective actions. While the controller shifts focus, small oscillatory corrections are produced.

Finally, considering the actuator response in Figure 12, it can be noticed that during the first part of the trajectory, in some runs, both the engine thrust commands and the TVC angles saturate. This occurs because the controller must compensate for the state errors introduced by the initial condition uncertainties. As these errors are reduced along the trajectory, the actuator demands move away from their limits, and the system begins to operate closer to the nominal guidance profile. The abrupt drop in commanded thrust observed at the end of some runs corresponds to the activation of the touchdown safety mechanism.

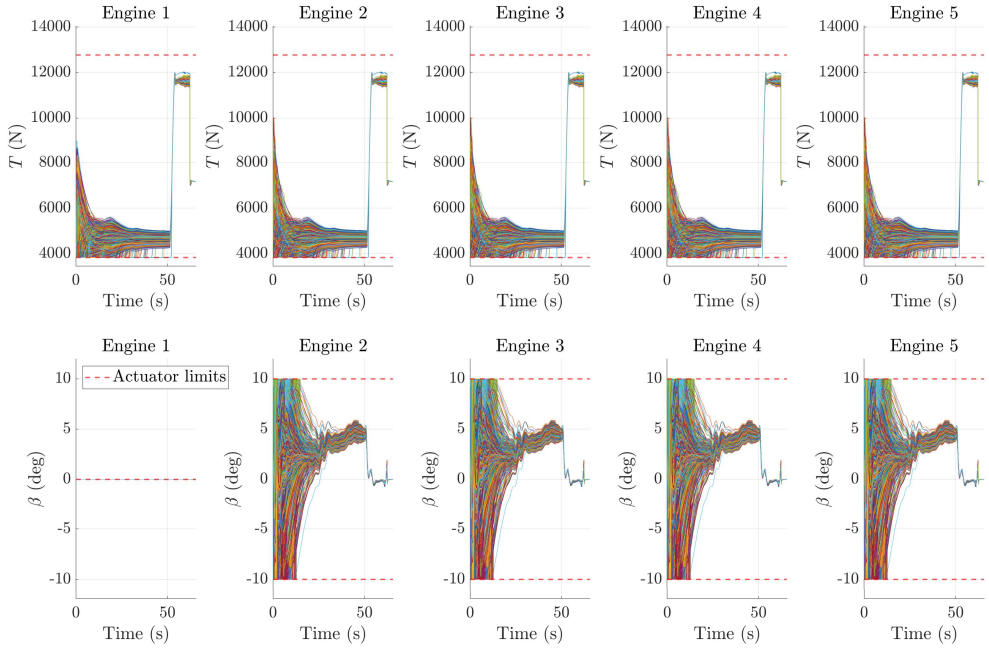


Fig. 12 Fault-free scenario actuator action evolution

2. Jamming Scenario

As shown in Figure 10, the jamming-fault scenario exhibits a very high success rate. Nevertheless, a small number of runs still fail, and these failures occur only under very specific conditions. As illustrated in Figure 22 and in Figure 23, both the non-fault-aware and fault-aware approaches exhibit failures when the jamming occurs at the extreme ends of the jamming-angle range.

In the non-fault-aware case, failures appear at both extremes, with a clear predominance in the negative limit of the gimbal angle. Moreover, all failed runs have a fault starting during the first half of the mission, specifically before the controller demands the maximum positive β angle, as can be observed in Figure 4. When one engine becomes stuck at a negative β_{jam}, the vehicle sometimes is unable to reach the required attitude, the affected engine counters the necessary corrective moment, causing the LV to deviate from its commanded orientation and demanding a higher action from the remaining healthy engines, which can affect the following of the trajectory. In one of the failed cases, this results in the pitch angle diverging by 90 deg from the upright position, which immediately terminates the mission and considers it a failure. In the remaining failures, the vehicle does not lose control catastrophically but reaches a state from which it can no longer achieve a landing that satisfies the success criteria.

Considering, on the other hand, the fault-aware case, several of the runs that previously failed are now successful, demonstrating the benefit of adapting the fault-aware allocation to the jammed engine. However, the method is still not fully effective in all circumstances. The one already mentioned run still experiences a complete loss of attitude control, indicating that even with fault-awareness, the developed system cannot always compensate for extreme jamming conditions.

Furthermore, the results can also be analysed from the perspective of which engine experiences the fault (note that in this problem formulation, engines 2 and 4 can be considered equivalent, since their only difference is their position along the y-axis, which is outside the problem plane). In the non-fault-aware case, among the failures occurring at a negative jamming angle $\beta_{\text{jam}} < 0$, 53.33% correspond to faults in engine 5, while the remaining cases involve engines 2 or 4. For the two failures occurring at a positive jamming

angle $\beta_{\text{jam}} > 0$, one is caused by a fault in engine 3 and the other by a fault in engine 4. In the fault-aware case, the pattern is even more noticeable: 80% of the failures originate from a jamming in engine 5, with the rest involving engine 4. This distribution is not random, and it can be explained by examining the moments generated when an engine becomes jammed. When $\beta_{\text{jam}} < 0$, the thrust vector's z_B-component produces a positive moment about the vehicle's CG. One of the primary components counteracting this moment is the x_B-component of thrust produced by engine 5. However, when engine 5 is jammed at a large negative angle, this compensating component is reduced, making the vehicle more susceptible to loss of attitude control. A symmetric argument applies for the opposite jamming direction, even though, due to this specific mission trajectory, it is not as crucial.

Consequently, when jamming occurs, the lateral engines (3 and 5) have a greater influence on vehicle performance because the dominant effect of the fault is on the moment generated. This, in addition to the mission at hand, explains why failures mainly occur with faults in engine 5 in both awareness scenarios and why even with fault-awareness, it is complicated to solve some fault cases in this engine.

Examining the landing conditions (Figure 17 and Figure 18), and noting that the run in which attitude control is completely lost is excluded since it did not result in a landing, it becomes evident that the unsuccessful cases fail due to deviations in the final horizontal velocity and pitch angle. It is also important to note that these deviations are very close to the success boundaries, particularly in the fault-aware scenario, where the final landing states are improved in comparison with the non-fault-aware case. This improvement in the terminal states helps explain the higher success rate achieved with the fault-aware strategy.

More in general, the results indicate that the jamming scenario is, in most cases, effectively managed by the used system. However, difficulties appear in extreme conditions, particularly when large TVC jamming angles, specific fault-occurrence times, and certain engine locations coincide. The introduction of fault awareness slightly improves the system's ability to deal with this fault.

3. Partial Engine Loss Scenario

For the partial engine loss scenario, as their success rates differ by a big margin, both the non-fault-aware and fault-aware cases are analysed in detail in this subsection.

To identify the main cause of unsuccessful runs in this faulty case, inspection of Figure 19 and Figure 20 can be made to observe the landing states of the not-fault-aware and fault-aware cases, respectively. This shows that the final lateral position remains within RP limits for all simulations. The failures are then related to the final velocities and final attitude states, where a significant number of cases are outside the predefined success bounds. In particular, the vertical velocity component is the most affected. This result is understandable because the severity of the fault can exceed the capability of the remaining healthy actuators to compensate. With a sufficiently severe fault, the rocket can not be able to reduce the descent rate enough in order to allow for a smooth landing.

In Figure 13, the actuator responses for the fault-aware case are shown, and give a justification for the previously mentioned landing results. The actuator analysis confirms that, for some runs, the required thrust pushes the engines to their saturation limits near the end of the trajectory, where the nominal trajectory already demands high thrust. Similarly, in the TVC angle behaviour, it also reaches saturation in several cases. However, this does not necessarily reflect difficulty in achieving the commanded moment, as the TVC angle behaviour up to approximately 50 seconds closely matches the fault-free case shown in Figure 12. However, once thrust saturation occurs, the control allocation algorithm prioritises generating maximum vertical force over satisfying the full moment requirement, since the force error becomes dominant. As a result, the TVC angle shifts towards configurations that maximise vertical thrust, even if this compromises attitude control. This helps explain the larger dispersion in final attitude states in the fault-aware simulations compared with the non-fault-aware ones.

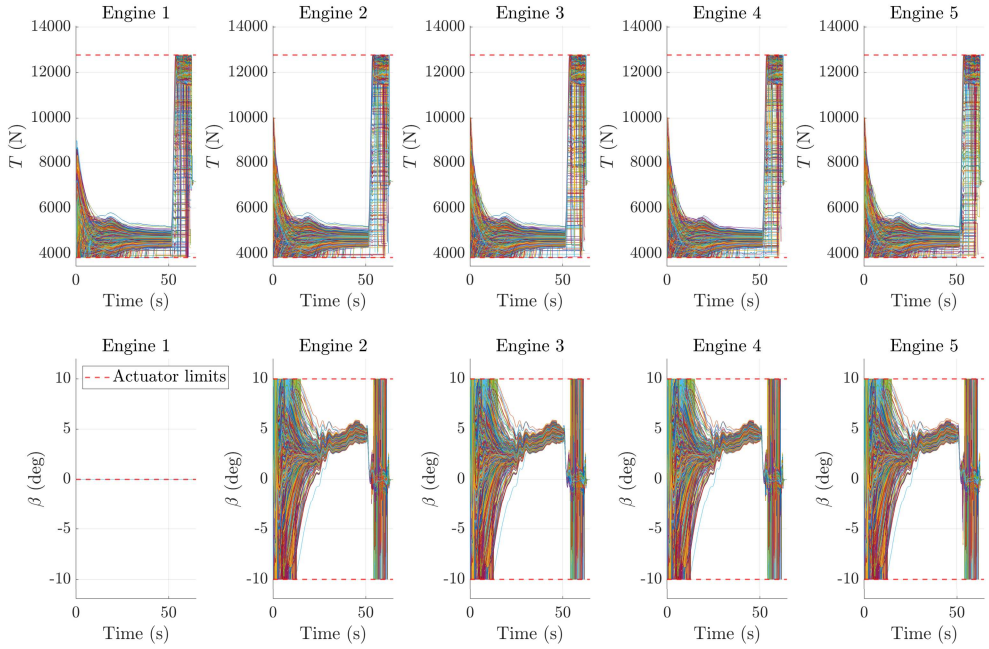


Fig. 13 Fault-aware partial engine loss scenario actuator action evolution

Having established that critical faults do occur, the next step is determining in what situations these faults can happen, as is done for the jamming scenario. An overview of the successful cases as a function of the faulty engine's available power and the time of occurrence is shown in Figure 24 and Figure 25. From these plots, it can be concluded that the fault-occurrence time has little influence on mission success. The only situations where time seems to matter are those in which the fault occurs so late in the trajectory that it has no practical impact on the landing outcome, therefore, no further analysis is done on this characteristic of the fault.

Focusing on another fault characteristic, the magnitude, it is possible to conclude that it has an actual influence on the success of the mission. A threshold at around 40% is visible for the fault-aware case and around 90% for the non-fault-aware case. Below this percentage, the success of the mission becomes highly unlikely, which indicates that, around that value, the remaining power provided by the engines may not be enough to perform a safe landing.

To better define this threshold, a plot, which in this paper is named "success curves", is used in order to visualise the probability of success. In such plots, the percentage available power is divided into several equal intervals, and inside these, the probability of success is computed according to the number of successful and attempted cases. Furthermore, confidence intervals are computed for each of these intervals using the Wilson confidence interval for a confidence of 95%, since this technique is appropriate for binomial proportions [22], and so appropriate for the current case with the success rates.

"Success curves" provide a good visual indication of the threshold. Nevertheless, a complementary estimate is obtained by combining model predictions with statistical analysis in order to obtain actual values. This estimate is obtained by building a regression model from the results of each engine. With this regression model constructed, the average time at which the faults occur is used ($t = 31.04$ s), since, as has been seen, the time does not have a relevant role in the results unless it happens too late in the trajectory. Using this model, the available-power threshold that guarantees a target success probability of 95% is computed. Finally, the results for each engine are averaged to obtain a single value that includes all the engine situations. The purpose of this approach has two objectives: first, to understand if faults in different engines lead to

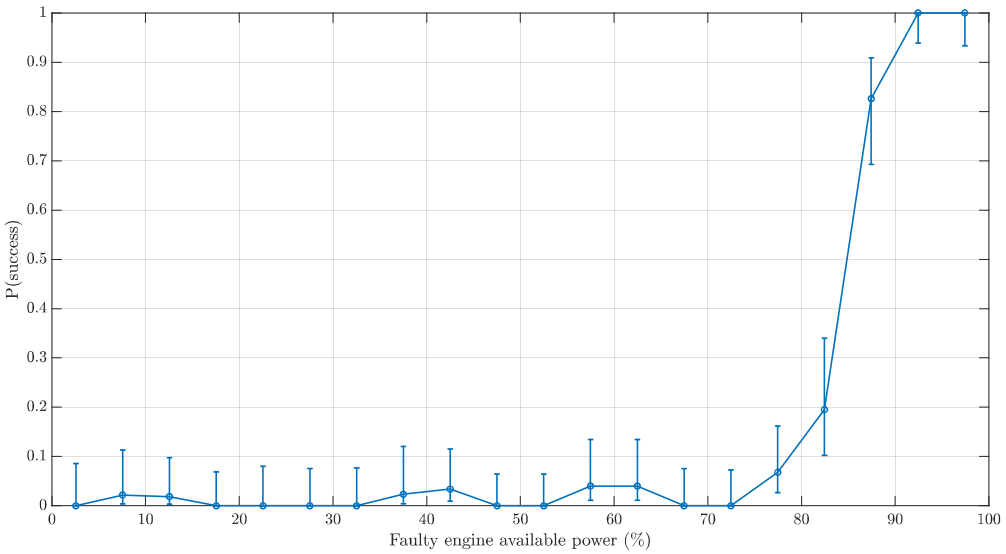


Fig. 14 ”Success curve” with confidence intervals relative to faulty engine available power in non-fault-aware partial engine loss scenario

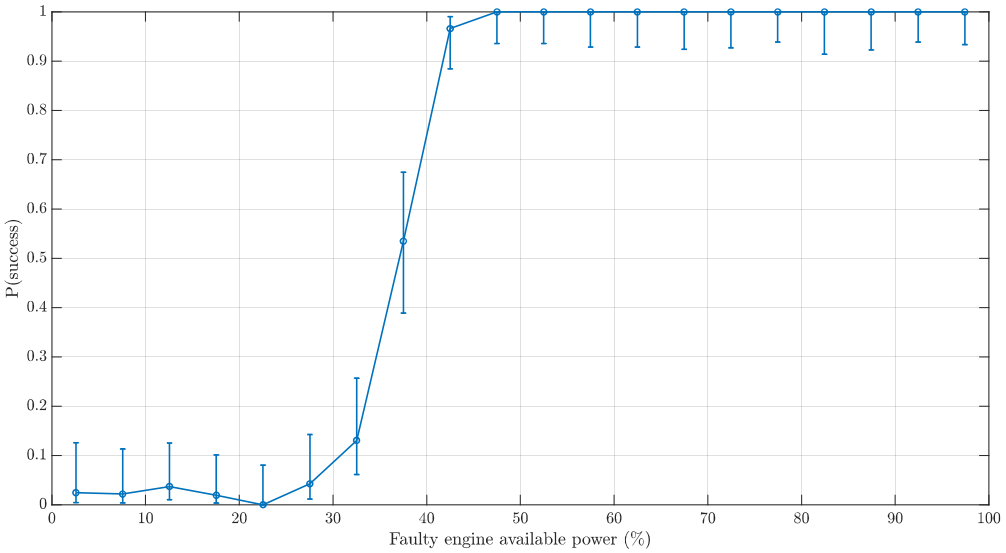


Fig. 15 ”Success curve” with confidence intervals relative to faulty engine available power in fault-aware partial engine loss scenario

different threshold values, and second, to isolate the effect of available power by fixing the fault-occurrence time at a value. This method allows the estimated threshold to reflect only the influence of the individual engine and available power, independent of when the fault occurs. The results of this method are presented in Table 9 and result in a value which is mentioned as Critical Available Power (CAP).

As observable in Table 9, the success rates obtained for the fault in each individual engine are very

Table 9 Available power thresholds for mission success

Engine	Non-fault-aware CAP (95% success)	Fault-aware CAP (95% success)
1	100.35%	47.26%
2	99.51%	44.13%
3	95.36%	46.14%
4	98.54%	46.04%
5	97.57%	45.09%
Average	98.29%	45.78%

close to each other, and their influence varies in the two different awareness scenarios, which indicates that no engine can be highlighted as more critical than the others. Therefore, from these data, nothing can be concluded about the influence of each engine. The lower CAP values of some engines can be attributed to the stochastic nature of the Monte Carlo simulations rather than to any difference in their true probability of success. Nevertheless, to strengthen this conclusion, an additional Monte Carlo campaign is carried out (not presented in this paper due to space constraints) in which faults are introduced in specific engines for the entire batch. These supplementary simulations again produced very similar success percentages across all engines, and the ordering of success rates differed from that suggested by the results in this table, which confirms that the engine location does not appear to have an influence on this fault.

Comparing all the results, it can once again be confirmed that the fault-aware allocation in this fault greatly improves the success rate.

4. Total Engine Loss Scenario

The final considered scenario is the total engine loss. The same reasoning used for the partial engine loss, which stated that there is a critical available power below which the mission is led into failure, can be applied to this case. Since the total thrust of one engine is lost, all runs are below this critical value in an even more severe way, as can be confirmed by the analysis of the final vertical velocities observed in Figure 21, which are even higher than the ones observed for the partial engine loss in Figure 20. This is expected since the total thrust produced is even lower, and therefore, it has more difficulty counteracting the gravity force.

Nevertheless, an examination of Figure 10 shows that the success rate does not drop completely to zero. This behaviour is verified and found to result from faults occurring very late in the trajectory, at a point where the vehicle is already close to landing and can still meet the success criteria even with the fault. In these rare cases, the fault-aware strategy is able to recover one additional successful outcome compared with the non-fault-aware approach out of the thousand simulations. However, this marginal improvement does not change the overall conclusion that for this type of fault, the control solution is fundamentally unable to recover the launch vehicle, and mission failure is effectively unavoidable.

VI. Conclusion

This paper investigates the influence on performance and soft landing ability of an FTC allocation algorithm in comparison with the nominal one, and how different faults affect an RLV powered descent and landing mission. Based on this study, the obtained conclusions are presented in this section. It is important to notice that although the results are influenced by the specific launch vehicle and mission profile considered, the general trends are expected to extend to other RLVs operating in powered descent and landing scenarios.

It is also important to note that the results are obtained using the previously described LQR controller. Nevertheless, different control techniques would likely lead to similar results. Although the specific success rates and final results might differ, these differences would not be substantial, since any alternative control

technique, if functioning as expected, would also aim to drive the state errors to zero, but with different performance characteristics. So, even if the final results are a bit different, the conclusions taken from them in terms of fault influence and types of allocation would be similar.

Firstly, it is confirmed that the LQR is appropriate for controlling the rocket in a fault scenario. Secondly, the analysis shows that some faults have a far greater impact on landing success than others. In the jamming scenario, the success rate remains high because the remaining engines can usually compensate for the control moment error created by the faulty engine, except in a few extreme cases. On the other hand, engine loss scenarios reveal a limit for recoverability. The rocket can recover up to a certain available power level, beyond which reaching the vertical velocity condition for landing is very unlikely, and hence usually leads to mission failure.

Another perspective of the results revealed that some fault characteristics are more involved in the probability of failure than others. The jamming case shows that the jammed TVC angle is, as expected, a critical factor. However, the timing of the fault and the affected engine are also shown to be part of the reason why some runs fail. In the engine power-loss scenarios, the decisive parameter is the available engine power. The particular engine in which the fault occurs is not found to have any meaningful influence, and the timing of the fault only matters when it occurs so late in the mission that it no longer affects the landing conditions. The total engine-loss scenario reinforces the conclusions drawn from the partial loss analysis, but, as expected, the degradation in vehicle behaviour is even more severe. In this case, mission failure is concluded to be unavoidable.

Generally, the study shows that the fault-aware allocation strategy always improves mission success. In the jamming scenario, the improvement is marginal since the baseline success rate is already high in the nominal case. However, in the partial engine loss case, a great increase in the system's fault tolerance is observable. For the total loss scenario, almost all cases fail regardless of the allocation strategy. Therefore, fault awareness offers no meaningful improvement in the total thrust loss scenario.

Future extensions of this work could focus on developing a more robust control strategy or on tuning the LQR in a different way. Furthermore, the LQR, or any control method chosen, could be reconfigured online according to the fault, so that different fault scenarios trigger an appropriate gain-scheduling response.

Another promising direction is integrating the work presented with a fault-tolerant reconfiguration of the guidance system as performed in the work [23]. In the present formulation, the vehicle follows the same nominal trajectory regardless of the fault, but the ability to adapt the trajectory online would improve mission success by aligning both control and guidance with the fault conditions. This would be particularly impactful in engine loss scenarios, where an earlier increase in thrust, rather than the sharp rise during the final ten seconds observed in this case, could provide a better final vertical velocity and hence recoverability probability.

VII. Appendix

A. Linearisation

With the objective of performing the system state space linearisation several partial derivatives had to be computed as it is going to be demonstrated in this appendix (for the sake of simplicity, the partial derivatives of the first state derivatives that result in 0 and 1 are not explicit in this appendix but can be analysed in Equation 16):

- **Vehicle Speed,** V

$$\frac{\partial V}{\partial v_x} = \frac{v_x}{V} \qquad \frac{\partial V}{\partial v_z} = \frac{v_z}{V} \tag{19}$$

- **Dynamic Pressure,** Q

$$\frac{\partial Q}{\partial v_x} = \rho v_x \qquad \frac{\partial Q}{\partial v_z} = \rho v_z \tag{20}$$

- **Angle of Attack,** α

$$\frac{\partial \alpha}{\partial v_x} = -\frac{v_z}{V^2} \qquad \frac{\partial \alpha}{\partial v_z} = \frac{v_x}{V^2} \qquad \frac{\partial \alpha}{\partial \theta} = 1 \tag{21}$$

- **Mach Number**

$$\frac{\partial Mach}{\partial v_x} = \frac{\partial V}{\partial v_x}\frac{1}{c} \qquad \frac{\partial Mach}{\partial v_z} = \frac{\partial V}{\partial v_z}\frac{1}{c} \tag{22}$$

- **Aerodynamic Forces,** F_B^{aero}
  Here, an example of the axial aerodynamic force is presented. The same procedure can be applied to the normal force, or even to the pitching moment, by including the diameter as a multiplying constant.

$$\begin{aligned}
\frac{\partial \mathcal{A}}{\partial v_x} &= S\left(C_{\mathcal{A}}(\alpha, Mach)\frac{\partial Q}{\partial v_x} + QC_{\mathcal{A}_\alpha}(\alpha, Mach)\frac{\partial \alpha}{\partial v_x} + QC_{\mathcal{A}_{Mach}}(\alpha, Mach)\frac{\partial Mach}{\partial v_x}\right) \\
\frac{\partial \mathcal{A}}{\partial v_z} &= S\left(C_{\mathcal{A}}(\alpha, Mach)\frac{\partial Q}{\partial v_z} + QC_{\mathcal{A}_\alpha}(\alpha, Mach)\frac{\partial \alpha}{\partial v_z} + QC_{\mathcal{A}_{Mach}}(\alpha, Mach)\frac{\partial Mach}{\partial v_z}\right) \\
\frac{\partial \mathcal{A}}{\partial \theta} &= QSC_{\mathcal{A}_\alpha}(\alpha, Mach)\frac{\partial \alpha}{\partial \theta}
\end{aligned} \tag{23}$$

- **States First Derivative,** $\dot{\mathrm{x}}$

$$
\begin{aligned}
\frac{\partial \dot{v}_x}{\partial v_x} &= \frac{1}{m}\left(-\frac{\partial \mathcal{A}}{\partial v_x}\cos\theta - \frac{\partial \mathcal{N}}{\partial v_x}\sin\theta\right) \\
\frac{\partial \dot{v}_x}{\partial v_z} &= \frac{1}{m}\left(-\frac{\partial \mathcal{A}}{\partial v_z}\cos\theta - \frac{\partial \mathcal{N}}{\partial v_z}\sin\theta\right) \\
\frac{\partial \dot{v}_x}{\partial \theta} &= \frac{1}{m}\left[-\frac{\partial \mathcal{A}}{\partial \theta}\cos\theta + \mathcal{A}\sin\theta - \frac{\partial \mathcal{N}}{\partial \theta}\sin\theta - \mathcal{N}\cos\theta + T\left(-\sin\theta\cos\beta - \cos\theta\sin\beta\right)\right] \\
\frac{\partial \dot{v}_x}{\partial T} &= \frac{1}{m}\left(\cos\theta\cos\beta - \sin\theta\sin\beta\right) \\
\frac{\partial \dot{v}_x}{\partial \beta} &= \frac{T}{m}\left(-\cos\theta\sin\beta - \sin\theta\cos\beta\right) \\
\frac{\partial \dot{v}_z}{\partial v_x} &= \frac{1}{m}\left(\frac{\partial \mathcal{A}}{\partial v_x}\sin\theta - \frac{\partial \mathcal{N}}{\partial v_x}\cos\theta\right) \\
\frac{\partial \dot{v}_z}{\partial v_z} &= \frac{1}{m}\left(\frac{\partial \mathcal{A}}{\partial v_z}\sin\theta - \frac{\partial \mathcal{N}}{\partial v_z}\cos\theta\right) \\
\frac{\partial \dot{v}_z}{\partial \theta} &= \frac{1}{m}\left[\frac{\partial \mathcal{A}}{\partial \theta}\sin\theta + \mathcal{A}\cos\theta - \frac{\partial \mathcal{N}}{\partial \theta}\cos\theta + \mathcal{N}\sin\theta + T\left(-\cos\theta\cos\beta + \sin\theta\sin\beta\right)\right] \\
\frac{\partial \dot{v}_z}{\partial T} &= \frac{1}{m}\left(-\sin\theta\cos\beta - \cos\theta\sin\beta\right) \\
\frac{\partial \dot{v}_z}{\partial \beta} &= \frac{T}{m}\left(\sin\theta\sin\beta - \cos\theta\cos\beta\right) \\
\frac{\partial \dot{q}}{\partial v_x} &= -\frac{1}{I_{yy}}\frac{\partial \mathcal{M}}{\partial v_x} \\
\frac{\partial \dot{q}}{\partial v_z} &= -\frac{1}{I_{yy}}\frac{\partial \mathcal{M}}{\partial v_z} \\
\frac{\partial \dot{q}}{\partial \theta} &= -\frac{1}{I_{yy}}\frac{\partial \mathcal{M}}{\partial \theta} \\
\frac{\partial \dot{q}}{\partial T} &= \frac{1}{I_{yy}}x^B_{TVC}\sin\beta \\
\frac{\partial \dot{q}}{\partial \beta} &= \frac{1}{I_{yy}}x^B_{TVC}T\cos\beta
\end{aligned}
\tag{24}
$$

B. MC Plots

For improved clarity and structure of the paper, some plots obtained from the Monte Carlo campaigns are presented in this appendix.

MC landing conditions

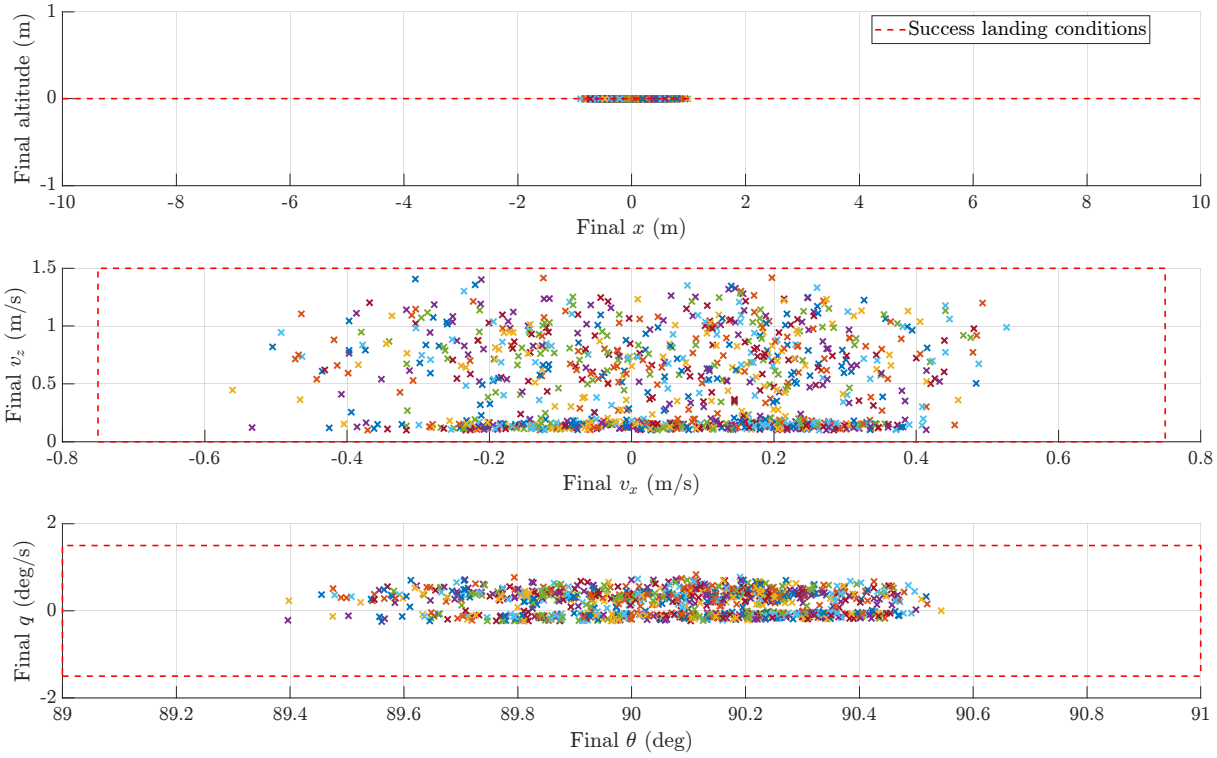


Fig. 16 Fault-free scenario final conditions

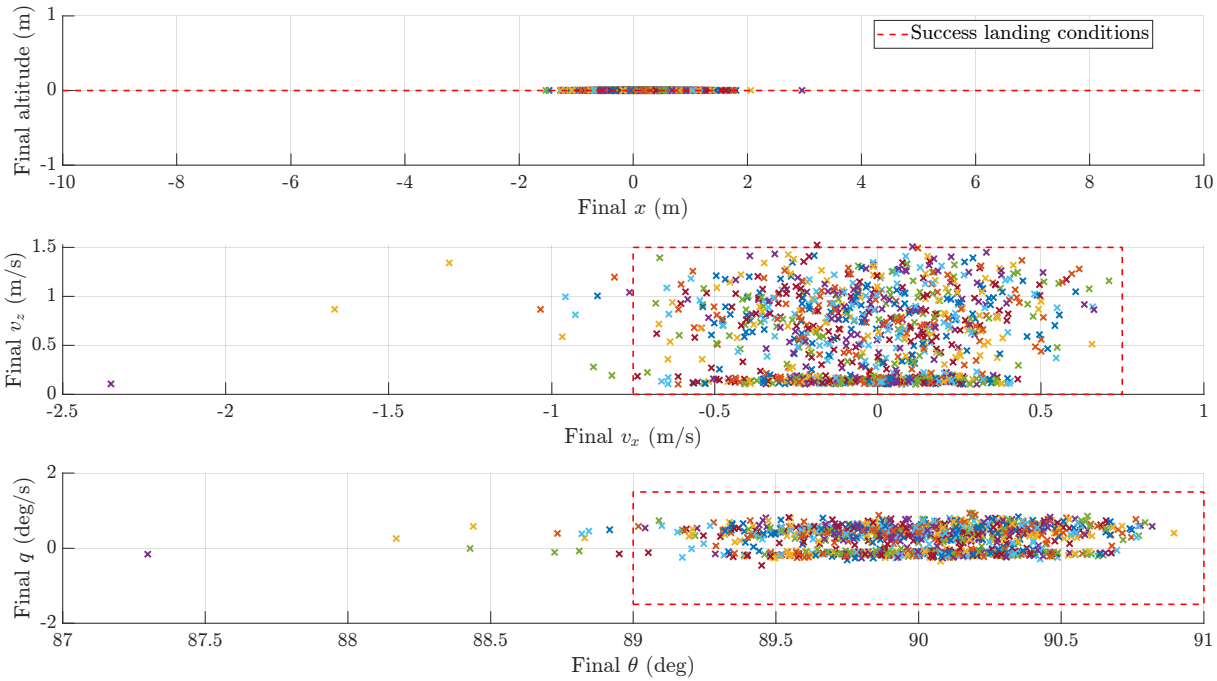


Fig. 17 Non-fault-aware jamming scenario final conditions

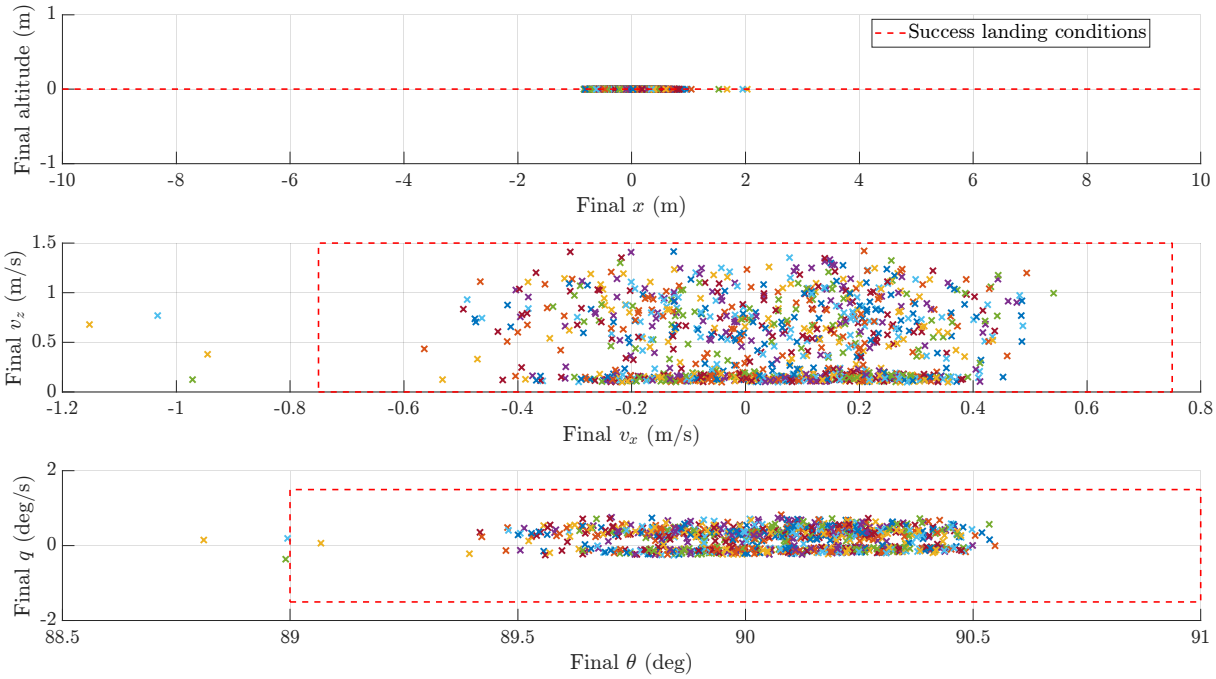


Fig. 18 Fault-aware jamming scenario final conditions

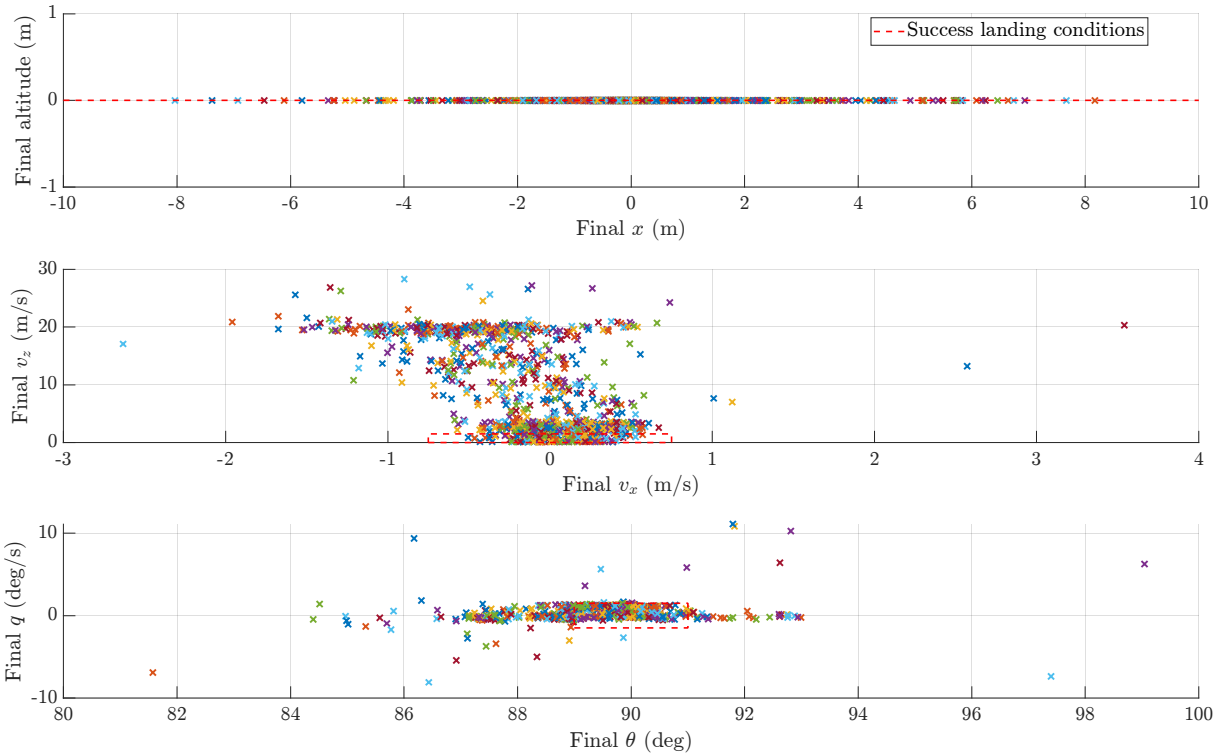


Fig. 19 Non-fault-aware partial engine loss scenario final conditions

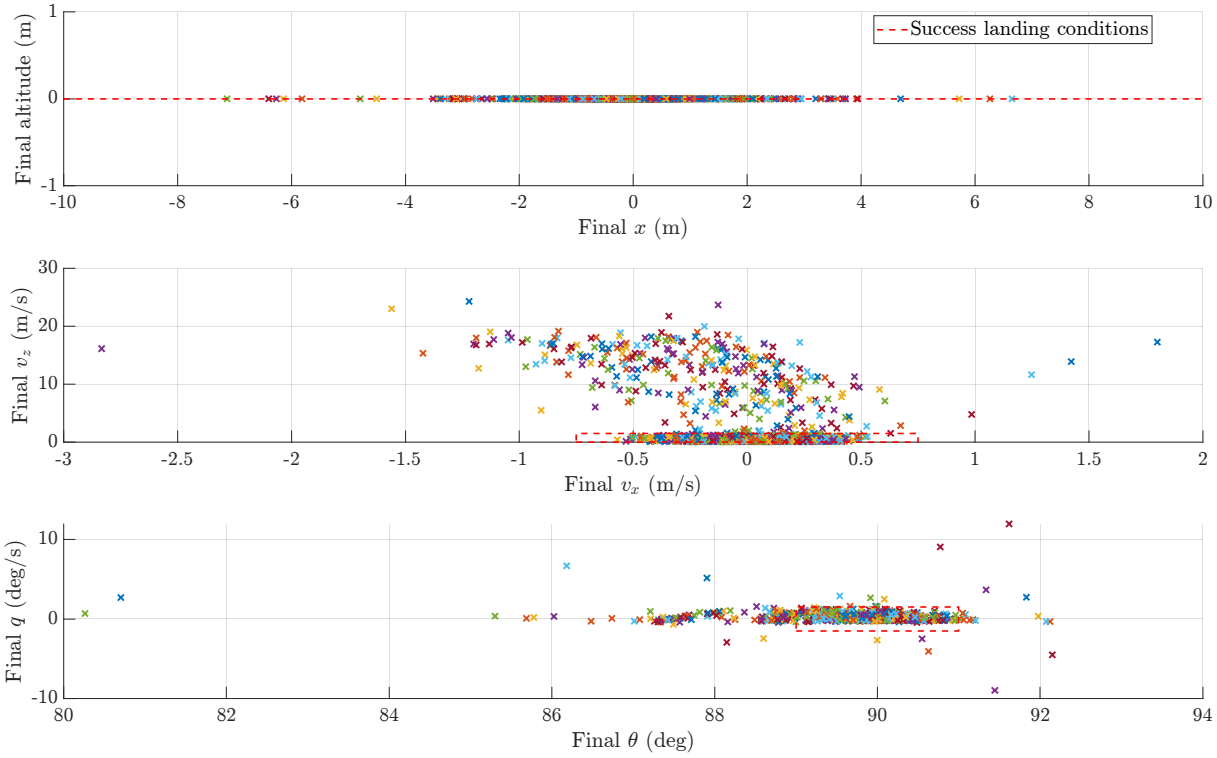


Fig. 20 Fault-aware partial engine loss scenario final conditions

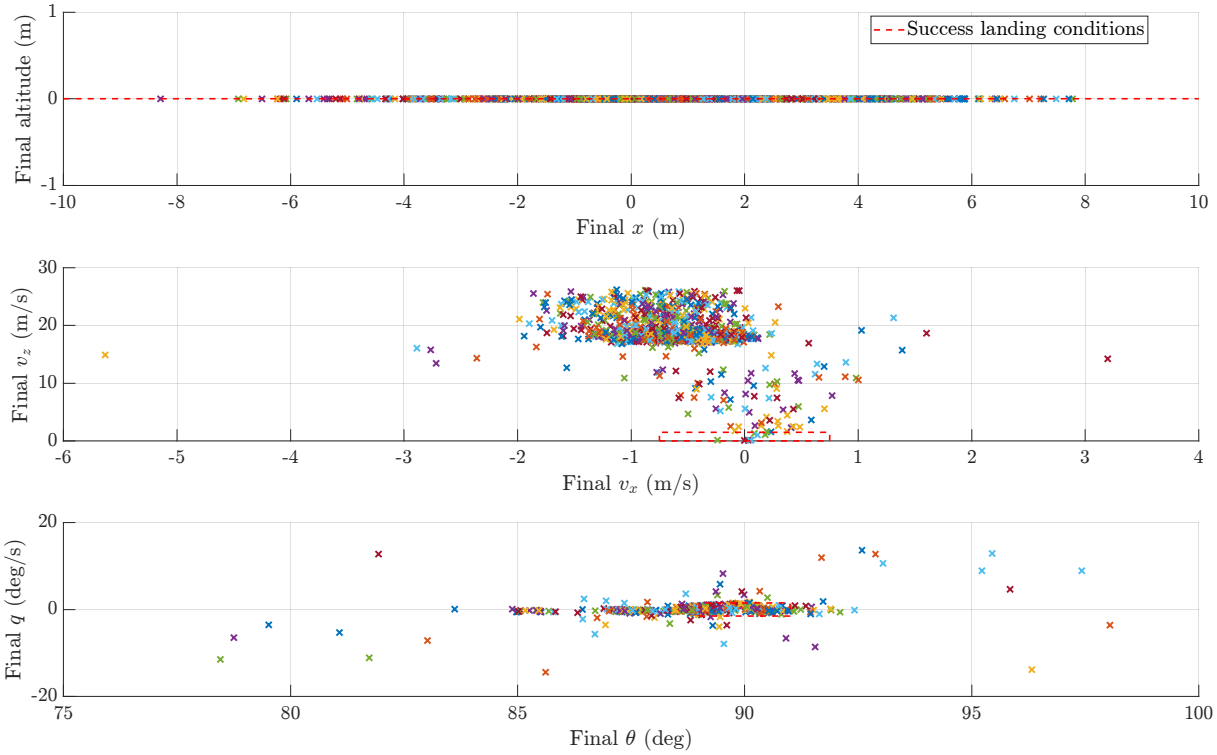


Fig. 21 Fault-aware total engine loss scenario final conditions

MC fault characteristics vs success

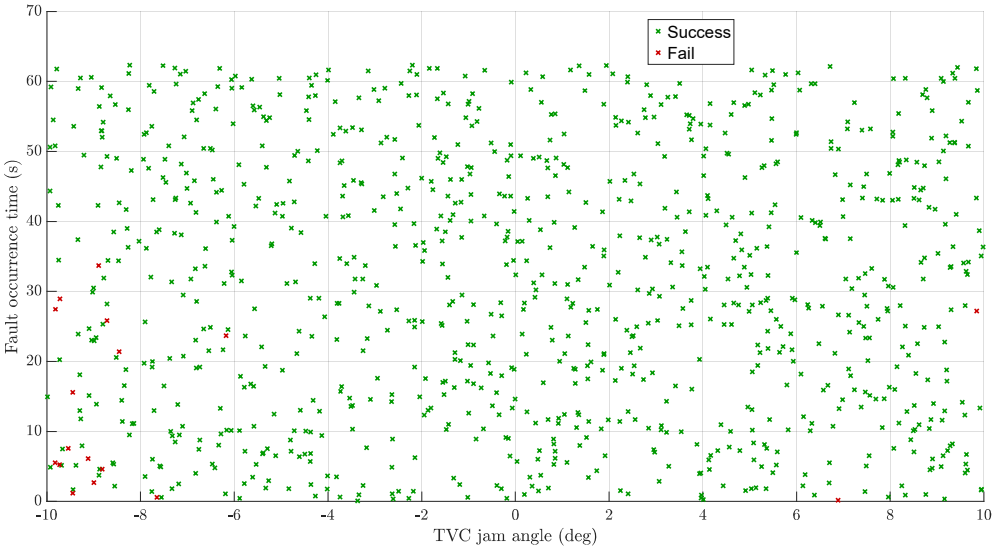


Fig. 22 Jammed angle and time of occurrence influence on success in non-fault-aware jamming fault scenario.

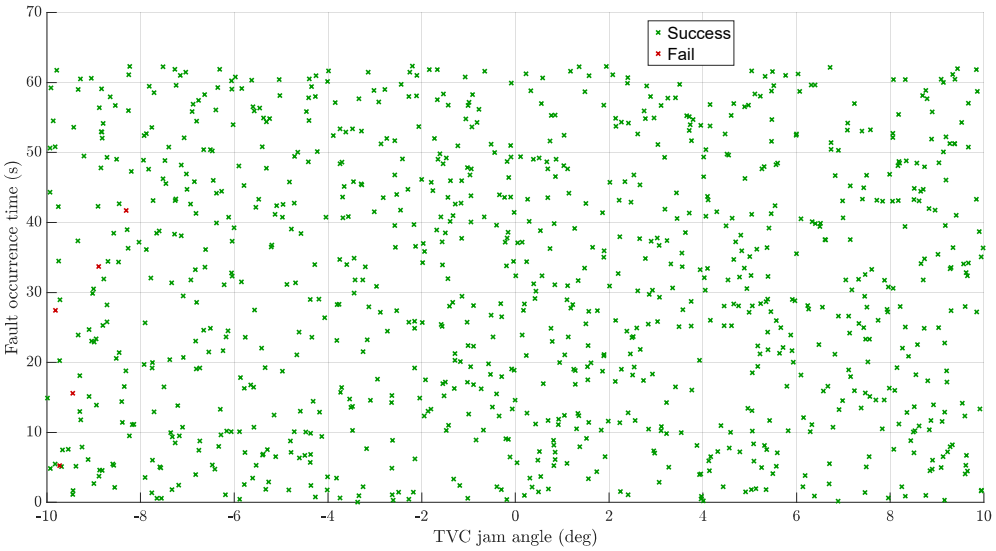


Fig. 23 Jammed angle and time of occurrence influence on success in fault-aware jamming fault scenario.

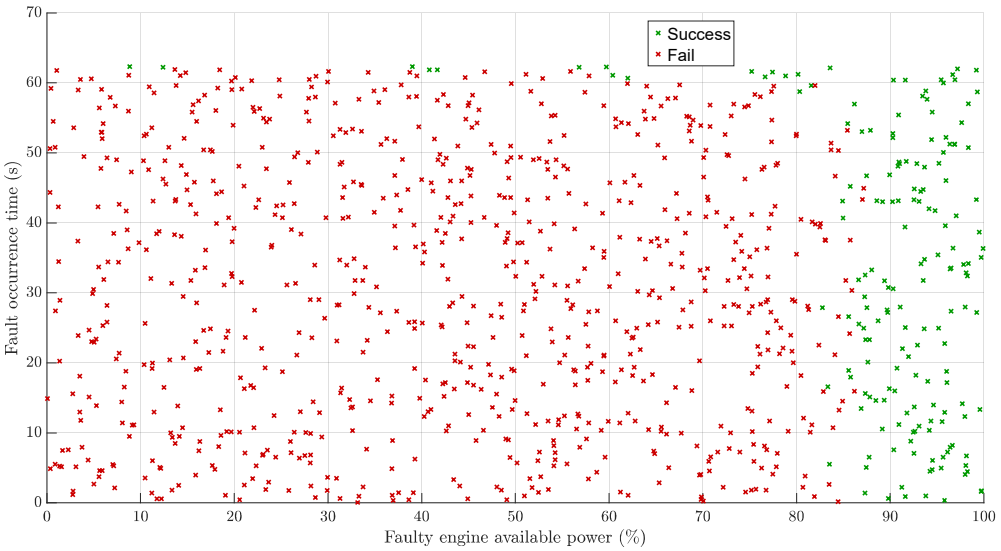


Fig. 24 Engine available power and time of occurrence influence on success in non-fault-aware partial engine loss scenario

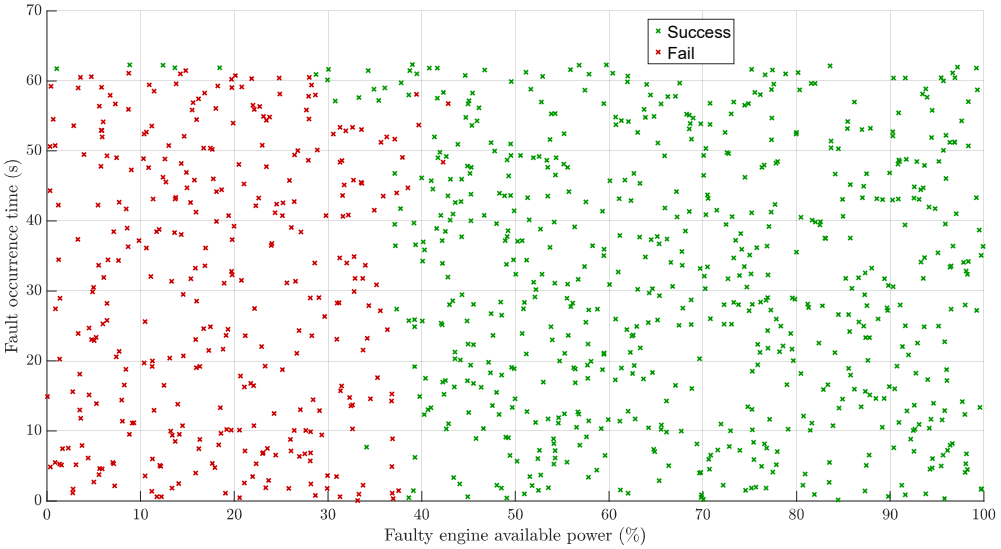


Fig. 25 Engine available power and time of occurrence influence on success in fault-aware partial engine loss scenario

References

[1] SpaceX, *Falcon User's Guide*, Hawthorne, CA, USA, 2025. URL `https://www.spacex.com/assets/media/falcon-users-guide-2025-05-09.pdf`, available online.

[2] Wagner, E., "Research Flights on Blue Origin's New Shepard," *Gravitational and Space Research*, Vol. 9, 2021, pp. 62–67. https://doi.org/10.2478/gsr-2021-0005.

[3] Blue Origin, "New Glenn Mission NG-2," , 2025. URL https://www.blueorigin.com/missions/ng-2, accessed: 2025-12-12; details on the New Glenn orbital launch vehicle's second mission and flight profile.

[4] ILLIG, M., ISHIMOTO, S., and DUMONT, E., "CALLISTO, a demonstrator for reusable launchers," 2022, p. 9 pages. https://doi.org/10.13009/EUCASS2022-7239.

[5] VILA, J., and HASSIN, J., "Technology acceleration process for the THEMIS low cost and reusable prototype," 2019, p. 11 pages. https://doi.org/10.13009/EUCASS2019-97.

[6] Baiocco, P., "Overview of reusable space systems with a look to technology aspects," *Acta Astronautica*, Vol. 189, 2021, p. 10–25. https://doi.org/10.1016/j.actaastro.2021.07.039.

[7] Fernández, L. A., Wiedemann, C., and Braun, V., "Analysis of Space Launch Vehicle Failures and Post-Mission Disposal Statistics," *Aerotecnica Missili Spazio*, Vol. 101, No. 3, 2022, p. 243–256. https://doi.org/10.1007/s42496-022-00118-5.

[8] Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M., *Diagnosis and Fault-Tolerant Control*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016. https://doi.org/10.1007/978-3-662-47943-8, URL https://link.springer.com/10.1007/978-3-662-47943-8.

[9] Zolghadri, A., Henry, D., Cieslak, J., Efimov, D., and Goupil, P., *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles: From Theory to Application*, Advances in Industrial Control, Springer London, London, 2014. https://doi.org/10.1007/978-1-4471-5313-9, URL https://link.springer.com/10.1007/978-1-4471-5313-9.

[10] Paulino, N., Roche Arroyos, C., Ferreira, L., Pascucci, M., Cachim, P., Lourenço, P., García, J., Navarro-Tapia, D., Marcos, A., Mohamed, L., Alexandre, P., Simplicio, P., and Bennani, S., "Fault Tolerant Control for a Cluster of Rocket Engines - Methods and outcomes for guidance and control recovery strategies in launchers," 2023. https://doi.org/10.5270/esa-gnc-icatt-2023-085.

[11] Miramont, P., "Ariane 5 on board software: redundancy management," *Proceedings of the 2nd Embedded Real Time Software Congress (ERTS'04)*, 2004.

[12] Chang-lin, X., Shu-ming, Y., Yu-qiang, C., and Li-jun, S., "Multiple model fault diagnosis and fault tolerant control for the launch vehicle's attitude control system," *The Aeronautical Journal*, Vol. 128, No. 1326, 2024, p. 1875–1894. https://doi.org/10.1017/aer.2024.14.

[13] Roche Arroyos, C., Pascucci, M., Paulino, N., Arnedo, J., Navarro-Tapia, D., Marcos, A., Lalami, M., Alexandre, P., Simplício, P., and Casasco, M., "Fault-Tolerant Control for a Cluster of Rocket Engines - Results for launch and landing of a re-usable launcher," 2022, p. 18 pages. https://doi.org/10.82124/CEAS-GNC-2024-045.

[14] Orr, J. S., and Slegers, N. J., "High-Efficiency Thrust Vector Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, 2014, pp. 374–382. https://doi.org/10.2514/1.61644, URL https://doi.org/10.2514/1.61644.

[15] Navarro-Tapia, D., Simplício, P., and Marcos, A., "Fault-Tolerant Dynamic Allocation Strategies for Launcher Systems," *Aerospace*, Vol. 12, No. 5, 2025, p. 393. https://doi.org/10.3390/aerospace12050393.

[16] Simplicio, P., Marcos, A., and Bennani, S., "A Reusable Launcher Benchmark with Advanced Recovery Guidance," 2019. URL https://eurognc19.polimi.it/, 5th CEAS Conference on Guidance, Navigation & Control, EuroGNC19 ; Conference date: 03-04-2019 Through 05-04-2019.

[17] Sagliano, M., Lu, P., Seelbinder, D., and Theil, S., "Analytical Treatise on Endo-Atmospheric Fuel-Optimal Rocket Landings," *Journal of Guidance, Control, and Dynamics*, Vol. 48, No. 3, 2025, pp. 450–469. https://doi.org/10.2514/1.G008547, URL https://doi.org/10.2514/1.G008547.

[18] Patterson, M. A., and Rao, A. V., "GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, 2014, pp. 1:1–1:37. https://doi.org/10.1145/2558904.

[19] Sagliano, M., Heidecker, A., Farì, S., Jose Alfredo, M. H., Schlotterer, M., Woicke, S., Seelbinder, D., and Dumont, E., "Powered Atmospheric Landing Guidance for Reusable Rockets: the CALLISTO studies," *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2024. https://doi.org/10.2514/6.2024-1761, URL https://arc.aiaa.org/doi/10.2514/6.2024-1761.

[20] Lu, P., and Davami, C., "Rethinking Propellant-Optimal Powered Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 47, No. 10, 2024, p. 2016–2028. https://doi.org/10.2514/1.G008343.

[21] MathWorks, "lqr — Linear-Quadratic Regulator (LQR) design," https://www.mathworks.com/help/control/ref/lti.lqr.html, 2025. MATLAB Documentation. Accessed: 2025-12-19.

[22] Brown, L. D., Cai, T. T., and DasGupta, A., "Interval Estimation for a Binomial Proportion," *Statistical Science*, Vol. 16, No. 2, 2001, pp. 101 – 133. https://doi.org/10.1214/ss/1009213286, URL https://doi.org/10.1214/ss/1009213286.

[23] Soldati, G., "Design of Fault-Tolerant Landing Guidance for Multi-Engine Reusable Launchers," Master's thesis, Technische Universität München, October 2025. URL https://elib.dlr.de/217280/, master's thesis, conducted in collaboration with the German Aerospace Center (DLR).

3

Literature Review

In this chapter, the theoretical foundation upon which this thesis is built is described. For that, literature relevant to the five main topics approached in this thesis is presented, always with the objective in mind of relating it to the study at hand. These topics include:

- Rocket modelling
- Trajectory optimisation
- LQR controller
- Fault-tolerant control
- Control allocation

3.1. Rocket Modelling

In this section, the coordinate systems usually used in rocket problems, as well as the derivations of the equations of motion, are going to be studied with the objective of providing the bases to develop the modelling of an LV.

3.1.1. Coordinate Systems and Kinematics

In order to perform the modelling of a vehicle, it is first necessary to decide which coordinate systems are used. Therefore, the usually used coordinate systems for LV modelling are described [24][25]:

- Inertial Reference Frames - In the literature, it can be seen that an inertial frame is almost always used, usually the Earth-Centred Inertial (ECI) frame.
- Earth-Fixed Reference Frame - Other typically used frame is the non-inertial frame that rotates with the Earth, such as the Earth-Centred Earth-Fixed (ECEF) system. Other local frames include the North-East-Down (NED) system.
- Body-Fixed Reference Frames - This noninertial system is fixed to the moving vehicle, typically at its centre of gravity. Furthermore, the axes are fixed relative to the vehicle structure.

Lastly, the inertia matrix (**I**) is also calculated based on the varying masses and distribution of the propellant in the tanks, typically requiring the parallel axis theorem to convert contributions to the vehicle's body frame coordinates [26].

3.1.2. Rigid Body Equations of Motion and Decoupling for Planar Motion

The fundamental dynamic equations of the launch vehicle are derived assuming it is a rigid body that does not undergo changes in size or shape [27]. The derivation is usually performed from one of these two standard methods:

- Newton/Euler Approach - This method uses Newton's second law to express linear motion and the rotational form of Newton's second law for the angular motion [27].
- Lagrange's Equation - This can also be employed for derivation, being especially useful for systems undergoing vibrations [28].

For this project, the focus is on the first method of derivation, the Newton-Euler approach. Whilst the majority of the references used in this literature review refer to a 6-DoF case, the equations presented are adapted for the 3-DoF case relevant to this work, which includes two translational degrees of freedom (x and z) and one rotational degree of freedom (about the y axis). This 3-DoF model assumes that motion is restricted to the longitudinal plane of symmetry (the x-z plane in the considered body and inertial frames). Therefore, in comparison with the classical 6-DoF case, the following variables are set to zero: y position and velocity, roll and yaw rates, and their derivatives.

Furthermore, regarding the force and moment components: the motion is described by the axial force (along the body x-axis), the normal force (along the body z-axis), and the pitching moment M (about the body y-axis) [24].

Finally, considering the specific 3-DoF case of this thesis, the only non-zero diagonal element of the inertia matrix is I_{yy}. Additionally, since the rocket is considered symmetric about the pitch plane, the non-diagonal terms are also zero [27].

As already mentioned, the motion of the CG is governed by Newton's Second Law. The 6-DoF translational equations can be formulated in the body frame, relating the body accelerations to the total forces acting on the rocket. However, they can also be formulated in the inertial frame:

- Body frame force equations (general case) [27] - these equations are defined by:

$$\sum \mathbf{F}_B = m\frac{d\mathbf{v}_B}{dt} + m(\boldsymbol{\omega} \times \mathbf{v}_B) \tag{3.1}$$

  which if expanded component-wise considering $\mathbf{v}_B = [u, v, w]$ and $\boldsymbol{\omega} = [p, q, r]$ results in:

$$\begin{aligned} \sum F_{x,B} &= m(\dot{u} + wq - vr) \\ \sum F_{y,B} &= m(\dot{v} + ur - wp) \\ \sum F_{z,B} &= m(\dot{w} + vp - uq) \end{aligned} \tag{3.2}$$

- Body frame force equations (planar 3-DoF simplification) - by applying the planar constraints to the x and z body axes ($v = p = r = 0$), results in the simplified translational acceleration components:

$$\begin{aligned} \sum F_{x,B} &= m(\dot{u} + wq) \\ \sum F_{y,B} &= 0 \\ \sum F_{z,B} &= m(\dot{w} - uq) \end{aligned} \tag{3.3}$$

- Inertial frame force equations (general case) - by performing the conversion of the forces from the body frame to the inertial frame, it is possible to obtain the following [25]:

$$\begin{aligned} \sum F_{x,I} &= m\dot{v}_{x,I} \\ \sum F_{y,I} &= m\dot{v}_{y,I} \\ \sum F_{z,I} &= m\dot{v}_{z,I} \end{aligned} \tag{3.4}$$

- Inertial frame force equations (planar 3-DoF simplification) - by applying the planar constraints, $\dot{v}_{y,I} = 0$, the following is obtained:

$$\begin{aligned} \sum F_{x,I} &= m\dot{v}_{x,I} \\ \sum F_{y,I} &= 0 \\ \sum F_{z,I} &= m\dot{v}_{z,I} \end{aligned} \tag{3.5}$$

Regarding the rotational dynamics, these are described by Euler's rotational equations, derived from the rate of change of angular momentum (**H**):

- Rotational pitch equation (general case) [27] - the total external moment ($\sum \mathbf{M}$) about the CG is related to the angular momentum vector ($\mathbf{H}$) and angular velocity ($\boldsymbol{\omega}$) by:

$$\sum \mathbf{M} = \frac{d\mathbf{H}}{dt} + \boldsymbol{\omega} \times \mathbf{H} \tag{3.6}$$

  In the body frame, the Pitch Moment (M) equation (about the y axis) is then:

$$\sum M = \dot{q} I_{yy} + pr(I_{xx} - I_{zz}) + (p^2 - r^2) I_{xz} \tag{3.7}$$

- Rotational pitch equation (planar 3-DOF simplification) - by applying the planar constraint the equation simply transforms to:

$$\sum M = \dot{q} I_{yy} \tag{3.8}$$

3.1.3. Linearisation and Perturbation Methods

The full dynamic equations are nonlinear. To analyse stability and design control systems, the equations can be linearised by assuming small perturbations in the variables about a nominal reference trajectory. This process results in a linear system, which enables the use of several analytical tools [29].

3.1.4. External Load and Environment Modelling

This subsection goes through the main external forces and moments acting on the launch vehicle, which are included in the rigid body equations of motion. When resolved along the body axes, the sum of external forces ($\sum \mathbf{F}$) and moments ($\sum \mathbf{M}$) includes components due to gravity, thrust, and aerodynamics.

Aerodynamics

Aerodynamic models quantify the forces and moments resulting from the vehicle's motion through the atmosphere. Aerodynamic forces are commonly represented using nondimensional force coefficients that can be related to the wind axis system originating the lift, drag and side force or how it is used in this project directly in the body axis system: the axial, side, and normal force components are associated respectively with the coefficients ($C_{\mathcal{A}}$), ($C_{\mathcal{Y}}$) and ($C_{\mathcal{N}}$), each corresponding to a force aligned with one of the body-frame axes (x, y, z) [29]. Similarly, aerodynamic moments are described using the roll, pitch, and yaw moment coefficients (C_l), (C_m), (C_n), which represent the rotational effects about the body-frame axes (x, y, z), respectively [29]. These coefficients are functions of flight parameters, which can be, for example, the Mach number ($Mach = \frac{V}{c}$) and the aerodynamic angles, the angle of attack (α) and the angle of sideslip.

For the planar 3-DOF model adopted in this thesis, only the axial ($\mathcal{A}$) and normal ($\mathcal{N}$) aerodynamic force components remain nonzero, together with the pitch moment ($\mathcal{M}$). In addition, the planar constraint implies zero sideslip angle. Consequently, the aerodynamic forces and moments reduce to the following expressions [30]:

$$\begin{aligned} \mathcal{A} &= QSC_{\mathcal{A}}(Mach, \alpha) \\ \mathcal{M} &= QSdC_m(Mach, \alpha) \\ \mathcal{N} &= QSC_{\mathcal{N}}(Mach, \alpha) \end{aligned} \tag{3.9}$$

with $Q = \frac{1}{2}\rho V^2$ corresponding to the dynamic pressure, S to the reference aerodynamic area, and d to the diameter of the rocket.

Thrust

Another external force and moment that must be modelled is the one related to the rocket thrust. These are computed taking into account the thrust magnitude (T) and TVC gimbal angles (β).

Considering the 6-DoF case, the thrust force in the body-fixed frame is given by [26]:

$$\mathbf{F}_B^{\text{thrust}} = T \begin{bmatrix} \cos\beta_y \cos\beta_z \\ \sin\beta_y \cos\beta_z \\ -\sin\beta_z \end{bmatrix} \tag{3.10}$$

Since in the considered 3-DoF case $\beta_y = 0$, the obtained thrust force is:

$$\mathbf{F}_B^{\text{thrust}} = T \begin{bmatrix} \cos\beta_z \\ 0 \\ -\sin\beta_z \end{bmatrix} \tag{3.11}$$

Regarding the moment created by the TVC system (M_B^{thrust}) this is obtained with the cross product of the vector distance from the CG to the TVC gimbal point ($\mathbf{r}_{GP,B}$) and the previously computed thrust force vector [26]:

$$M_B^{\text{thrust}} = \mathbf{r}_{GP,B} \times \mathbf{F}_B^{\text{thrust}} \tag{3.12}$$

Gravity

Lastly, gravity is another force acting on the vehicle's CG. The gravity acceleration can be computed using several methods as the J4 gravity model, for example. Nevertheless, for the problem at hand for trajectory analysis, since high precision is not required for this study, the gravitational acceleration acting on the vehicle can be modelled by assuming the central body behaves as a point mass. Under this approximation, the gravity field is treated as spherically symmetric, and the magnitude of the gravitational force is computed directly from Newton's law of universal gravitation [27][31]:

$$F^{\text{grav}} = \frac{GMm}{r^2} \tag{3.13}$$

Since the body axes are centred at the vehicle CG, the gravitational moments about the body axis are zero [24].

3.2. Trajectory Optimisation

Since this study is about reusable launch vehicles, one of the steps is to be able to develop an adequate mission for this type of vehicle, specifically focusing on obtaining a powered descent and landing optimal trajectory, so the LV can be safely recovered.

The type of optimal trajectory development is expressed as an Optimal Control Problem (OCP). An OCP typically requires the minimisation of a cost function, for example, propellant consumption (mass) or mission duration, while complying with dynamic constraints and boundary conditions. For rocket landing applications specifically, the objective is frequently expressed in Mayer form to minimise the negative terminal mass, $J = -m(t_f)$, or alternatively in Lagrange form to minimise the integral of thrust magnitude (or throttle if preferred) $\int_{t_0}^{t_f} T(t)dt$ [32]. Given that the continuous representation of the optimal control problem results in a semi-infinite optimisation problem, characterised by infinitely many decision variables and constraints, direct computational methods are not feasible for this problem [33]. The definition of a direct computational method is explained in Section 3.2.1.

3.2.1. Theoretical Foundation

OCP Formulation

The OCP formulation is explained by [34]. The goal is to find the optimal state vector, $\mathbf{x}(t)$, and control vector, $\mathbf{u}(t)$, over a time interval $[t_0, t_f]$ that minimises the chosen cost function. The standard representation for OCPs is the Bolza problem, which combines a terminal cost term for punctual constraints (Mayer term, Φ) and an integrated cost term for variable minimisation or maximisation across the entire throughout the entire flight (Lagrange term, Ψ):

$$\min J = \Phi[t_f, \mathbf{x}(t_f), \mathbf{u}(t_f)] + \int_{t_0}^{t_f} \Psi[\mathbf{x}(t_f), \mathbf{u}(t_f)]dt \tag{3.14}$$

This minimisation is performed subject to several constraints:

- Dynamic Constraints, given by Ordinary Differential Equations (ODEs), which represent the physical constraints of the vehicle's motion: $\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{u})$
- Path Constraints, which include limitations on states and controls, such as physical bounds or structural limits, typically represented as inequalities: $g_L \leq g(t, \mathbf{x}, \mathbf{u}) \leq g_U$

Furthermore, even if not by path constraints, both state variables and control inputs are generally constrained by upper and lower bounds.

Overview of Solution Techniques
To solve the OCP, two main numerical approaches exist: direct methods and indirect methods:

- Indirect Methods - These methods try to satisfy the first-order necessary conditions of optimality derived from optimal control theory [33]. These conditions typically result from applying Pontryagin's Maximum Principle [34].
- Direct Methods - These methods take the continuous OCP and directly discretise it, converting the problem into a large parameter optimisation problem, which is typically a finite-dimensional Nonlinear Programming (NLP) problem [33]. The solution to the OCP is then defined by solving this NLP.

Hamiltonian and Costate Variables
As explained by [35], the Hamiltonian and costate variables are components of the necessary conditions for optimality in optimal control theory.

The Hamiltonian (H) is a function defined to facilitate finding the optimal control trajectory. For the optimal control problem aiming to minimise (or maximise) $\int h(t, \mathbf{x}, \mathbf{u})dt$ subject to the state equations $f(t, \mathbf{x}, \mathbf{u})$, the Hamiltonian is defined as:

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = h(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\top f(\mathbf{x}, \mathbf{u}, t) \tag{3.15}$$

here, $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control vector, and $\boldsymbol{\lambda}$ is the costate vector.

The costate function, $\boldsymbol{\lambda}(t)$, is the multiplier associated with the differential state equations. In continuous time dynamic optimisation, since the state constraint must hold at every moment t, the costate is a continuous function of time. The evolution of the costate is given by the multiplier equation, $\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial H}{\partial \mathbf{x}}$. This value measures the rate at which the optimal value of the objective function would change if the state variable $\mathbf{x}$ were marginally augmented at that time t.

3.2.2. Gauss Pseudospectral Method (GPM)
Direct transcription approaches, which transform the OCP into a usable NLP problem, are a type of tool to use for these types of problems. Amongst these techniques, Pseudospectral Methods (PSMs) have been demonstrated to be suitable in addressing various non-convex optimisation problems, including Powered Descent Guidance (PDG) [34].

One of the PSM's approaches is the GPM, which is a direct transcription technique that depends on approximating state and control trajectories through global polynomials. Furthermore, the dynamic equations are collocated at a particular non-uniform grid named Gauss collocation points. [36].

As explained by [36], the GPM chooses its collocation points only inside the time interval, not placing them at the interval endpoints. This is different from methods such as the Legendre Pseudospectral Method, which include the boundary points. Because GPM uses only interior Gauss points, an important result follows: the optimality conditions obtained after discretising the continuous optimal control problem (the Karush–Kuhn–Tucker (KKT) conditions) match exactly the discretised form of the continuous first-order necessary conditions for optimality. In other words, discretising with interior Gauss points preserves the structure of the continuous optimality conditions, so the discrete solution satisfies the same kinds of necessary conditions as the continuous one. Furthermore, this equivalence allows for the reconstruction of the continuous costates, $\boldsymbol{\lambda}(t)$, directly from the KKT conditions returned by the NLP solver.

General-Purpose Optimal Control
Based on these theoretical fundamentals of GPM, development of computational tools for solving OCPs have been made. Many modern implementations utilise advanced collocation techniques, such as variable-order hp-adaptive Legendre-Gauss-Radau quadrature collocation methods. The use of these methods allows the software to automatically adjust the number and placement of collocation points across the trajectory to achieve a desired accuracy tolerance. The use of Radau points facilitates the efficient enforcement of constraints at the endpoints of each element [37].

Typically, the user defines the problem in terms of continuous and endpoint functions, specifying system dynamics, constraints, and cost objectives. The software then transcribes the OCP into an NLP. Established NLP solvers, such as IPOPT (Interior Point OPTimiser), are then employed to attempt to find an optimal solution [37].

3.2.3. RLV Descent and Landing

PDG Problem
This thesis concentrates on the PDG, defined as the powered descent and landing sequence, occurring at the last several kilometres before surface contact. The principal objective is to enable the landing vehicle to execute a soft landing on a planetary body through the use of its propulsion system, usually with particular attention given to determining trajectories that minimise fuel expenditure [33].

Non-Convexity in RLV Problems
RLV descent and landing OCPs are characterised by several operational constraints necessary for vehicle structural integrity and mission success. As an example, some constraints that might be used are [33] [38]:

- Actuators' performance limits
- Glideslope constraints
- Attitude constraints
- Dynamic pressure limits and aerodynamic load constraints

Since RLV problems often have highly nonlinear dynamics and the aforementioned constraints that are non-convex a lot of the time, they present substantial challenges for numerical solution. Therefore, methods like PSM that transcribe the OCP into an NLP problem need to be used to efficiently handle these non-convexities [33] as it is previously explained in Section 3.2.2.

Bang-Bang Thrust Profile
The structure of these optimal trajectories is, as already mentioned, determined by the objective function, often the minimisation of fuel consumption, and the physical and operational constraints imposed on the vehicle. A known feature of the fuel-optimal trajectory structure is the bang-bang characteristic in the thrust magnitude profile [39].

The bang-bang thrust profile is characterised by the control variable (thrust magnitude) switching instantaneously between its lower (T_{min}) and upper (T_{max}) bounds. This structure is the result of the solution for achieving the theoretical minimum in propellant consumption [40].

The frequency of switching between maximum and minimum thrust levels is determined by the trajectory's complexity. Nevertheless, in the case of vacuum fuel-optimal descent manoeuvres in a constant gravitational field, it has been demonstrated that the optimal thrust profile exhibits no more than two transitions between its boundary values. [32].

3.3. Infinite-Horizon Linear Quadratic Regulator (LQR)

3.3.1. Fundamentals of LQR

For this thesis, an LQR controller is chosen due to being a well-known type of controller used in several contexts, including in LVs, as mentioned in Section 3.3.3. Furthermore, the tuning of this type of controller can be done in an easy to interpret manner, which helps in tuning the controller for several points of the trajectory and a better understanding of the impact of the controller on the behaviour of the rocket. Finally, this controller choice is also based on the fact that the focus of the thesis is on control allocation and that

some physical effects are not included as wind gusts, sensor noise, actuator disturbances, propellant slosh, and structural flexibility. If this were the case, a robust controller would be a more reasonable option.

The main objective of LQR is to compute an optimal state-feedback control law, which minimises a specified quadratic cost function [41].

Performance Index and Weighting Matrices
Based on [42], the infinite-horizon LQR formulation involves defining a cost function J, also known as the quadratic index, to be minimised. For continuous systems, this is typically given by the integral:

$$J = \int_0^\infty \left(\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{u}\right) \tag{3.16}$$

This integral represents the energy to minimise, accounting for both the energy associated with the system's state variables ($\mathbf{x}^\top \mathbf{Q}\mathbf{x}$) and the energy related to the control input ($\mathbf{u}^\top \mathbf{R}\mathbf{u}$).

This method includes two weight matrices, **Q** and **R**:

- **Q** (State Weighting) - This matrix weights the state variables. It is typically selected to be positive semi-definite or positive definite ($\mathbf{Q} \geq 0$). Minimising the state energy component ensures a quick transient response, causing the state variables to converge rapidly towards zero.
- **R** (Control Weighting) - This matrix weights the input energy and must always be positive definite ($\mathbf{R} > 0$).

The relative magnitudes of **Q** and **R** allow for balancing state performance with control effort [41].

Optimal Control Law and the Riccati Equation
As explained in [41], the LQR methodology results in a linear state regulator control law in the form of a linear state feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$, where the optimal feedback gain matrix **K** is obtained using the solution to the Algebraic Riccati Equation (ARE). The ARE must be solved for the unique positive definite matrix **P**:

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} + \mathbf{Q} = 0 \tag{3.17}$$

The optimal control gain $\mathbf{K}_{\text{opt}}$ is then calculated using this solution **P**:

$$\mathbf{K}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} \tag{3.18}$$

The LQR design guarantees that if the system is completely stabilisable and **Q** and **R** are appropriately defined, the result guarantees the asymptotic stability of the closed-loop system. This methodology is introduced in [43], where the foundation for LQR by introducing the Riccati Differential Equation as the computational algorithm to be used is established.

3.3.2. Selection and Tuning of Weighting Matrices (Q and R)

The main objective of selecting **Q** and **R** is to establish the desired trade-off between two goals: minimising the states' deviations from zero and minimising the necessary use of control inputs. For example, if the elements of **R** are chosen to be larger than the elements of **Q**, the resulting optimal control law prioritises minimising the control input energy over minimising the state error. Therefore, selecting these matrices relies on engineering judgment to achieve desired closed-loop stability and time response characteristics [41] [44] [22]. The practical tuning process is typically iterative [41]. Often, this process is started with initial weights and uses techniques like examining trends between weights and resulting gains.

Bryson's Rule for Weight Selection
In [41], a practical guideline for selecting initial numerical values for the diagonal entries of the weighting matrices, named Bryson's Rule, is described. It involves normalising the weights based on the maximum allowable deviation of the corresponding variables.

For the implementation of Bryson's rule, the maximum allowable deviation for each component of the performance output should be defined, z_i, as $z_{i\text{max}}$, and for each component of the control input, u_i, as

$u_{i\text{max}}$. The diagonal weighting coefficients for the states/outputs and for the controls are then selected using the inverse of the square of these maximum deviations:

$$q_i = \frac{1}{z_{i\text{max}}^2} \qquad r_i = \frac{1}{u_{i\text{max}}^2} \tag{3.19}$$

The reasoning behind this choice is that if the allowed limit $z_{i\text{max}}$ for a specific state component is small, the corresponding weight q_i becomes large, thereby having a greater penalty on deviations in that component in the cost function, enforcing smaller deviations in the resulting closed-loop system. This technique is a structured approach to generate suitable initial starting weights for the subsequent iterative tuning required in LQR design [22].

3.3.3. LQR Successful Application in Launch Vehicle Control

The LQR framework has been successfully used in the aerospace industry, including in launch vehicle control, where systems are often nonlinear, time-variant, and unstable [45]. The success of LQR in this context is largely attributed to its capability to manage Multiple-Input, Multiple-Output (MIMO) systems through linearisation and gain scheduling [44].

LQR has been implemented successfully across numerous launch vehicle applications, and some examples are presented here. For instance, it is used in the gain design and scheduling procedure for the Orion Launch Abort Vehicle control system to address the vehicle's instability and aerodynamic characteristics [22]. LQR is also utilised to develop attitude control along the roll and yaw axes for a jet vane thrust-vectored rocket. Given the rocket's highly nonlinear and time-variant dynamics, the design approach involved linearising the system around 429 distinct trim points spanning the 13 second powered ascent phase, enabling the controller to strategically select the optimal gain matrix **K** corresponding to the rocket's instantaneous state [44].

3.4. Fault Tolerant Control

FTC is a study field that can be used for several applications since, generally, systems are susceptible to faults. In the case of an RLV, a fault can lead not only to missing the objective but also to putting the entire system's survival at risk, and so FTC can be useful. To apply FTC solutions, some fundamental concepts are going to be presented in this section.

This is done by following the approaches in [8] and [46]. To start, some definitions can be presented. Firstly, the concept of a fault and distinguishing it from a failure: a failure refers to a complete breakdown of the system, a definitive stopping in its ability to perform a required mission. In contrast, a fault represents a malfunction, not a total collapse. It is defined as a deviation of at least one characteristic property or parameter of the system from its normal condition [46].

System faults can be categorised into several types: actuator, sensor faults and process faults. Actuator faults are malfunctions in the equipment responsible for actuating the system. Sensor faults involve significant variations or inaccuracies in measurement. Process faults occur when changes within the system invalidate its dynamic behaviour, such as a tank leak [46].

The focus of this thesis is on actuator faults. To analyse this situation, a standard control problem of the form $< \mathcal{O}, \mathcal{C}, \mathcal{L} >$ can be defined [8]:

- $\mathcal{O}$ - Objective that the system is expected to achieve.
- $\mathcal{C}$ - Constraints that the system must satisfy over time.
- $\mathcal{L}$ - Control algorithms that can be implemented.

3.4.1. Fault Impacts

As becomes clear in [8], the existence of a fault affects the entire control problem, and so it is important to analyse what it affects specifically.

Impact of faults on constraints

In the presence of a fault, the behaviour of the system may change, which can be represented by introducing a parameter κ that reflects the fault's influence. Consequently, the control problem is redefined as $< \mathcal{O}, \mathcal{C}(\kappa), \mathcal{L} >$.

Two situations may happen: the constraints can be maintained, and only the parameters change $\mathcal{C}(\kappa_n) \rightarrow \mathcal{C}(\kappa_{faulty})$. On the other hand, the actual constraints can change $\mathcal{C}_n(\kappa_n) \rightarrow \mathcal{C}_{faulty}(\kappa_{faulty})$. These 2 cases can be generalised by using $\mathcal{C}_n(\kappa_n) \rightarrow \mathcal{C}_{faulty}(\kappa_{faulty})$ considering that for the first situation $\mathcal{C}_n = \mathcal{C}_{faulty}$ [8].

Impact of faults on objectives
The occurrence of faults should not change the system objectives. If there are means of still achieving the objectives, the system is fault-tolerant, else, it is not. Nevertheless, in case the second situation occurs, the problem can be transformed into a new one by finding new objectives to deal with the current situation [8].

Impact of faults in control
The occurrence of faults may change the set of admissible control laws since faults may occur in the computing and communication devices in which they are implemented [8].

3.4.2. Available Knowledge

Another topic about FTC is the amount of information about the fault that is known. Therefore, the available knowledge of the system is explained based on [8]. The available system knowledge is provided by a Fault Detection and Isolation (FDI) module. Within this module, Fault Detection (FD) identifies that the system is no longer operating under the nominal condition, the control problem has changed from $< \mathcal{O}, \mathcal{C}_n(\kappa_n), \mathcal{L} >$ to a new, faulty configuration. Fault Isolation (FI) then provides information about which parts of the system remain unaffected. More in detail, it identifies the subset of constraints $\mathcal{C}_n(\kappa_n)$ that are still valid (associated with healthy components), and may also identify a reduced set of control laws $\mathcal{L}_{faulty} \subset \mathcal{L}_n$ that can still be applied.

In addition to detection and isolation, further insights about the nature of the fault can be obtained through Fault Estimation, which attempts to quantify the impact of the fault on the system. Fault Estimation may provide information in one of the following three ways:

1. It provides a single estimated model of the faulty system, given by $\hat{\mathcal{C}}_{faulty}(\hat{\kappa}_{faulty})$ for the constraints and $\hat{\mathcal{L}}_{faulty}$ for the usable control actions.
2. It provides a set of possible models, denoted as $\hat{\Gamma}(\hat{\kappa}_{faulty})$, where $\hat{\Gamma}$ represents a set of possible constraints, and $\hat{\kappa}_{faulty}$ is a corresponding set of parameters. This is accompanied by an estimate of the usable control set $\hat{\mathcal{L}}_{faulty}$.
3. It only detects and isolates the fault, but does not provide any estimation of the fault's magnitude or impact on the system. In this case, no explicit information about the modified constraints or control laws is available.

3.4.3. Passive and Active FTC

An FTC solution can be reached in several ways, the solutions are usually divided into two categories: passive and active control strategies [8].

Passive FTC
In the passive approach, the control law remains unchanged when a fault occurs ($\mathcal{L}_n = \mathcal{L}_{faulty}$). This approach treats faults as bounded uncertainties, ensuring stability and acceptable performance without having to reconfigure the controller [8].

One of the adopted fault-tolerant technologies is done by hardware redundancy, which is a passive approach. The Ariane 5 launch vehicle is an example of passive fault tolerance, showing a system designed to tolerate anticipated failures through pre-planned redundancies and fixed recovery mechanisms. In this architecture, features such as independent electrical chains and two identical On-Board Computers guarantee the continuation of operation in the presence of faults [9]. This example of passive FTC shows how these types of solutions can operate, tolerating faults without changing the control strategy, relying instead on robust system design and fallback procedures.

Active FTC
On the other hand, active FTC adapts the control law in response to the detected fault, transforming the problem from $< \mathcal{O}, \mathcal{C}_n(\kappa_n), \mathcal{L}_n >$ to $< \mathcal{O}, \mathcal{C}_{faulty}(\kappa_{faulty}), \mathcal{L}_{faulty} >$. Active FTC can itself be subdivided into two main categories:

- Fault Accommodation - Changing the control law used, but not the plant. Solves the problem $< \mathcal{O}, \hat{\mathcal{C}}_{faulty}(\hat{\kappa}_{faulty}), \hat{\mathcal{L}}_{faulty} >$ or $< \mathcal{O}, \hat{\Gamma}_{faulty}(\hat{\kappa}_{faulty}), \hat{\mathcal{L}}_{faulty} >$ which are the cases 1 and 2 of the available knowledge.
- Reconfiguration - Changing the control law and the plant. If the faulty system is unknown (case 3 of available knowledge), an attempt is made to find a solution for the healthy part of the system, basically switching off the faulty components. Let $\mathcal{C}_{faulty}(\kappa_{faulty}) = \mathcal{C}'_n(\kappa_n) \cup \mathcal{C}''_{faulty}(\kappa_{faulty})$ (healthy + faulty component constraints) and $\mathcal{L}_{faulty} = \mathcal{L}'_n \cup \mathcal{L}''_{faulty}$ following the same logic. Therefore, reconfiguration solves the problem $< \mathcal{O}, \mathcal{C}'_n(\kappa_n), \mathcal{L}'_n >$ by shutting down faulty components and controlling healthy ones.

3.4.4. Thesis FTC Solution

Taking into account the theoretical foundations previously explained, it is possible to relate them to the solution applied in this thesis.

Firstly, in this work, a fully informed system is assumed. This represents an ideal scenario that cannot be achieved in practice, but it most closely approximates the first of the available knowledge cases considered. In this case, the FDI system is capable of both detecting and isolating the fault instantly.

The proposed control strategy is an active FTC approach according to what is described in this theoretical framework. The high-level controller is a fixed LQR law, which remains unchanged in both healthy and faulty conditions. This is, of course, not in accordance with the active FTC approach, since the control law does not reconfigure when faults occur. However, once a fault is detected, the control allocation mechanism can be reconfigured, making the distribution of control effort to be done among the available healthy actuators. This is an active response, since the requests from the controller/allocation output to the actuator set are modified online to only rely on the healthy actuators, compensating for the faulty actuators. In other words, the solution does not redesign or replace the controller itself, but performs reconfiguration through a fault-aware allocation algorithm. This approach only adjusts how the unchanged LQR command is passed to the actuator level, allowing the original controller to operate over a modified, reduced actuator set.

3.5. Control Allocation

The objective of a control allocation is to distribute high-level control commands between several effectors, each with its own limits, to achieve the closest vehicle behaviour according to the controller's request.

3.5.1. Foundational Concepts

Over-Actuated Sytems

The objective of this section is to understand the process of distributing a desired virtual control input among a redundant set of physical actuators in over-actuated systems, which is done by following the work of [12]. Firstly, it is necessary to explain the concept of over-actuation, which occurs when the number of control effectors exceeds the dimension of the control space, resulting in an underdetermined system with infinite feasible solutions. Control effectors examples are thrusters, aerodynamic surfaces, and momentum exchange devices that produce forces and moments that affect vehicle motion, while actuators are electromechanical devices that modulate the magnitude and orientation of these control forces. Over-actuated systems offer redundancy, meaning there are more control inputs (l) than degrees of freedom required by the control system to control (s), resulting in $l > s$. This allows satisfaction of main control objectives and possible secondary optimisation criteria, such as power minimisation or actuator rate limits

General Problem

The design of control algorithms for over-actuated systems is typically organised into a three-level hierarchy [12]. At the highest level, a control algorithm computes a vector of desired virtual control efforts, τ_c. These virtual inputs are commonly selected as forces and moments for vehicle control.

The control allocation algorithm operates at the second level of this hierarchy. Its main function is to compute the physical control inputs $\mathbf{u} \in \mathcal{K} \subset \mathbb{R}^l$ ($\mathcal{K}$ captures the feasible control region bounded by saturation effects and physical limitations) such that their combined effect produces the commanded virtual input τ_c defined in the first level. Since the system is over-actuated ($l > s$), the inverse problem of finding

the control input **u** given command τ_c has non-uniqueness of solutions. Taking this into account, control allocation allows for [12]:

- Modular Design - The high-level motion controller can be designed independently of the specific effectors and actuators present, enabling a clear separation between control objectives and actuator allocation implementation.
- Constraint Handling - The allocation module is responsible for enforcing physical limitations, such as input saturation and rate constraints, that define the admissible set of control inputs $\mathcal{K}$. When the high-level command demands forces or moments beyond the effectors' capabilities due to saturation or other constraints, the control allocation algorithm must degrade performance while minimising the allocation error.
- Fault Tolerance and Redundancy - Actuator and effector fault tolerance can be dealt with the control allocation in the sense that if a fault occurs, the algorithm can reconfigure the system by adjusting constraints or weighting parameters, making sure that the remaining healthy effectors are optimally utilised to maintain the best system response possible.
- Secondary Objectives - Since the solution **u** is generally not unique due to over-actuation, secondary objectives can be incorporated to select an optimal solution from the feasible set. Selected objectives can be minimising power consumption (or fuel usage), reducing actuator wear, and balancing control effort distribution between effectors.

Finally, the third level of the hierarchy includes separate low-level controllers for each individual effector. These level has the objective of transforming the desired individual control inputs (**u**) calculated by the second module into the actual signals required to operate the physical hardware, the actuators. The main objective of this level is to control its actuators in order to achieve the desired force and moment for that specific effector, as commanded by the control allocation.

3.5.2. Classical and Numerical Methods for Linear Models

As previously discussed, the allocation problem exhibits non-unique solutions. This can be better understood by examining the formulation of control allocation methods, which are typically based on a linear effector model of the form, $\boldsymbol{\tau} = \mathbf{Eu}$. In over-actuated systems, where $l > s$, the control effectiveness matrix $\mathbf{E} \in \mathbb{R}^{s \times p}$ is rectangular. This leads to an underdetermined inverse problem, since the number of control inputs exceeds the number of virtual control variables. Consequently, for a given commanded virtual control τ_c, determining the corresponding control input vector **u** results in a system with more unknowns than equations, producing infinitely many feasible solutions [12].

Unconstrained Control Allocation (Generalised Inverses)

A technique for solving this redundancy and selecting a unique solution can involve the use of generalised inverses or pseudo-inverses as explained in [12]. In this approach, constraints are initially neglected, and a quadratic cost function is minimised to select a unique solution from the infinite set. This unconstrained least squares (LS) problem often takes the form:

$$\min_{\mathbf{u} \in \mathbb{R}^l} \ \frac{1}{2}(\mathbf{u} - \mathbf{u}_p)^\top \mathbf{W}(\mathbf{u} - \mathbf{u}_p) \quad \text{subject to} \quad \tau_c = \mathbf{Eu}, \tag{3.20}$$

where $\mathbf{W} \in \mathbb{R}^{l \times l}$ is a positive definite weighting matrix, and $\mathbf{u}_p$ is a preferred value of **u**.

The resulting explicit solution for the control input **u** is given by matrix multiplication involving a generalised inverse **G**:

$$\mathbf{u} = (\mathbf{I} - \mathbf{GE})\mathbf{u}_p + \mathbf{G}\boldsymbol{\tau}_c, \tag{3.21}$$

where

$$\mathbf{G} = \mathbf{W}^{-1}\mathbf{E}^\top(\mathbf{E}\mathbf{W}^{-1}\mathbf{E}^\top)^{-1}. \tag{3.22}$$

The simplest and most widely known implementation is obtained when the weighting matrix $\mathbf{W} = \mathbf{I}$ (the identity matrix) and the preferred control input $\mathbf{u}_p = 0$. In this special case, the generalised inverse $\mathbf{G}$ becomes the Moore–Penrose pseudo-inverse [47] [48], defined as

$$\mathbf{G} = \mathbf{E}^{+} = \mathbf{E}^{\top}(\mathbf{E}\mathbf{E}^{\top})^{-1}, \tag{3.23}$$

and the solution simplifies to

$$\mathbf{u} = \mathbf{E}^{+}\boldsymbol{\tau}_c. \tag{3.24}$$

As stated in [49], pseudo-inverse allocators are highly attractive due to their simplicity and suitability for real-time implementation. Since the computation of the pseudo-inverse involves only matrix multiplication, this explicit algebraic solution is often the conventional choice for control allocation. Nevertheless, despite mechanical designs typically aiming to prevent such situations, the $\mathbf{E}$ matrix may become rank-deficient as a result of singularities or effector/actuator faults. A rank-deficient $\mathbf{E}$ matrix implies that certain forces or moments cannot be generated in specific directions of the $\mathbb{R}^s$ space, meaning that not all commanded virtual controls $\boldsymbol{\tau}_c$ can be achieved, even when actuator saturation limits are ignored. To address rank deficiency, other solutions include SVD-based methods and regularisation or damped least-squares techniques.

Constrained Heuristic and Hybrid Methods

As noted in the previous subsection, algebraic solutions based on generalised inverses do not, by definition, impose physical constraints on the control input $\mathbf{u}$. A simple way to impose constraints is to compute the unconstrained solution $\mathbf{u}$ and then clip it componentwise to the admissible set. The allocated generalised force is generally different from the commanded $\boldsymbol{\tau}_c$. Simple clipping neither guarantees that a feasible $\boldsymbol{\tau}_c$ is achieved when it is attainable, nor that the allocation error $\|\mathbf{E}\bar{\mathbf{u}} - \boldsymbol{\tau}_c\|$ is minimised. In fact, no single choice of generalised inverse together with posterior componentwise saturation can be guaranteed to produce an allocation in accordance with the requirements whenever one exists [12].

To achieve improved performance under actuator constraints, numerous constrained, heuristic, and hybrid allocation schemes have been developed. One of the adopted approaches is the redistributed pseudo-inverse method [50] [51], which operates as follows: initially, control inputs are computed without regard to actuator limits. If all computed inputs fall within their permissible ranges, the solution is accepted. Otherwise, saturated actuators are fixed at their constraint boundaries, and the remaining control effort is redistributed amongst the unsaturated actuators by solving a reduced-dimension allocation problem. This iterative process continues until either a feasible allocation is obtained or no further progress can be made [12].

Even though the redistributed pseudo-inverse method is computationally efficient and frequently effective in practice, it remains fundamentally heuristic. It provides no guarantee of finding a feasible allocation when one exists, nor does it ensure minimisation of the allocation error. Furthermore, suboptimal decisions regarding which actuator components to saturate early in the iterative process can propagate through subsequent steps, ultimately leading to suboptimal final allocations [52].

Another used approach is daisy chaining [12]. In this method, actuators are grouped and assigned priority levels. The allocation is performed sequentially, starting with the highest-priority group. If any actuator in that group reaches its limit, the entire group is fixed at its current values, and any remaining control demand is passed to the next group to allocate.

While this method is simple to implement and an improvement in comparison with the redistributed pseudo-inverse, this sequential strategy can still lead to suboptimal results, as it may not fully utilise all available actuators compared to more globally coordinated methods.

3.5.3. Constrained Optimisation-Based Methods

As seen in the previous sections, one of the challenges of using simple generalised inverse methods in control allocation is that they generally do not guarantee that input constraints are satisfied, or even if they do, the result can be suboptimal. When constraints are explicitly modelled, optimisation-based design becomes possible, especially since the computational complexity of these optimisation methods is already compatible with modern computer technology [12].

In general, the control allocation problem seeks a constrained control input $\mathbf{u} \in \mathcal{K}$ that minimises the weighted allocation error, possibly including an additional cost function. For linear effector models ($\boldsymbol{\tau} = \mathbf{Eu}$) and linear inequality constraints defining a polyhedral feasible set $\mathcal{K}$, the problem can be formulated as either a Linear Programming (LP) or a Quadratic Programming (QP).

Linear Programming
The formulation present in Equation 3.5.2 can be extended to incorporate input constraints. Usually $\mathcal{K}$ is polyhedral which results in:

$$\mathcal{K} = \{\mathbf{u} \in \mathbb{R}^l \quad | \quad \mathbf{Nu} \leq \mathbf{b}\} \tag{3.25}$$

for appropriately chosen constraint matrix **N** and bound vector **b**.

When the cost function involves the 1-norm (L_1) or ∞-norm (L_∞), the control allocation problem is an LP. This problem can be solved with iterative numerical LP algorithms that consider the actuator constraints by using auxiliary variables. This results in a standard LP that can be solved with established algorithms (e.g., simplex, active-set or interior-point methods) [12].

LP solutions lie at the vertices of the feasible set, which tends to produce solutions that use a smaller number of effectors for cost functions defined with the 1-norm. In comparison, using the ∞-norm leads to a different result, since it promotes a more even allocation among the available actuators by using all effectors to a smaller degree. Additionally, practical implementations exploit warm-starts from the previous sample to reduce iteration count [12].

Quadratic Programming
QP is the most common choice for control-allocation objectives based on the 2-norm, since minimising weighted squared errors and actuator deviations typically results in a more smoothly distributed use of actuators. The allocation can be presented as a standard QP that may include slack variables to ensure feasibility. The problem is strictly convex and has a single optimal solution. Unlike the 1-norm LP-based formulations, QP solutions tend to use all effectors to a lesser degree instead of concentrating effort on a few actuators [12].

In practice, QPs are most of the time solved with active-set or interior-point algorithms. Active-set methods are well-suited to control allocation because they support warm starts by using the previous sample's solution, which often reduces iteration counts in real-time implementations. On the other hand, interior-point methods scale better to larger problems but are typically initialised near the interior [12].

3.5.4. Nonlinear Control Allocation

All control allocation methods presented above assume a linear effector model, $\boldsymbol{\tau} = \mathbf{Eu}$. In the majority of real-world scenarios, the effectors behave as nonlinear systems. These linear methods can still be applied by linearising the nonlinearities around an operating point. However, in certain cases, the nonlinear effector model, $\boldsymbol{\tau} = h(\mathbf{u}, \mathbf{x}, t)$, has advantages to be used directly. This is true when linear approximations are inadequate or when the cost function is better described by a norm not achievable in LP, for example [12].

Challenges of Nonlinear Programming (NLP)
The application of NLP to control allocation presents several challenges. When the general optimisation framework incorporates a nonlinear effector model, the resulting allocation problem becomes a nonlinear programme that is frequently non-convex. Consequently, numerical optimisers may converge to a local minimum instead of a global optimum, potentially reaching suboptimal allocations and degraded system performance. Furthermore, solving NLP problems necessitates iterative numerical procedures, which have substantially higher computational costs compared to linear or convex formulations. These computational demands can be particularly problematic in online control applications where timing constraints must be satisfied [12].

Nonlinear Optimisation Algorithms
To solve these NLP problems, some algorithms can be utilised. One used approximation technique is sequential linearisation with quadratic programming, a Sequential Quadratic Programming (SQP) approach as described in [12]. This method iteratively constructs local quadratic approximations of the nonlinear problem, solving a sequence of simpler QP subproblems to progressively approach the solution.

Alternatively, specialised algorithms such as interior-point methods can be used directly on the nonlinear problem. Interior-point algorithms navigate towards the optimal solution by maintaining feasibility while using barrier functions to prevent constraint violations. These methods require smooth objective and constraint functions with accessible first derivatives, and preferably second derivatives for improved convergence. A commonly available implementation is MATLAB's `fmincon` function, which offers interior-point variants specifically designed for constrained nonlinear minimisation [53].

3.5.5. Allocation Methods for RLV Propulsion Systems

Historically, control allocation methods based on generalised inverses have been widely used to distribute virtual control commands amongst redundant effectors on aircraft and spacecraft. Even though attractive for their computational simplicity, generalised inverse solutions frequently fail to fully exploit the available control authority, as discussed earlier in this section. To address this limitation, the allocation problem can be converted to constrained optimisation problems and solved using LP or QP solvers, resulting in a more effective actuator utilisation [54]. Alternatively, nonlinear programming methods can be applied to achieve the control allocation objectives. Both approaches, along with their possible specific implementations methodologies, are examined in the following subsection.

For the propulsion case, the allocation problem involves coordinating the individual engine parameters, thrust magnitudes(T_i) and gimbal angles (β_i) to produce the overall commanded force and torque references τ_c. The relationship between these inputs and the resulting forces and moments is nonlinear, since the used equations contain trigonometric functions of the gimbal angles. Even when simplifying assumptions, such as small-angle approximations for TVC deflection are applied, the control effectiveness matrix remains nonlinear due to the coupling between thrust magnitude and gimbal angle [53].

Furthermore, this nonlinearity frequently results in optimisation problems characterised by non-convex objective functions and/or constraint sets [12]. Since non-convex problems may have multiple local minima, numerical solvers have the risk of converging to suboptimal solutions, potentially degrading control performance. Given these challenges, two main optimisation strategies are used in this thesis: constrained nonlinear programming to address the non-convex problem directly, and convex optimisation based on a reformulation of the allocation problem characteristics to ensure convexity. Both strategies have been successfully demonstrated in similar other aerospace control allocation problems, as evidenced in [20] and [53].

Convex Optimisation (Reformulated Approach)

The physics of an RLV allocation problem are non-convex. However, the use of convex optimisation offers several advantages, as already explained in this literature review (e.g. guaranteed global optimality and computational reliability). To exploit these advantages, the allocation problem can be reformulated as a convex problem.

For the purpose of implementation, specific software is available that offers a framework for modelling the formulation and solution of the Disciplined Convex Programmes (DCPs) in the MATLAB environment for example [55].

Constrained Nonlinear Optimisation (Non-Convex Approach)

The direct approach to solving the RLV allocation problem involves formulating it as a nonlinear programming problem that accommodates the nonlinear and non-convex physics of the propulsion system. This formulation makes no simplifying assumptions about convexity.

The MATLAB function `fmincon` (find minimum of constrained nonlinear multivariable function) is a tool for solving constrained NLP problems. By default, `fmincon` employs the interior-point algorithm. Other algorithms, such as Sequential Quadratic Programming, are also offered and may have certain benefits depending on the problem [56].

Part II

Additional Results

4

Additional Results

4.1. Individual Run Analysis

In the paper, the global results of the MC campaign are presented, which show the overall behaviour and success rates of the system across the range of fault scenarios, initial conditions and parameter variations. Those global statistics are essential, but they do not by themselves reveal the behaviour of the system that produces success or failure in a single trial. To gain that insight, individual runs can be inspected to better understand the causes that lead to the final outcomes.

Therefore, for each batch of MC runs, a single illustrative trial is selected for closer analysis. This will allow comparisons between the fault-aware and fault-unaware allocation strategies, showing how and why one approach outperforms the other in different fault circumstances, and better understand the effects of the faults.

For each batch of Monte Carlo runs, the selected case uses the same fixed initial conditions and variable parameters. Additionally, the time of fault occurrence and the position of the faulty engine are kept the same across all scenarios, while the nature of the fault itself changes as required for each case. This consistent setup ensures that any differences observed in the vehicle response can be attributed directly to the type of fault and to the allocation strategy, rather than to variations in initial conditions. All the corresponding values are presented in Table 4.1.

Table 4.1: Initial conditions, parameter and fault characteristics of the analysed run

Property	Offsets and Injected Fault	Units
$\Delta r_{x,z,RP,0}$	$[86.57, -75.22]$	m
$\Delta v_{x,z,RP,0}$	$[-1.40, 0.69]$	m/s
$\Delta\theta_0$	3.49	deg
Δq_0	1.56	deg/s
Δm_{dry}	11.70	kg
$\Delta I_{yy,\text{dry}}$	-760.00	kg m^2
$\Delta C_{\mathcal{A},m,\mathcal{N}}$	$[-8.67\%, 2.85\%, 6.50\%]$	-
$\Delta\rho$	-4.62%	kg/m^3
T_{avail}	$8475.50\,(51.95\%)$	N
β_{jam}	0.39	deg
$t_{\text{fault occurence}}$	31.23	s
$\text{Engine}_{\text{fault}}$	4	-

4.1.1. Fault-Free Scenario

Firstly, it is necessary to understand how the system behaves in the fault-free scenario, as this represents the nominal operating condition and provides a benchmark against which all faulty cases can be compared.

For this purpose, the error evolution is shown in Figure 4.1, while the corresponding actuator behaviour is presented in Figure 4.2.

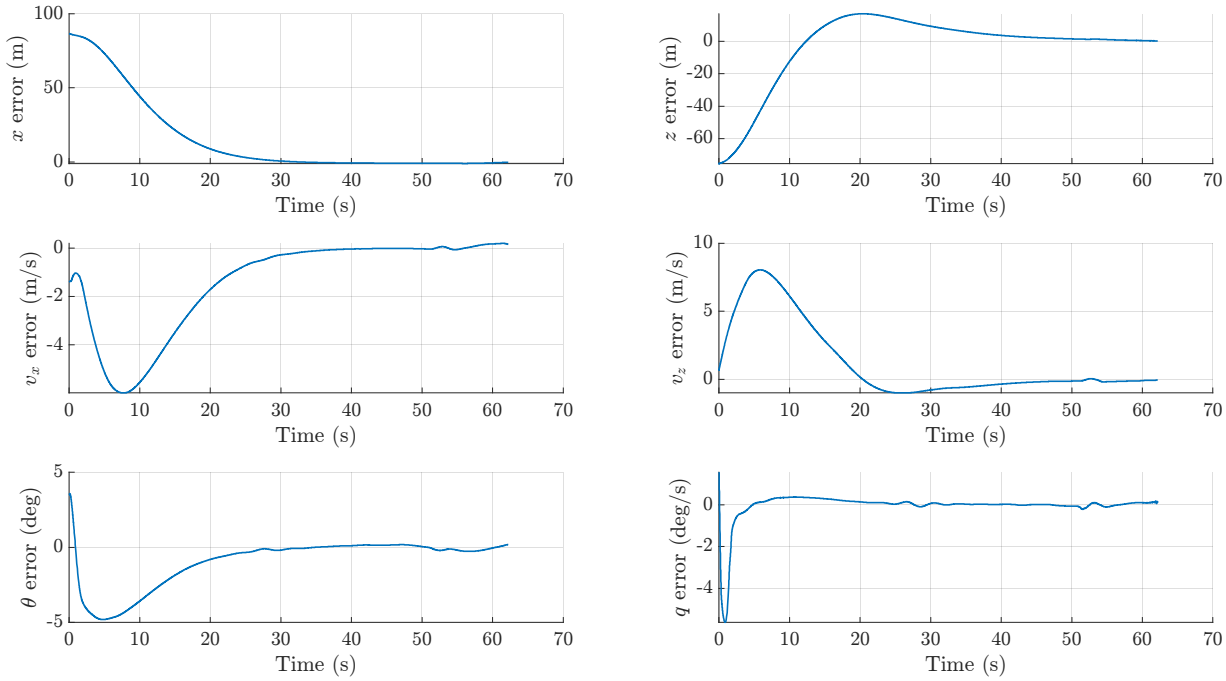


Figure 4.1: Absolute errors time evolution for individual run in fault-free scenario

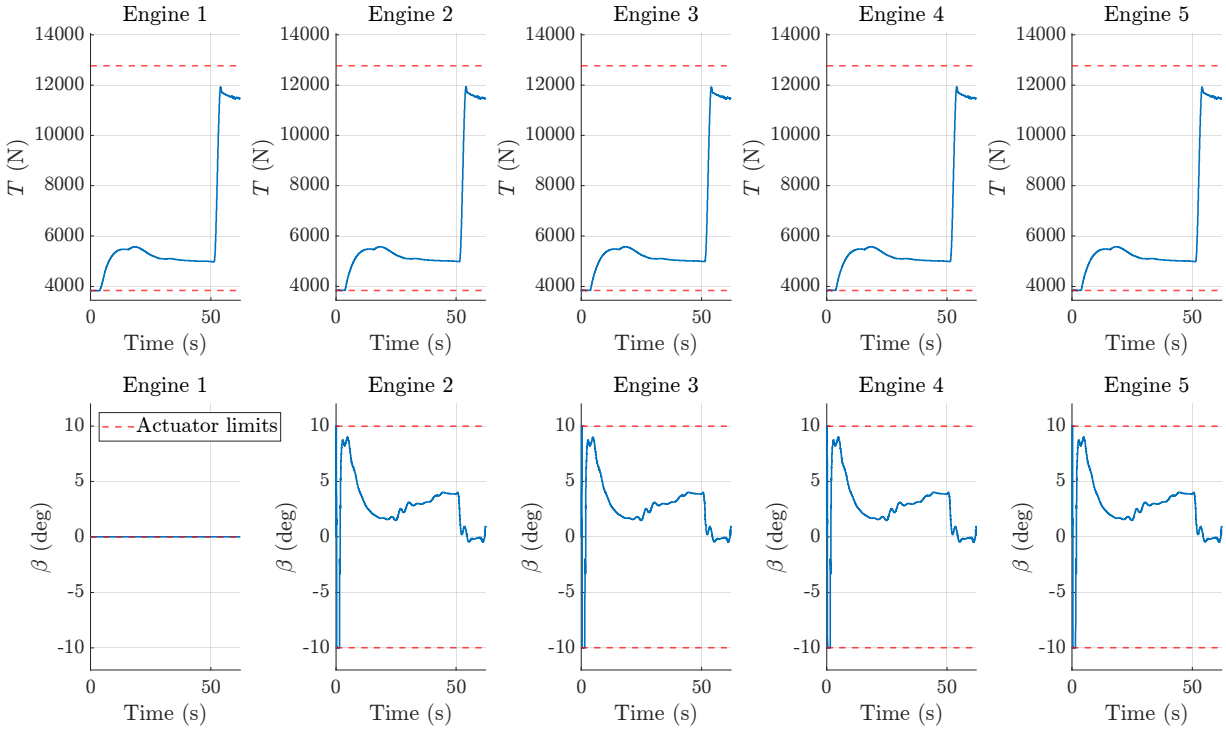


Figure 4.2: Actuator action time evolution for individual run in fault-free scenario

From the error plots, it can be seen that all errors are driven to zero as expected, although at different rates. In particular, the attitude components converge more rapidly, which is a direct consequence of the system characteristics and the controller design, with the higher priority assigned to attitude regulation

at the beginning of the trajectory. In contrast, the velocity and position errors converge more slowly, but nevertheless reach values sufficiently close to zero after approximately 50 s. This behaviour is consistent with the intended tuning of the control law and confirms correct nominal performance.

4.1.2. Jamming Scenario

In the jamming fault scenario, both the fault-unaware and the fault-aware allocation strategies exhibit very similar behaviour and are able to reduce the errors to acceptable levels, as shown in Figure 4.3, leading to a soft landing and guaranteeing mission success in both cases. The only noticeable difference is the presence of slightly increased oscillations in the attitude components for the fault-unaware case following the occurrence of the fault. The timing of these oscillations is not random: they occur both at the instant of fault occurrence (around 31 s) and during the transition to a high thrust demand (around 51 s).

This behaviour can be explained by the different ways in which the two allocation strategies react to the fault. In the fault-unaware case, the controller responds only once the attitude errors become large enough to start a more meaningful corrective action. By contrast, the fault-aware allocation accounts for the fault as soon as it occurs and compensates for it directly at the allocation level, correcting the disturbance at its source. This explains why oscillations appear at these two specific instants: both correspond to sudden changes in the conditions either imposed by the fault or requested from the system, which temporarily increase the tracking error. Nevertheless, the resulting oscillations in the fault-unaware case remain well within recoverable limits and do not compromise the mission success. When compared with the fault-free scenario, the overall error evolution remains similar, except for this additional oscillatory behaviour.

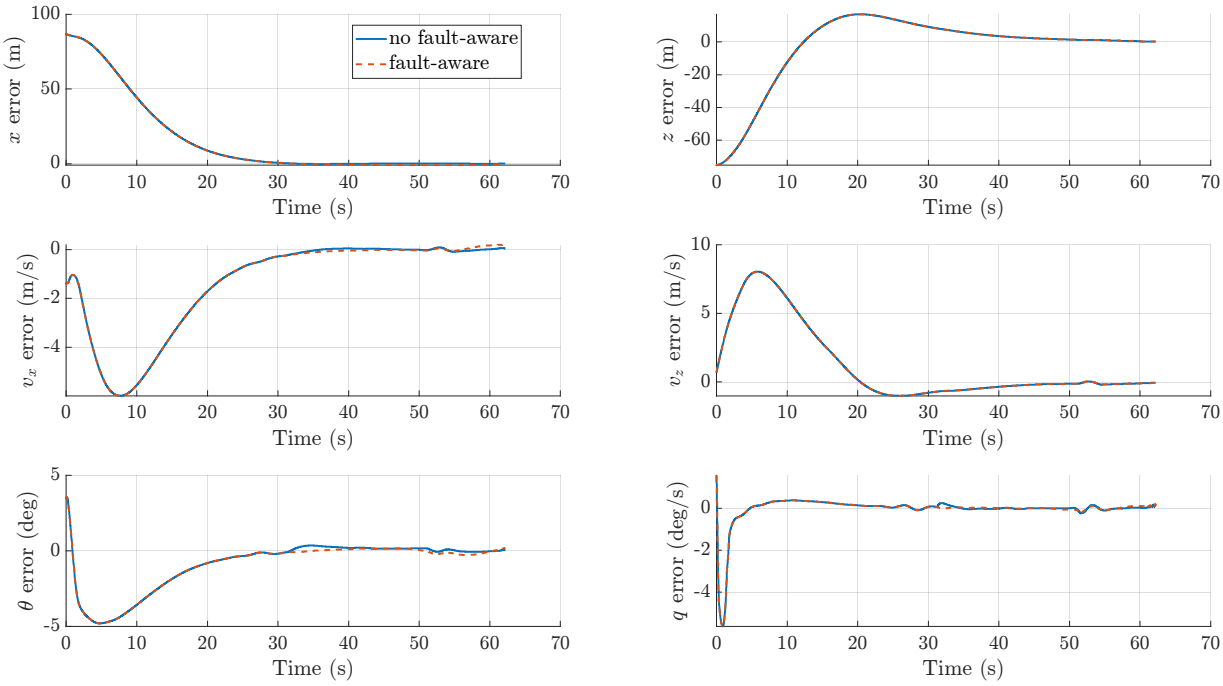


Figure 4.3: Absolute errors time evolution for individual run in jamming scenario

In terms of actuator activity, the responses of both allocation strategies are again very similar. A small adjustment in the gimbal angle is observed to compensate for the jammed engine, but no significant differences arise either between the two allocation methods or with respect to the nominal case. For this reason, the corresponding actuator plots are not presented, as they would not provide additional insight.

4.1.3. Partial Engine Loss Scenario

In the case of the partial engine loss scenario, it can be observed that the errors, in Figure 4.4, are once again driven towards zero during the initial phase, in a way similar to the fault-free case, even after the fault occurrence, this continues to happen. However, a clear difference between the fault-unaware and fault-aware strategies appears when the guidance commands maximum thrust at around 51 s. At this point,

the fault-unaware case struggles to maintain the error close to zero, while the fault-aware case exhibits only a small error peak that is rapidly corrected.

This behaviour occurs because, following this guidance request, the launch vehicle is no longer able to deliver the level of thrust demanded due to the presence of the fault if the gimbal angles of the rocket continue with their nominal behaviour, even with the remaining healthy engines operating at their maximum capability. Immediately after the increase in thrust demand, the vertical velocity error begins to grow. In the fault-unaware allocation case, the controller attempts to compensate for this velocity deviation but, in doing so, deprioritises the angular states, particularly the pitch rate. As a result, the control authority is ”distracted” away from correcting the vertical velocity error effectively. Although the velocity error is partially reduced, it does not return sufficiently close to zero.

Consequently, the mission in the fault-unaware case terminates earlier than in the fault-aware case, as the launch vehicle reaches the ground with an excessive vertical velocity. In this scenario, the final vertical speed is $v_{z,f} = 8.85$ m/s, which violates the soft-landing criteria and therefore leads to a non-successful mission. By contrast, the fault-aware allocation explicitly accounts for the thrust loss and adjusts the gimbal angles accordingly to use the remaining control authority more effectively without ever putting the attitude at risk, and so the controller never loses focus on the vertical states. As a result, the vehicle maintains better vertical thrust alignment, achieves a later touchdown, and completes a successful landing.

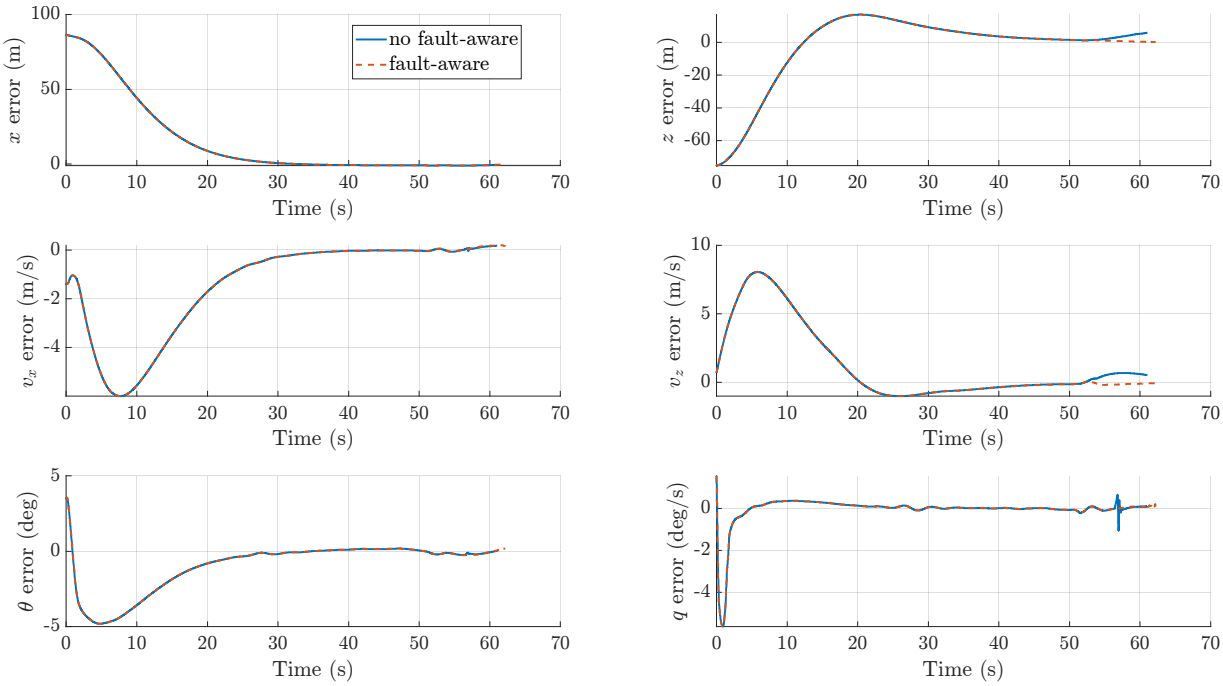


Figure 4.4: Absolute errors time evolution for individual run in partial engine loss scenario

These conclusions are also supported by the actuator responses. Following the increase in thrust demand, the thrust of all engines rapidly reaches their maximum allowable values in both allocation cases. At the same time, in the fault-unaware allocation case, all gimbal-capable actuators exhibit big peaks in their gimbal angles as they attempt to compensate for the growing attitude errors, later than in the fault-aware case. The responses of all the TVC angles of the individual gimballed actuators are practically equal in both cases, with an example shown in Figure 4.5.

This behaviour further degrades the ability to recover the vertical velocity: since the actuators operate at non close to zero gimbal angles, a larger fraction of the available thrust is used into lateral components rather than being aligned with the vertical axis. As a consequence, the vertical thrust component is reduced, not allowing the vertical velocity error to decrease and therefore contributing to the unsuccessful landing outcome in the fault-unaware case.

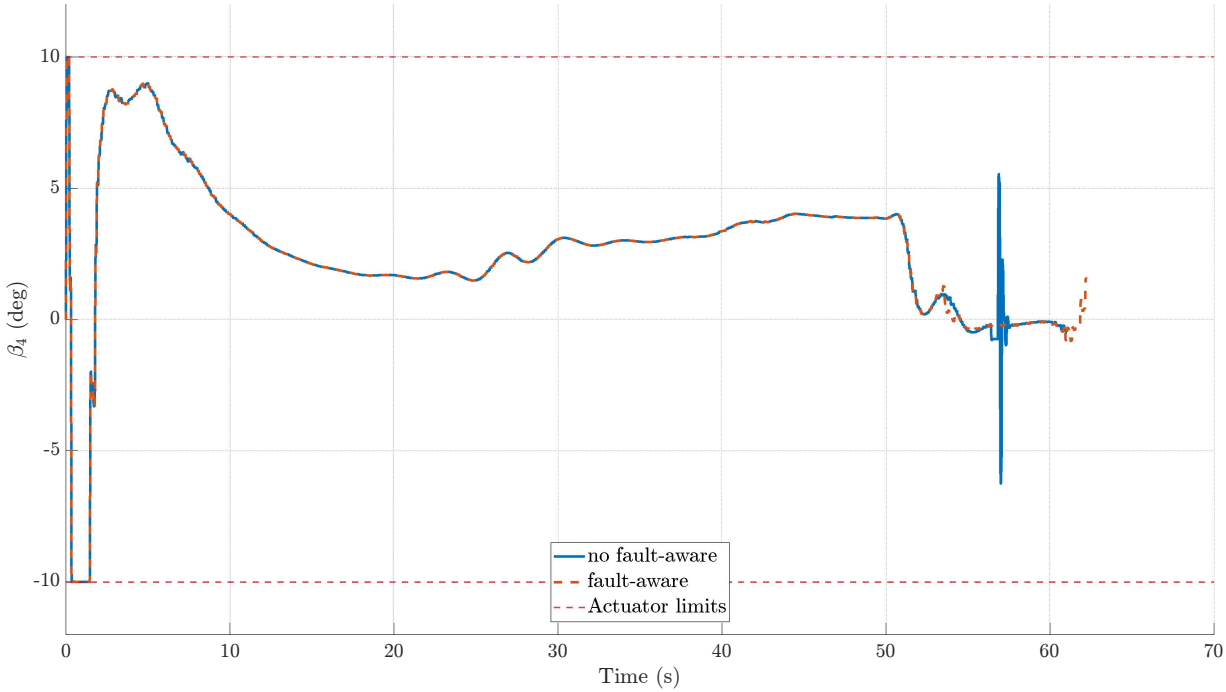


Figure 4.5: Gimbal angle time evolution for engine 4 in partial engine loss scenario

4.1.4. Total Engine Loss Scenario

Finally, in the total engine loss scenario, the behaviour can be interpreted as an amplified version of the partial engine loss case. The errors remain close to zero until the request for high thrust. After this point, however, both allocation strategies lead to mission failure. Nevertheless, the fault-aware allocation exhibits slightly smaller errors, as shown in Figure 4.6, indicating a slightly improved response even though successful recovery is no longer physically achievable.

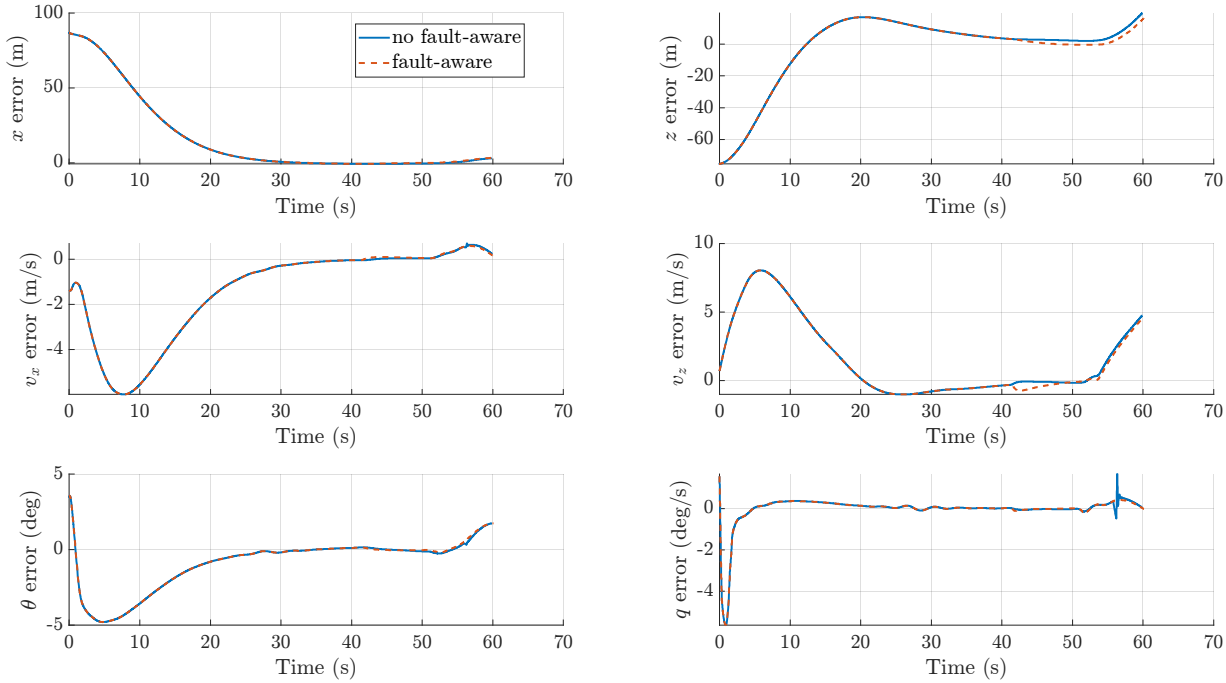


Figure 4.6: Absolute errors time evolution for individual run in total engine loss scenario

All actuators are driven to their maximum thrust, while the gimbal angle behaviour is similar to that observed in the partial engine loss case. However, in this scenario, the gimbal angles reach higher peak values, which shows a more aggressive attempt to recover the vehicle attitude, which is related to the fact that this attitude is more affected in this case than in the partial engine loss case.

4.2. Faulty Engine Influence

In addition to the MC campaign already presented in the paper, additional batches of simulations are performed to investigate and confirm the influence of faults occurring in specific engines in the fault-aware scenario. For this purpose, additional batches of one thousand runs are executed in which faults are applied exclusively to a single engine at a time. Since Engines 2 and 4 produce equivalent effects, as they are both located out of the mission plane and are in the same position from this problem's perspective, they are grouped together and analysed within the same batch of runs.

For the jamming fault cases, only the remaining gimbal-capable engines, namely engines 3 and 5, are considered. For the partial engine loss scenario, Engine 1 is also included in the analysis. This led to the definition of the following batches of simulation runs:

- MC 8 - TVC jamming scenario, with a fault-aware control allocation for out-of-plane engines 2 and 4.
- MC 9 - TVC jamming scenario, with a fault-aware control allocation for engine 3.
- MC 10 - TVC jamming scenario, with a fault-aware control allocation for engine 5.
- MC 11 - Partial engine thrust-loss scenario, with a fault-aware control allocation for engine 1.
- MC 12 - Partial engine thrust-loss scenario, with a fault-aware control allocation for out-of-plane engines 2 and 4.
- MC 13 - Partial engine thrust-loss scenario, with a fault-aware control allocation for engine 3.
- MC 14 - Partial engine thrust-loss scenario, with a fault-aware control allocation for engine 5.

4.2.1. Jamming Scenario

For the jamming scenario, the global results obtained from these additional batches of runs are presented in Figure 4.7. The first bar in the chart corresponds to the overall results already presented in the paper and serves as a reference for comparison with the new, engine-specific fault cases.

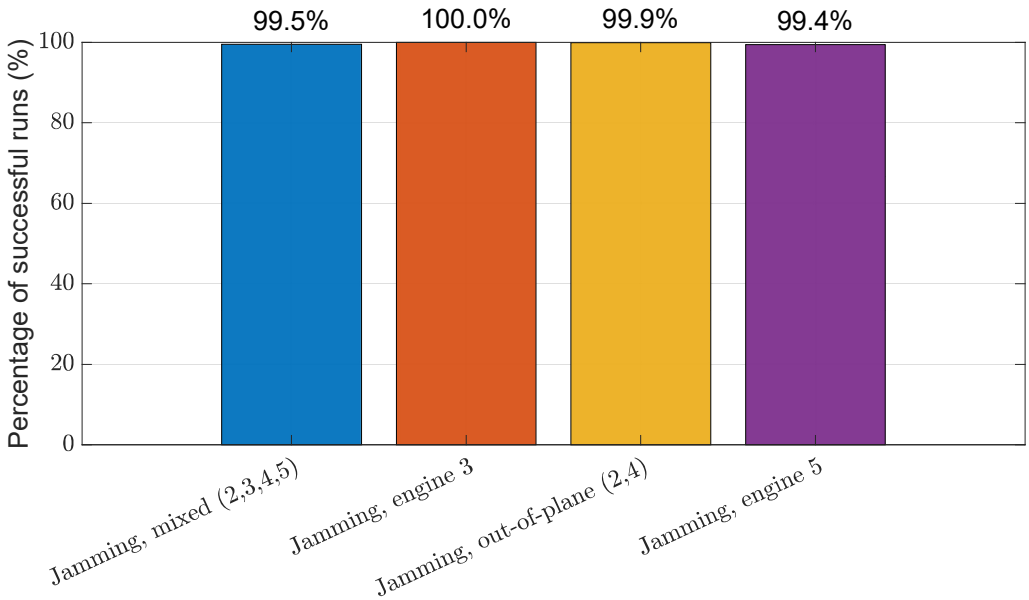


Figure 4.7: Monte Carlo campaign jamming case comparison

Although the differences between the results are very small, they nevertheless confirm the conclusions already drawn in the paper. As previously discussed for the fault-aware case, engine 5 is the one for which

the highest number of failures occurs, while only rare failures are observed for the out-of-plane engines, and no failures at all are recorded for engine 3 in the fault-aware allocation case (in the paper results, they only occurred in the non-fault-aware case). The high rates of success reinforce that the only reason why the run might fail with a jamming fault is in very specific extreme conditions, once again supporting the conclusions of the paper.

4.2.2. Partial Engine Loss Scenario

The results for the partial engine loss scenario with faults applied to individual engines are summarised in the bar chart shown in Figure 4.8.

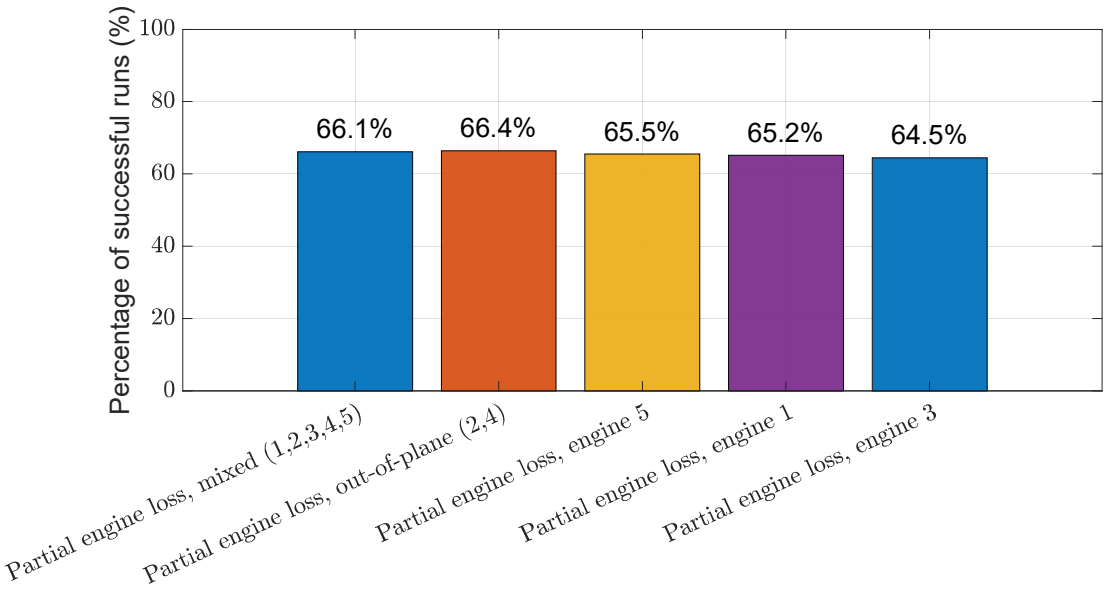


Figure 4.8: Monte Carlo campaign partial engine loss case comparison

From the analysis of these results, it can be concluded that the overall success rates are very similar, as is also observed in the paper. However, in this case, the ranking of success rates across different engines differs from that presented in the paper for both allocation strategies. This suggests that the specific engine in which a fault occurs has little or even no influence on the outcome and that the observed variations are primarily due to the stochastic nature of the Monte Carlo campaign, therefore confirming the conclusions reached in the paper.

5

Verification and Validation

In this chapter, the verification and validation of the proposed methodology is presented. The aim is to ensure that the results obtained are reliable and can be confidently used to support the conclusions drawn.

5.1. Trajectory

The first step of the problem at hand is to derive an optimal trajectory. In order to guarantee it is an adequate trajectory for the problem, it will be confirmed that the trajectory is indeed optimal and that the trajectory can actually be followed.

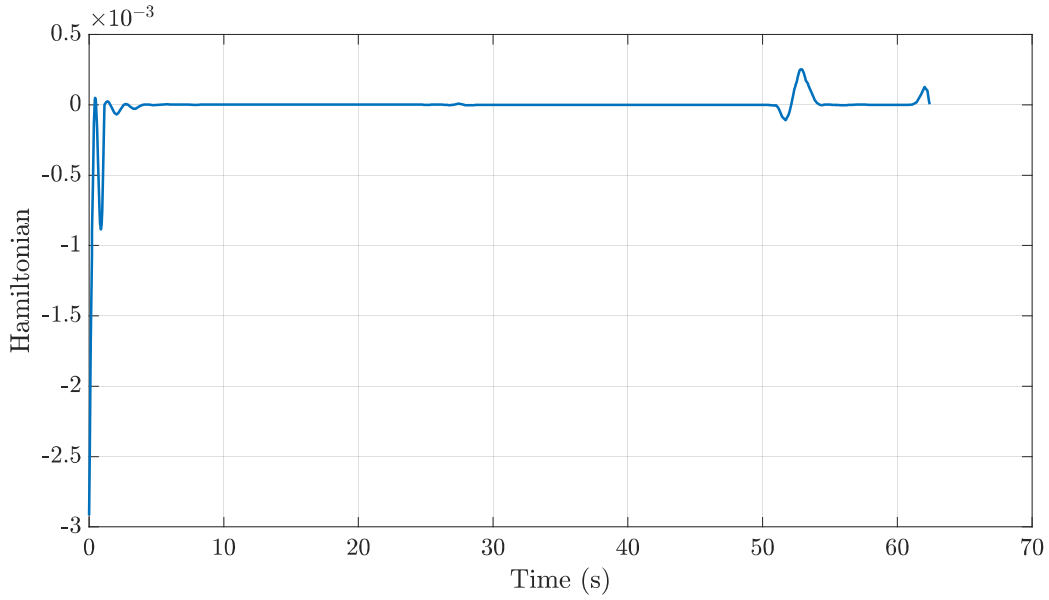


Figure 5.1: Reconstructed Hamiltonian

To verify the optimality of the trajectory, the previously introduced concept of the Hamiltonian in Section 3.2.1 can be used. According to [32] and [45], the Hamiltonian being zero along the trajectory is a necessary condition for optimality in an OCP that features an unspecified terminal time and where the Hamiltonian does not explicitly depend on time. This can be explained since when the Hamiltonian does not explicitly depend on time, it must remain constant along the optimal trajectory. And, for a problem with an unspecified terminal time, the transversality condition requires the Hamiltonian at the final time to be zero. Consequently, if H is constant and $H(t_f) = 0$, the Hamiltonian must be zero throughout the entire optimal trajectory. Therefore, by reconstructing the Hamiltonian using the trajectory solution (states and controls) and the computed costates provides an a posteriori verification of optimality. If the computed Hamiltonian remains close to zero across the trajectory, it confirms that the calculated solution is in accordance with this necessary optimality condition.

Analysing Figure 5.1, it is possible to observe that the reconstructed Hamiltonian remains very near zero, and so the solution is confirmed to be optimal.

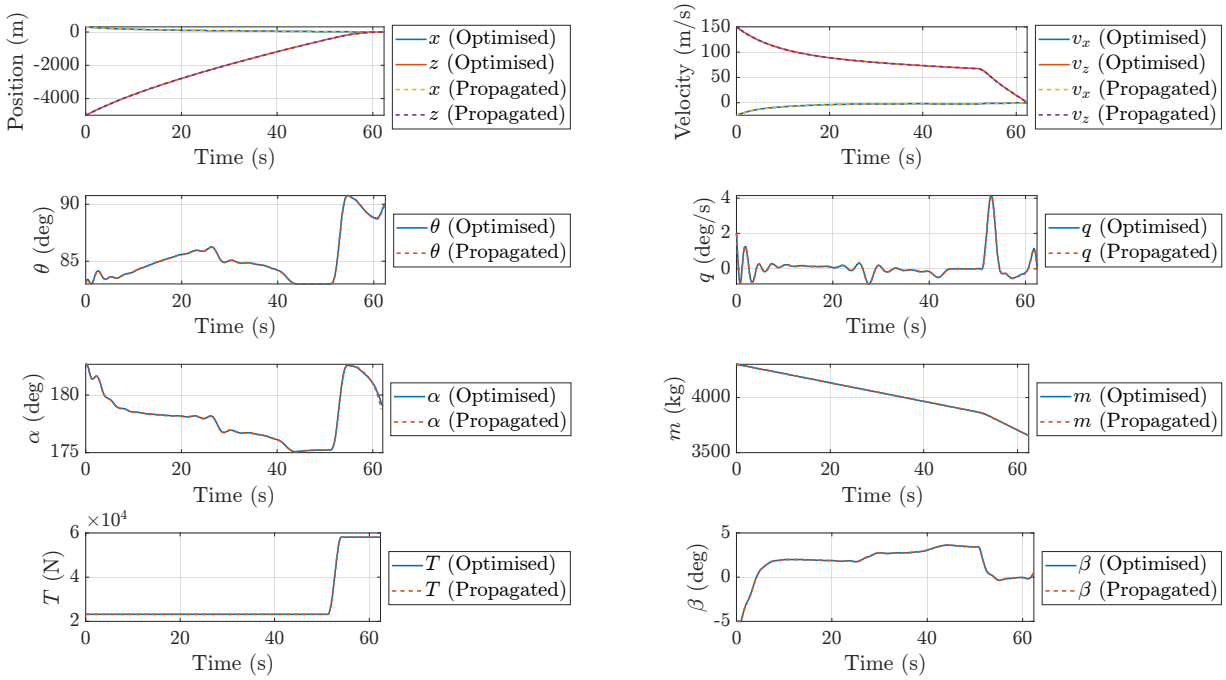


Figure 5.2: Time evolution of trajectory parameters, optimised vs propagated

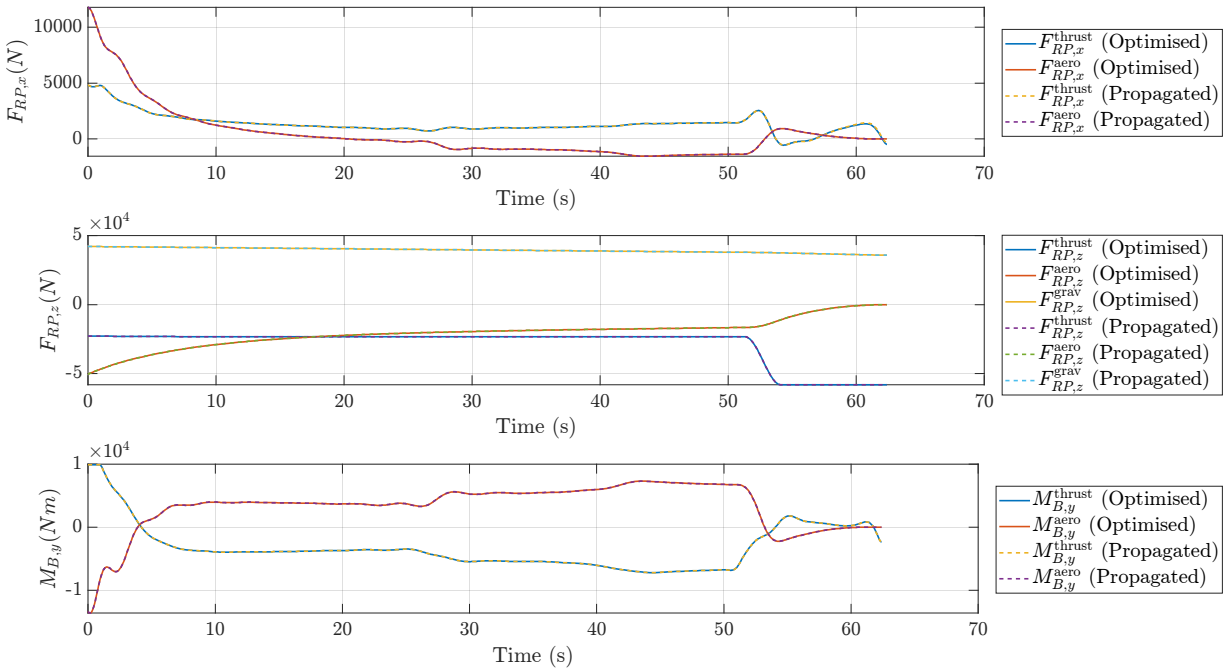


Figure 5.3: Time evolution of forces and moments, optimised vs propagated

Secondly, the feasibility of the trajectory must also be confirmed. To achieve this, the trajectory can be reconstructed by feeding the augmented control inputs ($\mathbf{u}_{\text{aug}} = [\dot{T}, \dot{\beta}]$) into the LV dynamics used to obtain the optimal solution, while applying the corresponding obtained initial conditions. This requires the use of a numerical algorithm to solve ODEs, which in this case is the Runge–Kutta 45. By propagating the system using this algorithm, a reconstructed trajectory is obtained that can be directly compared with

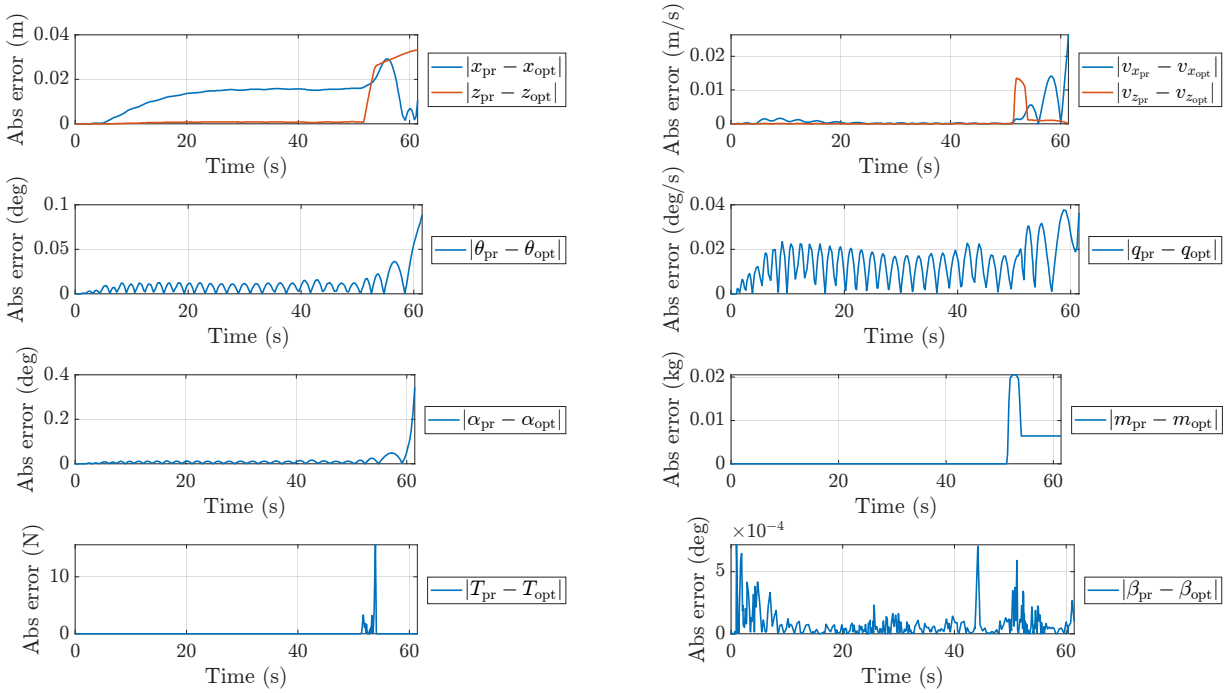


Figure 5.4: Time evolution of absolute error of trajectory variables, optimised vs propagated

the optimal solution. This comparison makes it possible to determine if the solution is consistent with the implemented dynamics, and therefore if it can be considered feasible.

In Figure 5.2 and Figure 5.3, the comparison between the trajectory variables, as well as the forces and moments, for both the optimal and the propagated solutions can be observed. The results from the two approaches are essentially coincident. To confirm this, the absolute error of the trajectory variables is computed and plotted in Figure 5.4, demonstrating the closeness between the solutions and therefore confirming the feasibility of the optimal trajectory.

5.2. Linearisation

The linearisation process is the basis for tuning the controller, and so it is necessary to verify its correctness. To achieve this, the analytical method used to derive the state-space matrices is compared with a numerical approach [57]. Considering f as the nonlinear function, the derivative of f with respect to each state component ($\mathbf{x}_i$) can be approximated using the central difference method, given by:

$$\frac{\partial f}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + \epsilon \mathbf{e}_i) - f(\mathbf{x} - \epsilon \mathbf{e}_i)}{2\epsilon} \tag{5.1}$$

Here, ϵ is a small perturbation (in this case, 10^{-5} is used), and $\mathbf{e}_i$ is a unit vector with a 1 in the i^{th} position and 0 elsewhere. By applying this perturbation, the change in the system's nonlinear output can be evaluated when a single state is slightly varied in both the positive and negative directions while all other states remain constant. The result provides an approximation of the partial derivative of the system with respect to that particular state. A similar procedure can be applied by perturbing the input vector, allowing the numerical estimation of the **B** matrix of the state-space representation.

To compare both matrices and understand the order of magnitude of each entry in the state-space matrices, the results obtained for ($t = 20$ s) are presented below:

$$A_{\text{an}} = \begin{bmatrix} 0 & 1.0000e+00 & 0 & 0 & 0 & 0 \\ 0 & -1.4402e-01 & 0 & -6.3832e-03 & 1.8183e+00 & 0 \\ 0 & 0 & 0 & 1.0000e+00 & 0 & 0 \\ 0 & -3.0422e-03 & 0 & -1.2146e-01 & 5.0325e-01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000e+00 \\ 0 & 3.2121e-02 & 0 & 3.6610e-03 & -2.8686e+00 & 0 \end{bmatrix} \tag{5.2}$$

$$A_{\text{num}} = \begin{bmatrix} 0 & 1.0000e+00 & 0 & 0 & 0 & 0 \\ 0 & -1.4402e-01 & 6.7163e-07 & -6.3832e-03 & 1.8183e+00 & 0 \\ 0 & 0 & 0 & 1.0000e+00 & 0 & 0 \\ 0 & -3.0422e-03 & -5.5215e-04 & -1.2146e-01 & 5.0325e-01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000e+00 \\ 0 & 3.2121e-02 & 9.1870e-06 & 3.6610e-03 & -2.8686e+00 & 0 \end{bmatrix}$$

$$B_{\text{an}} = \begin{bmatrix} 0 & 0 \\ 1.0740e-05 & -5.6140e+00 \\ 0 & 0 \\ -2.4177e-04 & -2.4939e-01 \\ 0 & 0 \\ -3.9771e-06 & -2.8241e+00 \end{bmatrix} \tag{5.3}$$

$$B_{\text{num}} = \begin{bmatrix} 0 & 0 \\ 1.0740e-05 & -5.6140e+00 \\ 0 & 0 \\ -2.4177e-04 & -2.4939e-01 \\ 0 & 0 \\ -3.9771e-06 & -2.8241e+00 \end{bmatrix} \tag{5.4}$$

It can be concluded that both methods produce very similar results, with the observed differences resulting from the fact that the numerical method is an approximation, while the analytical method relies on specific modelling assumptions. Additionally, another source of difference is due to the fact that in the analytical derivation, variations in certain parameters, such as the speed of sound and air density with altitude, are neglected because their influence on the linearised dynamics is expected to be minimal over the mission. As a consequence, the third column of the analytically derived matrix contains zeros, while the corresponding entries in the numerical matrix are non-zero, showing the subtle difference captured by the numerical perturbation. These non-zero values are several orders of magnitude smaller than the other terms of the system matrices, confirming that the neglected effects have only a small impact on the system's behaviour. This validates the simplifying assumptions used in the analytical approach and justifies their use for controller design.

To verify that both methods are sufficiently close throughout the entire trajectory, the mean difference of each matrix entry is computed over all time steps. The resulting errors, observable in Figure 5.5 and Figure 5.6, are always several orders of magnitude smaller than the corresponding matrix entries themselves, showing similarity in results between the analytical and numerical linearisation methods. This confirms that the derivation of the state space matrices and the simplifying assumptions used in the analytical derivation do not compromise the accuracy required for the controller design and that the linearised models are valid across the full trajectory.

5.3. Controller

To tune the controller, the LQR technique is used. To verify that the controller behaves as intended, namely, that the system errors are driven to zero with the desired behaviour defined by the LQR weighting matrices, the linear response of the system is studied by using the state space obtained with the linearisation.

This verification is repeated for the five points along the trajectory selected for gain scheduling. Following the same reasoning used when applying Bryson's rule in the paper, the same initial error is applied at all five points, even though the actual error would naturally decrease along the trajectory. However, because the analysis includes faulty conditions, it is impossible to predict the exact magnitude of the deviation that may occur at any stage. For this reason, the maximum admissible error (which corresponds to the largest

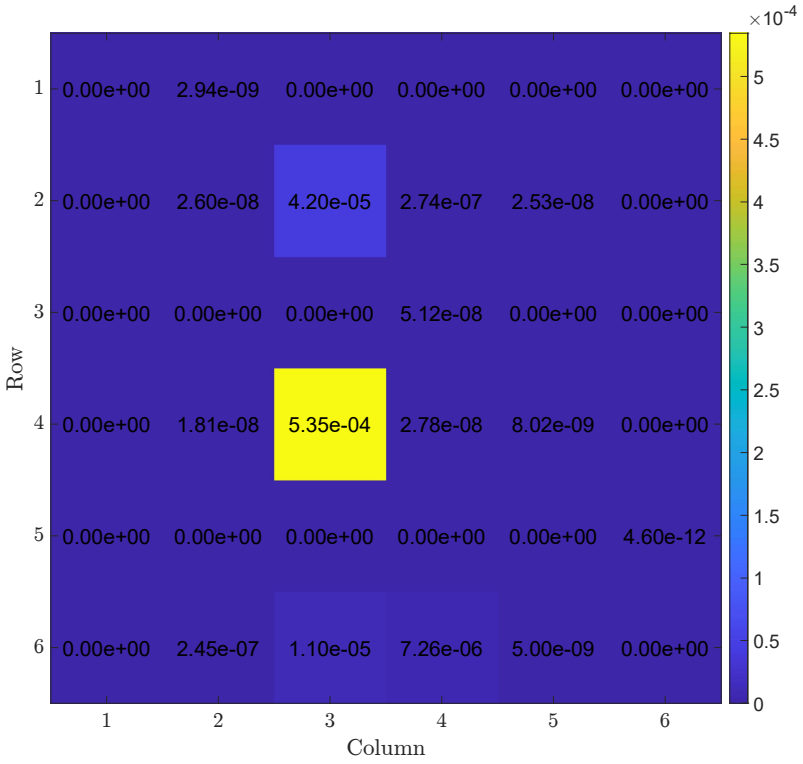


Figure 5.5: Mean difference between analytical and numerical linearisation methods of the **A** matrix throughout the trajectory

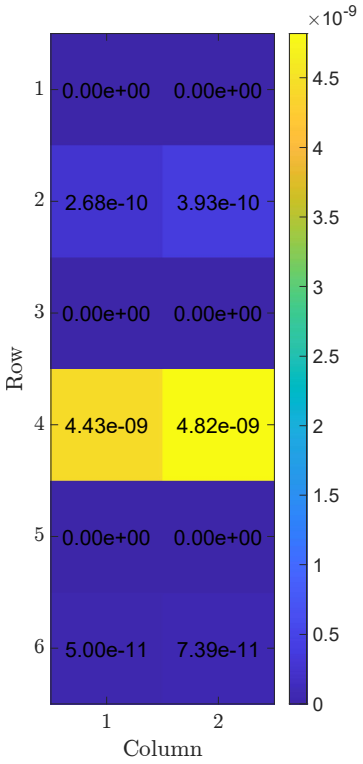


Figure 5.6: Mean difference between analytical and numerical linearisation methods of the **B** matrix throughout the trajectory

initial deviation) is evaluated at every scheduling point to ensure a good functioning of the controller in any circumstances.

In Figure 5.7, the differences between the equilibrium point of the linear system and the linear evolution of the system, starting from the specified errors, are shown. Overall, the plots indicate that the error consistently converges to zero, meaning that the system tends towards its equilibrium point and therefore towards the desired trajectory. The rate of error correction is satisfactory: even in the worst-case linear scenario, the system reaches zero error in less time than the duration of the full trajectory.

Another important aspect illustrated by the plot is whether the actuator responses remain within acceptable limits relative to their saturation bounds. This is not always the case, but violations occur only for the final two points of the trajectory. This is not expected to be a problem, as such large errors are unlikely to arise so late in the flight. Even if they did, the actuators would reasonably be driven to their limits when compensating for such an extreme condition.

This analysis is useful during the controller tuning process and provided an initial indication of the controller's influence at different points of the trajectory. Nevertheless, this verification is further confirmed by applying the controllers within the nonlinear simulator. As already shown in the final MC results of the paper, the controllers successfully drive the error to zero. In fact, the 1000 runs of the fault-free scenario in the MC campaign achieved a 100% success rate, even with initial conditions distributed across the maximum deviations considered, which confirms the well functioning of the controller.

5.4. Control Allocation

In this project, control allocation is performed using a nonlinear optimiser. However, several algorithms can be employed to accomplish this task with good results for TVC and thrust allocation [20]. To confirm successful allocation, the control allocation problem is also solved using a convex programming algorithm

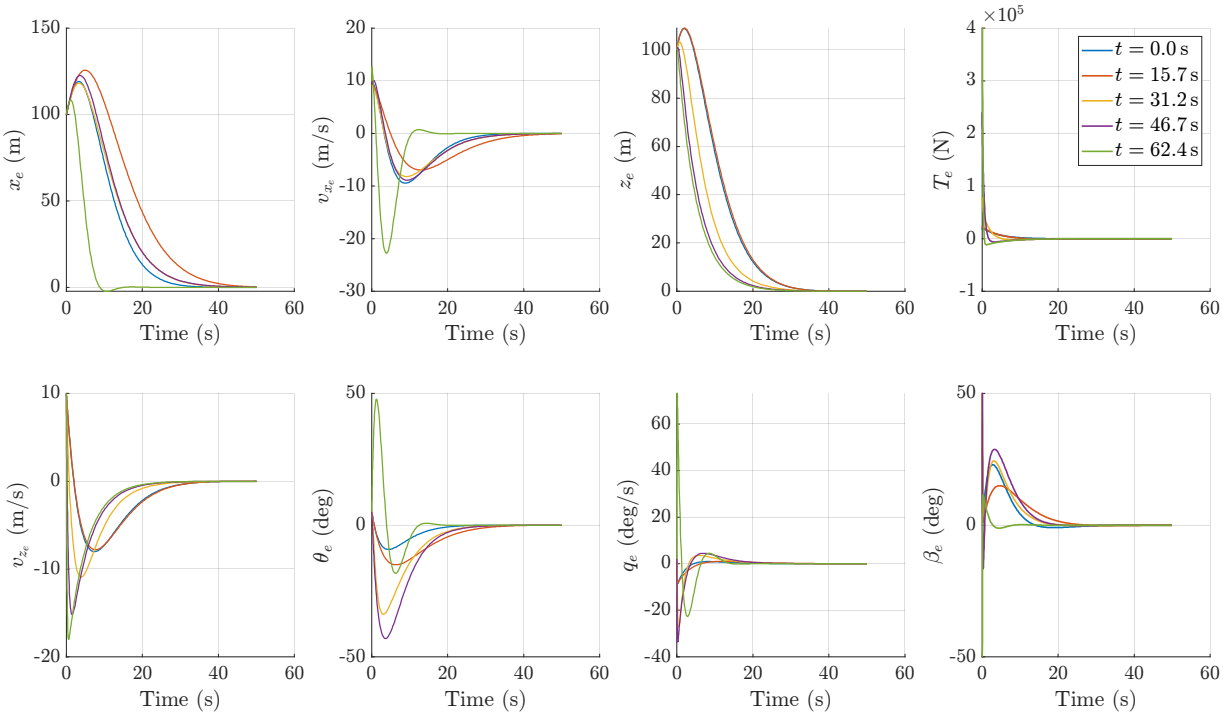


Figure 5.7: Error evolution of closed loop linear system

to check if similar results could be achieved.

For this new algorithm, the overall formulation, including the decision variables, the objective and the post-processing, is maintained as in the problem described in the paper. However, some of the constraints used with the nonlinear optimiser are incompatible with a convex problem formulation, so an alternative convex-safe constraint set is used. In particular, the box bounds previously applied as lower and upper limits cannot be entirely preserved. Instead, the convex requirement $f_{x,i,B} \geq T_{\text{min}}$ is imposed, which is a sufficient condition for the original minimum-thrust requirement since: $\sqrt{f_{x,i,B}^2 + f_{z,i,B}^2} \geq f_{x,i,B} \geq T_{\text{min}}$. So any vector satisfying the imposed inequality automatically satisfies the original lower bound on the thrust magnitude. For the upper bound of the thrust and the limits of the gimbal angle, the already used constraints are sufficient: magnitude and gimbal cone.

Adopting this new constraint preserves convexity of the feasible set but introduces a more conservative approach: it excludes gimballed vectors whose x component is below T_{min} even though their total magnitude would meet the original requirement with the addition of the z component. In practice, this is acceptable (and is exact for the central thruster, since $\beta_{\text{max},1} = 0$), because the x component is substantially larger than the z component as the gimbal angle is limited to a maximum of 10 deg. Consequently, the practical impact on the solution is not that relevant.

To test both allocation methods, the same fault scenario used to test the nonlinear optimiser allocation in the paper is reproduced. For clarity, the fault case considered is repeated here:

- Jamming of engine 3 at 1 degree at 20 seconds.
- Total loss of engine 2 at 40 seconds.
- Partial loss of 80% of engine 4 at 55 seconds.

By examining Figure 5.8, which compares the total forces and moments produced by both allocation methods with the desired values from the optimised trajectory, and Figure 5.9, which shows the absolute errors between each method's outputs and the desired forces and moments, several conclusions can be drawn. The allocation results produced by both algorithms are extremely similar, with only minor differences during the faulty conditions. Moreover, the error remains zero while the jamming fault is occurring and during the total loss of engine power, until the point at which the desired thrust from the trajectory reaches

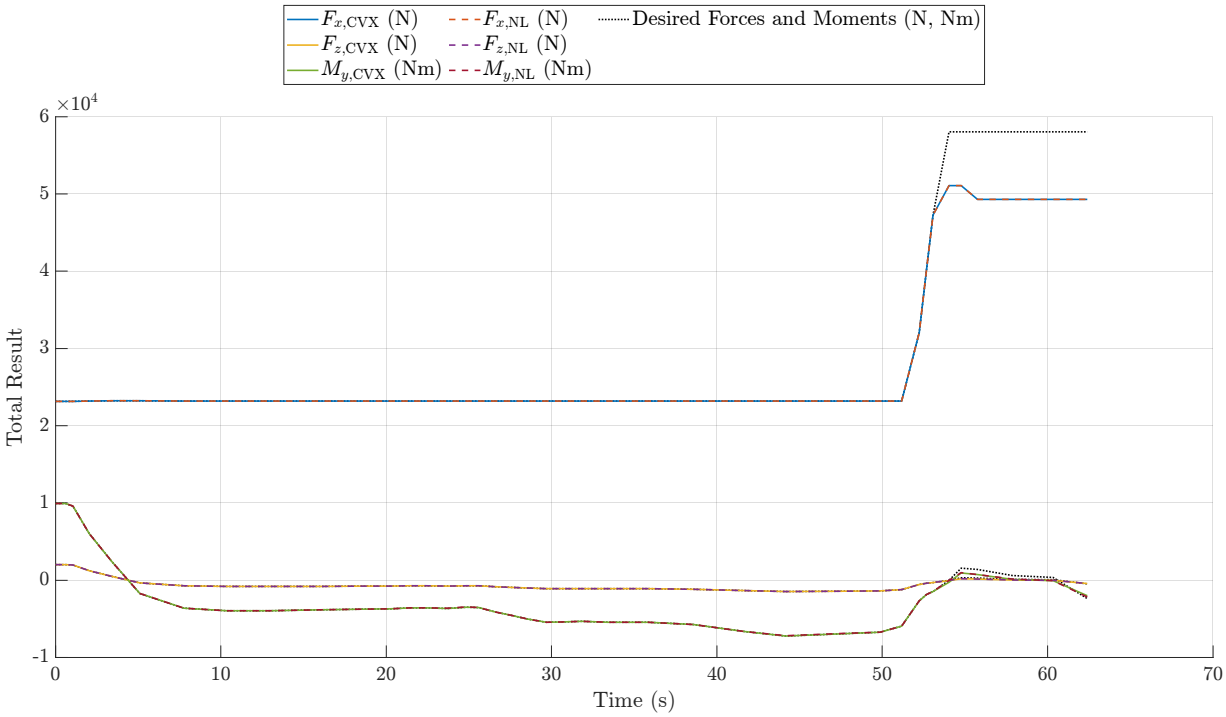


Figure 5.8: Allocation algorithms result comparison for specific fault scenario

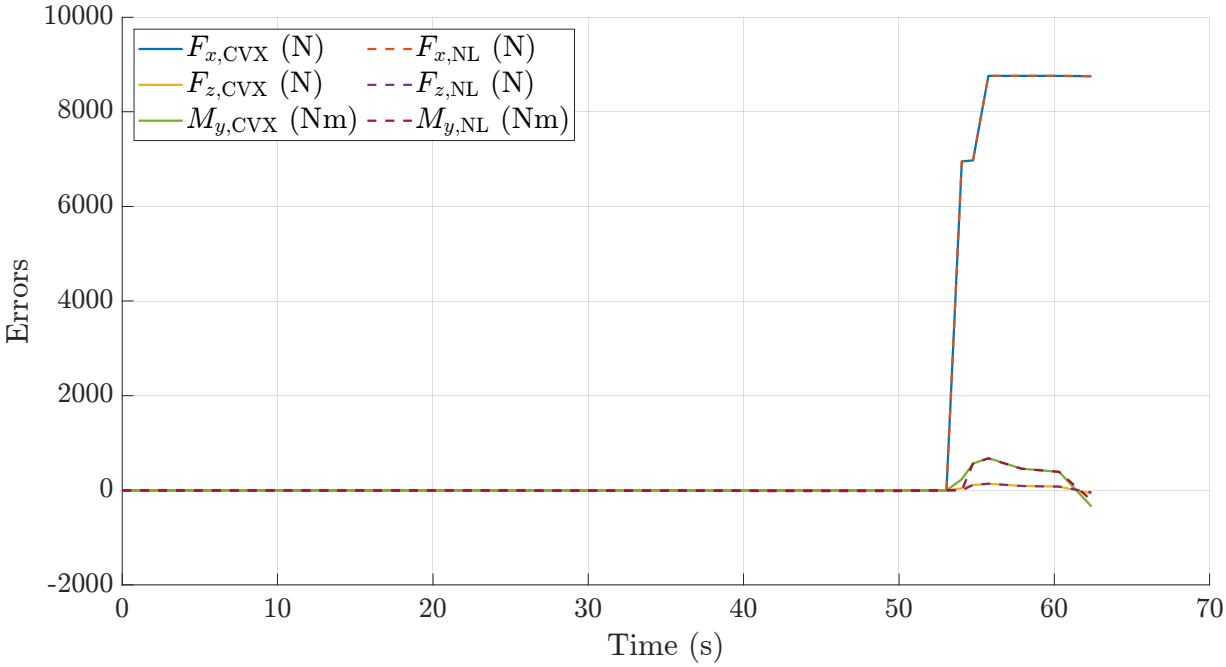


Figure 5.9: Allocation algorithms absolute error comparison for specific fault scenario

its maximum value. Once the trajectory demands become more challenging, the errors are no longer zero, with a further degradation observed when the partial loss of engine power also appears.

With these results, it can be confirmed that both allocation algorithms produce similar results, and the nonlinear allocator behaviour was already shown to make sense according to the example provided in the paper. Therefore, both allocation methods could be used. The final decision to use the nonlinear optimiser is due to its better computational performance compared with the convex approach. This speed advantage

occurs from the specific characteristics of the selected optimiser's algorithm's ability to generate dedicated solver code automatically, which leads to more efficient execution during simulation [20].

5.5. Nonlinear Simulator

In this project, a nonlinear simulator is developed to test the RLV. The simulator implements the nonlinear equations presented in the referenced paper, as well as additional blocks to model the controller structure, the control allocation, and the evolution of the vehicle's CG position and inertia based on the substance consumption of each tank. To validate its behaviour, the thrust, T, and gimbal angle, β, obtained from the optimised trajectory are fed directly into the simulator, so it acts as an open loop, and the resulting states are compared with the trajectory outputs to confirm their consistency.

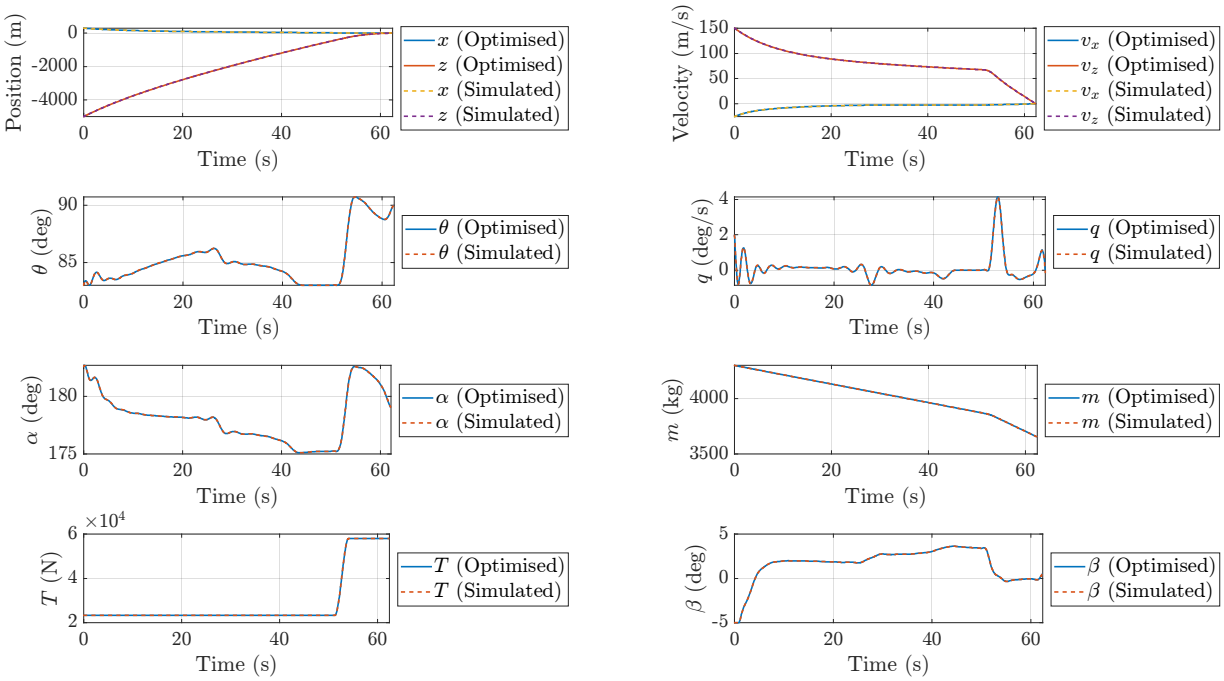


Figure 5.10: Time evolution of trajectory variables, optimised vs simulated

For this comparison, the simulator does not use the controller and does not perform control allocation. The LV is assumed to have a single engine capable of accepting the full set of inputs without any degradation in response. In addition, the simulator's capability to update the vehicle's inertia and CG based on fuel consumption is disabled, as this functionality is not included in the trajectory optimisation and its effects are supposed to be handled by the controller in the actual MC runs. In this comparison case, the fixed CG and inertia values used during trajectory optimisation are applied.

The comparison begins with an analysis of Figure 5.10 and Figure 5.11, where it can be observed that the optimised and simulated results are identical. To reinforce this conclusion, the absolute errors of several trajectory variables are also plotted in Figure 5.12, confirming the close agreement between the two. This comparison confirms the good functioning of the nonlinear simulator that is going to be used in the MC campaign, which guarantees that the results produced by it are physically correct.

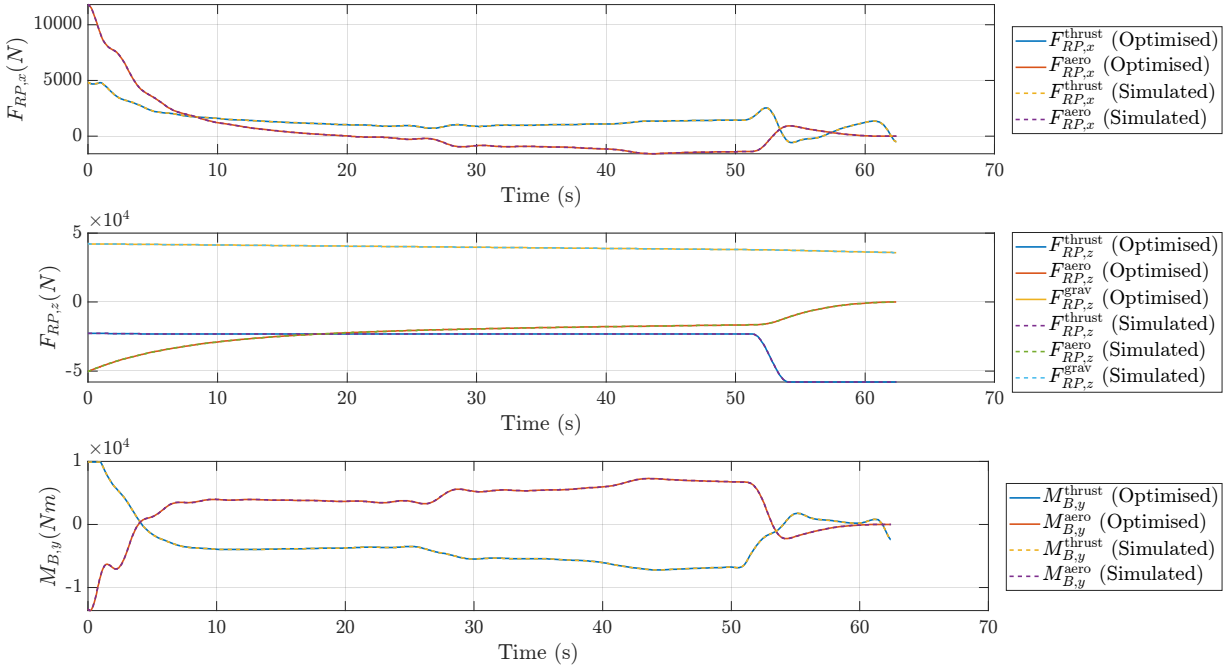


Figure 5.11: Time evolution of forces and moments, optimised vs simulated

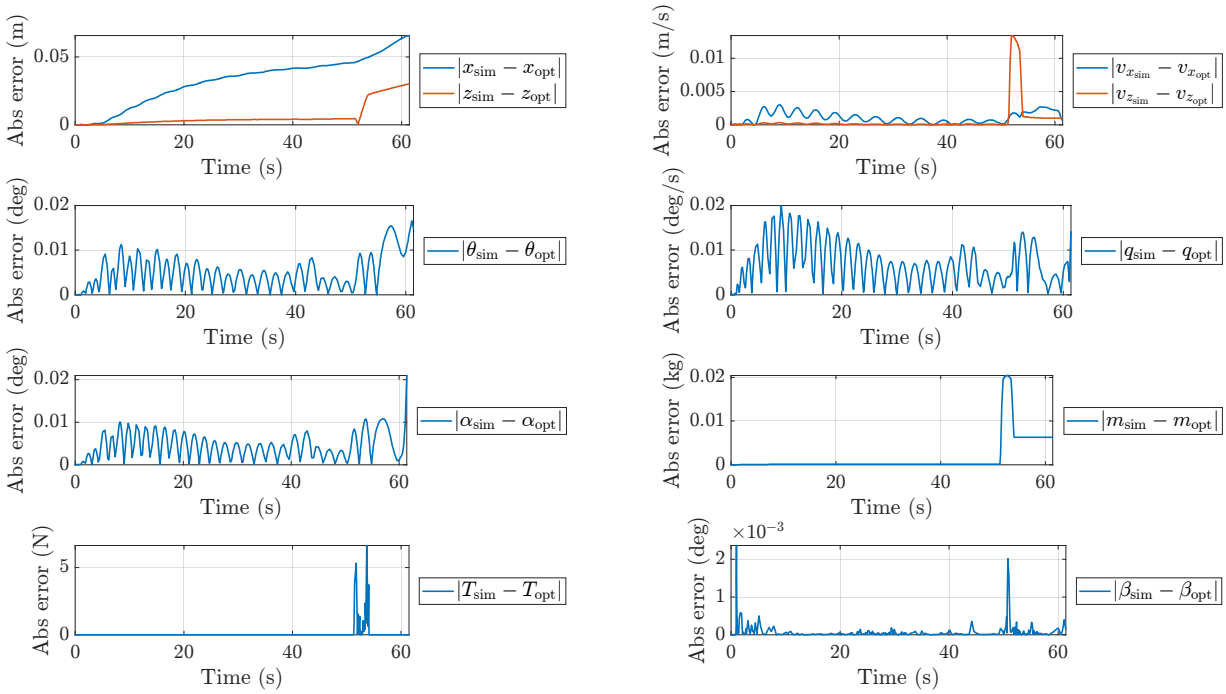


Figure 5.12: Time evolution of absolute error of trajectory parameters, optimised vs simulated

Part III

Closure

6

Conclusion

In this study, an RLV equipped with clustered engines is used to evaluate how a fault-tolerant control allocation strategy that is aware of actuator faults affects the outcome of the powered descent and landing phase, compared with an allocation scheme that is not aware of faults. The main objective is to determine if and how using fault-awareness in the allocation process improves the success of the mission outcomes.

In addition, the work explores how the characteristics of engine faults (timing, magnitude, affected engine, and type, such as jamming, partial thrust loss or total power loss) influence vehicle behaviour and mission success. These fault characteristics are not studied in isolation, they are studied in addition to the allocation strategies so that the interaction between fault type and allocation logic is understood. To achieve this, the study uses Monte Carlo testing in a nonlinear simulation environment to evaluate performance across a set of initial conditions and fault cases.

6.1. Revisiting the Research Questions

The research questions introduced in Section 1.5 are now revisited, taking into account the results obtained, in order to provide answers based on the outcomes of the study.

6.1.1. Research Question 1

Research Question 1

How do different fault conditions and the application of a fault-aware control allocation strategy influence the recovery performance, behaviour, and overall mission success of an RLV?

Subquestions:

1. How do different types of faults affect the vehicle's dynamic behaviour and its potential for successful post-fault recovery?
2. How do the magnitude, timing, and affected engine of a fault influence performance degradation and recoverability?
3. To what extent does a fault-aware control allocation technique improve recovery performance under various fault scenarios?

The response to this research question will allow for achieving the majority of the research objective. It will focus more on the effects of fault-aware allocation and the effects of different faults on the system's ability to perform a soft landing.

Focusing on the general research question by analysis of the results, it is possible to confirm that some different fault scenarios and the use of a fault-aware allocation highly influence the success probability, behaviour and performance of the RLV. The more detailed analysis for this answer will be given in the answers to each subquestion.

It is important to notice that, for this study, the LQR controller is used, but the answers to this research question would probably be similar for other control techniques, since, in general, a controller would always

try to bring the state errors to zero, even if with different performances. Therefore, the success rates and final results would differ slightly, but the main conclusions of this research question would be the same, as the fault impacts and the influence of different control allocations would have similar effects on the rocket.

Research Subquestion 1
Before examining the effects of applying the fault-aware allocation, it is necessary to understand how each type of fault influences the behaviour of the LV and its ability to achieve an acceptable landing. This provides the necessary baseline for interpreting the benefits of fault-awareness in the allocation strategy.

To address this aspect, both the detailed analysis of an individual representative run presented in Section 4.1 and the global success statistics discussed in Part I are considered. Together, these analyses offer information about both the physical response of the system and its overall performance.

Jamming Scenario

For the jamming fault scenario, the rocket's behaviour generally does not change by a lot following the failure, except under very specific and extreme conditions, which will be analysed in detail in subquestion 2.

In the majority of cases, however, the behaviour observed in the representative individual run, in Section 4.1, can be taken as an example. In this illustrative run, the attitude response is only slightly affected at two instants: at the moment the fault occurs and during a rapid change in thrust demand. In both allocation cases, the controller remains capable of stabilising the vehicle and maintaining acceptable tracking performance. This explains the high success rates observed for jamming faults in the Monte Carlo campaign.

Partial Engine Loss Scenario

On the other hand, the partial engine loss fault has a greater impact on the probability of mission success. The success of the LV mission is sometimes compromised when the total thrust generated by the engines is insufficient to match the thrust demanded by the guidance system. While this situation may happen at different points along the trajectory, for the specific mission considered in this thesis, it occurs most critically when the thrust demand reaches its maximum at around $t = 51$ s, since it is usually from this time on that the thrust provided by the engines is not sufficient.

An illustrative example of the vehicle response on the limit of thrust insufficiency is provided by the analysed individual run. This case shows the following behaviour: the immediate effect is a rapid increase in the velocity and position errors. Moreover, depending on the allocation strategy in use, this rise in tracking errors may cause the controller and allocation algorithm to place excessive focus on correcting the translational errors, at the expense of attitude. This can then induce oscillations in the attitude states as well, further degrading overall performance and, in the most severe cases, leading to an unsuccessful landing.

As observed in the individual case analysis for the fault-unaware strategy in Figure 4.5, whenever an increase in attitude errors occurs, the system attempts to correct them by commanding variations in the gimbal angles. While this action successfully brings the attitude back to acceptable values, it does so by disregarding translational control, as a part of the available thrust is redirected away from the vertical direction. Consequently, the correction of attitude errors compromises the system's ability to reduce the vertical velocity and position errors, thereby worsening the overall recovery capability under lack of thrust conditions.

Total Engine Loss Scenario

The total engine loss scenario represents the most severe fault condition considered and leads, in practice, to mission failure in almost every run. This is expected, as such a fault makes the problem physically unrecoverable by removing a substantial portion of the available thrust. As a result, the success rate is close to zero for both allocation strategies.

From another perspective, the analysis of the representative individual run again provides insight. The behaviour observed in this case is similar to that of the partial engine loss scenario, but with significantly more severe consequences. Before the high thrust commanded by the guidance, the vehicle is still able to follow the reference trajectory reasonably well. However, once this command is reached, the available

thrust is insufficient, and both the translational and attitude errors begin to grow. In this case, the vehicle is considered always unable to achieve a soft landing, and the errors continue to grow until the impact of the LV with the ground.

General analysis

It is important to clarify what can be generalised from these conclusions. The results show that jamming faults have only a limited influence on overall mission success, while engine loss faults have a much more significant impact, with total engine loss being, as expected, more severe than partial loss. However, it must be taken into account that the specific numerical success rates obtained in this study are mission-dependent. A different trajectory would lead to different quantitative results, even though a conventional powered descent and landing profile would probably preserve the same qualitative ranking in terms of fault severity and influence.

With regard to the system behaviour under faults, two critical moments are clearly identified for the jamming scenario: the instant at which the fault occurs, and periods when the guidance commands extreme actuator demands, either in gimbal angle or in thrust. In both situations, temporary increases in attitude errors may occur, but in the vast majority of cases, these errors are attenuated and do not compromise mission success.

By contrast, for engine loss scenarios, the critical moment affecting the vehicle behaviour is the moment at which the thrust demanded by the guidance exceeds the maximum thrust that can be delivered by the remaining healthy engines. Under these conditions, both translational and attitude errors grow. Whether these errors can subsequently be recovered depends on the characteristics of the fault. In the most severe cases, particularly for total engine loss, recovery becomes physically impossible, leading inevitably to mission failure.

Research Subquestion 2
In this subquestion, the objective is to understand the impact of the fault characteristics. To achieve this, the influence of each characteristic is analysed and discussed separately.

Magnitude

Starting with fault magnitude, which for the jamming case is interpreted as the gimbal angle at which the engine becomes locked, and for the engine loss case as the level of available thrust remaining after the fault, this parameter is, as expected, the characteristic with the greatest influence on the mission outcome.

For the jamming fault, only a limited number of extreme cases lead to mission failure or to significant difficulties in correcting the errors caused by the fault. These extreme conditions occur when a high jamming angle is combined with an unfavourable timing, which is discussed in more detail later. In the rest of the cases, the system remains capable of compensating for the disturbance and completing the landing successfully.

Jamming faults primarily affect the vehicle through the generation of unwanted moments, which must be counteracted by the remaining healthy engines. When the jammed angle is close to the allowable limit, the system experiences greater difficulty in dealing with the fault. The resulting increase in attitude errors can divert the controller's attention away from other critical states, such as translational motion, therefore leading to degradation of performance and, in extreme cases, mission failure.

For this specific mission, most of the observed failures occur when the jamming angle is locked at the negative extreme. However, this outcome is related to the particular reference trajectory and to the two-dimensional nature of the problem considered here. In a fully three-dimensional scenario, a specific jamming angle might again prove critical, but the additional degree of freedom provided by the vehicle roll could potentially be used to counteract its effect. Consequently, this observation cannot be generalised and should be regarded as a study-specific result rather than a universally valid conclusion.

In the case of engine loss, the fault magnitude plays a decisive role, with thrust-availability thresholds being identified for each allocation strategy. Below these thresholds, successful recovery becomes impossible regardless of the control effort, while above them, the mission can still be completed successfully. These threshold values are, once again, mission-dependent and therefore not directly generalisable. Nevertheless, it is reasonable to conclude that similar thresholds will exist for any mission conducted under

similar conditions, as they are related to the balance between the thrust required by the guidance and the thrust that can be delivered by the remaining healthy actuators.

Time of Occurence

The time of occurrence plays a comparatively minor role in all fault scenarios. However, several observations can be made.

For the jamming case, as previously mentioned, only a small number of extreme situations result in a non–soft landing. One of the contributing factors in these cases is the timing of the fault. If the jam occurs before the guidance demands the largest gimbal deflection of the trajectory, the system may face difficulties in the descent, potentially making recovery more difficult. This conclusion is, once again, trajectory-dependent and may differ for other missions with different profiles. Nevertheless, it can be stated that, for jamming faults, the moment at which the fault occurs can influence the outcome.

In contrast, for the engine loss scenario, the time of occurrence is less critical. In a powered descent and landing mission focused on fuel saving, the thrust demand increases towards the final phase of the descent. Therefore, regardless of when the fault takes place, its effects will be felt in the final seconds, when the engines are required to deliver high thrust. Only in the extreme case where the fault occurs very close to touchdown, leaving insufficient time for its effects to accumulate, does the timing become beneficial. In all other cases, the system may be exposed to a situation in which the available thrust is insufficient, meaning that the precise timing of the fault has little influence on the final outcome.

Faulty Engine

Lastly, the influence of the specific engine in which the fault occurs is examined.

For the jamming scenario, a clear relation can be noted between the sign of the jammed gimbal angle and which engines are more likely to produce failures. When the gimbal jammed at a negative angle, failures occurred almost exclusively in the out-of-plane engines or in engine 5. On the other hand, when the jam occurred at a positive angle, failures are associated with the out-of-plane engines or engine 3.

This behaviour is explained by the moments generated by the jammed thrust vector. A negative jamming angle produces a moment that normally would be mostly counteracted by the moment contribution of engine 5. If engine 5 is jammed at a negative angle, this compensating capability is reduced, making the vehicle more susceptible to attitude instability. The opposite argument applies for a positive jamming angle, where engine 3 has the compensating role.

While this explanation is strongly related to the geometry of this specific mission configuration, it illustrates how the interaction between thrust vector direction and engine placement can determine which engines become most critical under jamming faults. Once again, it is important to note that in a fully 3-D scenario, these effects could be significantly reduced. The vehicle would be able to use roll motion to its advantage, redistributing the thrust vectoring requirements across engines, reducing the sensitivity to which specific engine becomes jammed and at what angle.

In the engine loss scenarios, no evidence is found that the specific engine in which the fault occurs has any meaningful influence on the mission outcome. The small variations observed in the Monte Carlo results are attributable to the stochastic nature of the campaign rather than to the engine's location.

Research Subquestion 3

The global results presented in Part I show that, for all fault types, the use of the fault-aware allocation algorithm results in an improvement in mission success rates. The partial engine loss scenario is where this difference is most significant: the fault-aware approach substantially decreases the minimum available thrust required for a successful landing. The jamming case also benefits from the fault-aware allocation, although the improvement is smaller because the fault-unaware algorithm already performs very well, and a few very specific extreme cases, previously discussed, remain challenging to fully correct. Finally, for the total engine loss scenario, the improvement is marginal, as recovery from this fault is nearly impossible. Only a single additional successful run is observed.

In the jamming case, the improvement happens because, during the critical moments identified for this fault type, the informed allocation does not ”wait” for attitude errors to grow before responding. Instead, by recognising the damaged engine's reduced action, the algorithm redistributes control effort toward the healthy engines, allowing the system to correct deviations more smoothly and rapidly.

A similar mechanism explains the improvements observed for the engine loss scenarios. Although velocity errors increase in both allocation strategies, the fault-aware algorithm dissipates these errors more effectively while avoiding unnecessary deterioration of the attitude states. This preserved attitude small error, helps the system in recovering from the faulty condition and contributes to the overall higher success rate.

6.1.2. Research Question 2

Research Question 2

To what extent can a gain-scheduled Linear Quadratic Regulator synthesised considering a linearised rocket dynamics across the descent trajectory, maintain stability and acceptable performance under various fault scenarios during a descent and landing mission?

This project's emphasis is on the allocation algorithm. Hence, the control law is chosen to be an LQR optimal controller since LQR is a well-established technique whose properties and tuning effects are easy to interpret, which is advantageous here because the controller must be tuned differently for different mission moments. The controller is tuned according to the reasoning described in the paper, with weightings selected to prioritise the states that are most critical at each moment of the trajectory. This reasoning behind the tuning helped maintain a clear relation between design choices and vehicle behaviour.

Despite LQR being a standard method, it is not commonly applied to faulted powered descent and landing scenarios, and there is little prior work specifically addressing such cases. That motivated confirmation whether an LQR-based controller is adequate for the problem at hand. To that end, the controller is validated both in linear analyses and in the nonlinear simulator. In each case, its response is examined across the range of different initial conditions.

In the fault-free MC campaign, every run achieved a soft landing despite the spread in initial states and parameter uncertainties. This result demonstrates the controller's effectiveness under nominal operating conditions. When faults are introduced, performance depended on fault type and magnitude: success rates remained very high for jamming faults but fell noticeably for cases involving engine thrust losses. An inspection of the failed runs showed that most failures could be attributed to the magnitude or timing of the fault, rather than to any flaw in the controller. In each case, a clear reason could be identified, for example, a loss of thrust or an extreme jamming case, that explained why the vehicle could not be recovered. In other words, in many unsuccessful trials, the physics of the fault itself constrained achievable performance and therefore limited the chance of recovery.

Therefore, the controller is naturally not perfect, and alternative tunings or different control architectures could lead to improvements. However, the results suggest that only slight improvements are likely in many scenarios, because some fault cases simply reduce the actuator authority to a level where no controller could meet the original trajectory requirements. Overall, the evidence indicates that the chosen LQR is not a relevant limiting factor: it maintained stability across the tested scenarios, with only two divergent cases out of 7000 runs presented in the paper, which can be regarded as negligible, and it delivered acceptable performance for all fault scenarios in which trajectory correction is physically possible.

Finally, it is important to consider the study's modelling assumptions. Effects such as wind gusts, sensor noise, actuator disturbances, propellant slosh and structural flexibility are not included in the simulations. Incorporating these effects would increase uncertainty and could make controllers with robustness properties (for example, H_∞ methods) more suitable. This does not alter the reached conclusion that, for the specific case under study, the applied control solution is adequate.

It is important to notice that the obtained results could be improved in case the guidance is reconfigured upon detection of the fault, since in many cases the problem lies in the lack of adaptability of the trajectory for the current faulty scenario. For example, for some thrust loss cases, the thrust necessary to perform a soft landing is not available, which in some situations could be achieved with a different trajectory. This topic is further explained in Section 7.4.

6.2. Closing Remarks

To present the closing remarks, it is useful to first revisit the overall research objective:

Research Objective

Develop and evaluate a fault-tolerant control allocation strategy for reusable launch vehicles with a cluster of throttleable rocket engines, and analyse the effects of the control technique and fault conditions on vehicle performance and recovery capability, including quantifying the impact and severity of faults across various flight instants of the descent to enhance system resilience.

Throughout this work, an FTC allocation strategy is successfully implemented by including fault awareness into the allocation algorithm, therefore enabling more appropriate and effective control actions in the presence of faults. To the best of the author's knowledge, no prior literature has examined the performance of such algorithms in faulty powered descent and landing scenarios for reusable launch vehicles. This thesis therefore provides a first demonstration of the benefits of fault-aware allocation across multiple fault types and characteristics in a descent scenario.

To support this analysis, a representative powered descent and landing mission is designed to reflect the behaviour of a generic rocket descent and landing trajectory, allowing the results to be meaningfully extrapolated beyond the specific scenario studied. With this framework in place, an investigation of the control/allocation architecture, the effects of different faults and how these faults interact with this architecture is also performed.

All in all, the research objective has been successfully achieved. The results obtained confirm the effectiveness of the studied allocation technique and provide insight into its behaviour under various faulty conditions. These results and conclusions not only demonstrate the practical benefits of including fault awareness into allocation algorithms but also offer a basis for further developments in this area. Hopefully, this work can contribute, even in a modest way, to RLV technologies in general and to the effort of improving the reliability of landing launch systems.

7

Recommendations for Future Work

Taking this study into account, some ideas for future work can be suggested. Accordingly, this chapter presents a set of recommendations with the objective of either improving the work done in this thesis or expanding the work toward more advanced research directions.

7.1. Fidelity Improvements

Since the time available for this project is limited, several simplifications are made to keep the problem doable. These choices do not compromise the validity of the results or the main conclusions, particularly regarding the demonstrated influence and benefits of fault-aware allocation. However, for the development of a real launch vehicle or for determining precise operational thresholds for each fault type, further extensions would be required.

First, the analysis is carried out using a 3-DoF model. A possible next step would be to extend it to a full 6-DoF model, enabling the study of coupled translational and rotational dynamics and more realistic actuator interactions. Additionally, there are physical effects that are not included in the thesis but would be essential in a high-fidelity study, such as wind gusts, sensor noise, actuator disturbances, propellant slosh, and structural flexibility. If these effects were added to the model, this would allow the framework to have a better real-world representation and provide deeper insight into the performance of the FTC allocation strategy.

7.2. Robust Controller

With the inclusion of the additional effects mentioned in Section 7.1, requiring the study to be carried out in a higher-fidelity simulation environment, it would also be relevant to explore alternative control techniques, particularly those designed to deliver stronger robustness guarantees. Methods such as H_∞ control could be used to account for unmodelled dynamics, external disturbances, and increased uncertainty. The integration of these types of control strategies with the fault-aware allocation framework could lead to a more resilient overall system and provide a better perception into how the controller allocator interaction behaves under more realistic conditions.

7.3. Online Controller Reconfiguration

Considering the results obtained, it would be valuable to investigate how the controller parameters could be reconfigured in real time based on the specific fault that occurs. A strategy like this could allow the controller to prioritise different performance objectives, such as attitude stability or position and velocity tracking, according to the mission needs and the identified fault.

As observed in this study, the jamming-fault failures are primarily caused by excessive horizontal landing velocity or excessive pitch angle at touchdown. These issues could probably be mitigated through a reconfiguration of the controller that relaxes certain less critical objectives while tightening others, with the objective of keeping all states within acceptable limits. A similar argument applies to the engine-loss scenarios, although cases where the total available thrust is physically insufficient cannot be recovered, near threshold cases could benefit from a controller that temporarily prioritises vertical velocity over other states.

This type of controller reconfiguration could improve system resilience, making sure that control authority is allocated in a more effective way for each fault scenario.

7.4. Online Guidance Reconfiguration

As the controller can be reconfigured during flight, the same principle can be applied to the guidance system. The nominal trajectory used in this study is designed to be optimal under fully healthy conditions. However, when a fault occurs, those conditions are no longer met, and it may be advantageous to update the trajectory online. The results of this thesis strongly suggest that the integration of this work with a type of guidance reconfiguration could result in performance improvements.

For the jamming case, the benefit would likely be marginal, especially considering that a combination of allocation adjustments and controller reconfiguration would already bring the success rate very close to 100%. Nevertheless, trajectory adaptation could still help in the few extreme cases that remain problematic.

In contrast, for the engine loss scenarios, the improvement from online guidance reconfiguration would probably be substantial. As observed in this mission, even when an engine loses thrust, the guidance always requests maximum thrust in approximately the final 10 seconds of flight. If the guidance were allowed to re-optimise the trajectory at the moment of the fault, it could initiate the high-thrust phase earlier and request lower thrust levels for a longer period of time, avoiding many of the failures caused by excessive final vertical velocity. In some cases, this approach might even make certain total engine loss scenarios recoverable, in cases in which the fault occurs early enough to compute a feasible alternative trajectory.

Of course, this recovery would come at the cost of increased fuel consumption, since the nominal trajectory is optimised to minimise propellant usage. However, if sufficient fuel remains, spending more of it is preferable to mission failure.

7.4.1. Objective Reconfiguration Using Online Guidance Reconfiguration

As explained in Section 3.4.1, if the original objective of a mission cannot be reached in a faulty situation, the system is not fault-tolerant for the current fault. However, this does not mean the objective cannot be adapted. If a real mission were being developed, it is also worth considering that, in certain situations, the mission objective itself might need to change. A direction for future work would be to study alternative landing locations when the nominal mission becomes infeasible due to a fault. For example, the optimal placement of secondary landing pads that the vehicle could target when a soft landing at the primary site is no longer achievable could be studied. In a real mission, it would have to be taken into account that there would exist several practical constraints in the selection of such alternative landing locations that would have to be taken into consideration.

Such a study could identify regions where diverting the flight provides the highest probability of recovery of the LV, taking into account the fault type, vehicle capability, and remaining fuel. In scenarios where the main landing is no longer achievable, the rocket could switch to a fault-aware guidance mode as explained in Section 7.4 and follow an online computed trajectory in the direction of an alternative pad.

This would need to be combined with online trajectory reconfiguration (Section 7.4) and potentially with the controller adaptations mentioned earlier (Section 7.3). Together, these strategies would create a safer landing system capable not only of mitigating faults but also of redefining mission objectives when necessary.

7.5. Trajectory Optimisation Accounting for Faults

Using any form of online reconfiguration has a limitation: if the fault occurs too late in the trajectory, recovery may be impossible, regardless of the strategy used. This issue is worsened by the fact that neither guidance nor control reconfiguration is instantaneous. Even a small delay can critically reduce the recoverable envelope of the mission.

With this in mind, it would be valuable to study how the nominal trajectory optimisation could be adapted to maximise the probability of mission success in the presence of faults. This analysis would investigate a trade-off between optimality and robustness to determine how much fuel and time should be spent in the optimal path to gain a better level of resilience against faults.

This would help achieve a balance between mission success probability and selected trajectory characteristics, making sure that the trajectory design accounts for fault tolerance instead of this tolerance being only given by reconfigurations.

References

[1] Tománek, R. and Hospodka, J. “Reusable Launch Space Systems”. en. In: *MAD - Magazine of Aviation Development* 6.2 (2018), pp. 10–13. DOI: 10.14311/mad.2018.02.02.

[2] SpaceX. *Falcon User’s Guide*. Available online. Hawthorne, CA, USA, 2025. URL: https://www.spacex.com/assets/media/falcon-users-guide-2025-05-09.pdf.

[3] Wagner, E. “Research Flights on Blue Origin’s New Shepard”. In: *Gravitational and Space Research* 9 (2021), pp. 62–67. DOI: 10.2478/gsr-2021-0005.

[4] Blue Origin. *New Glenn Mission NG-2*. Accessed: 2025-12-12; details on the New Glenn orbital launch vehicle’s second mission and flight profile. 2025. URL: https://www.blueorigin.com/missions/ng-2.

[5] Illig, M., Ishimoto, S., and Dumont, E. “CALLISTO, a demonstrator for reusable launchers”. In: 2022.

[6] VILA, J. and HASSIN, J. “Technology acceleration process for the THEMIS low cost and reusable prototype”. en. In: (2019), 11 pages. DOI: 10.13009/EUCASS2019-97.

[7] Thies, C. “Investigation of the landing dynamics of a reusable launch vehicle and derivation of dimension loading for the landing leg”. In: *CEAS Space Journal* 14 (2022). DOI: 10.1007/s12567-022-00456-x.

[8] Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*. en. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. DOI: 10.1007/978-3-662-47943-8. URL: https://link.springer.com/10.1007/978-3-662-47943-8.

[9] Miramont, P. “Ariane 5 on board software: redundancy management”. en. In: *Proceedings of the 2nd Embedded Real Time Software Congress (ERTS’04)* (2004).

[10] Wagenblast, B. N. and Bettinger, R. A. “Statistical reliability estimation of space launch vehicles: 2000–2022”. en. In: *Journal of Space Safety Engineering* 11.4 (2024), pp. 573–589. DOI: 10.1016/j.jsse.2024.10.001.

[11] Fernández, L. A., Wiedemann, C., and Braun, V. “Analysis of Space Launch Vehicle Failures and Post-Mission Disposal Statistics”. en. In: *Aerotecnica Missili Spazio* 101.3 (2022), pp. 243–256. DOI: 10.1007/s42496-022-00118-5.

[12] Johansen, T. A. and Fossen, T. I. “Control allocation—A survey”. en. In: *Automatica* 49.5 (2013), pp. 1087–1103. DOI: 10.1016/j.automatica.2013.01.035.

[13] Orr, J. S. and Benson, B. *Multiple-Effector Control Allocation: Theory and Practice*. Greenbelt, MD, USA, 2022. URL: https://ntrs.nasa.gov/citations/20220018521.

[14] Zhang, Y. and Jiang, J. “Bibliographical review on reconfigurable fault-tolerant control systems”. In: *Annual Reviews in Control* 32.2 (2008), pp. 229–252. DOI: https://doi.org/10.1016/j.arcontrol.2008.03.008. URL: https://www.sciencedirect.com/science/article/pii/S1367578808000345.

[15] Zolghadri, A., Henry, D., Cieslak, J., Efimov, D., and Goupil, P. *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles: From Theory to Application*. en. Advances in Industrial Control. London: Springer London, 2014. DOI: 10.1007/978-1-4471-5313-9. URL: https://link.springer.com/10.1007/978-1-4471-5313-9.

[16] Chang-lin, X., Shu-ming, Y., Yu-qiang, C., and Li-jun, S. “Multiple model fault diagnosis and fault tolerant control for the launch vehicle’s attitude control system”. In: *The Aeronautical Journal* 128.1326 (2024), pp. 1875–1894. DOI: 10.1017/aer.2024.14.

[17] Paulino, N. et al. “Fault Tolerant Control for a Cluster of Rocket Engines -Methods and outcomes for guidance and control recovery strategies in launchers”. In: 2023. DOI: 10.5270/esa-gnc-icatt-2023-085.

[18] Roche Arroyos, C., Pascucci, M., Paulino, N., Arnedo, J., Navarro-Tapia, D., Marcos, A., Lalami, M., Alexandre, P., Simplício, P., and Casasco, M. “Fault-Tolerant Control for a Cluster of Rocket Engines - Results for launch and landing of a re-usable launcher”. en. In: (2022), 18 pages. DOI: 10.82124/CEAS-GNC-2024-045.

[19] Orr, J. S. and Slegers, N. J. “High-Efficiency Thrust Vector Control Allocation”. In: *Journal of Guidance, Control, and Dynamics* 37.2 (2014), pp. 374–382. DOI: 10.2514/1.61644. eprint: https://doi.org/10.2514/1.61644. URL: https://doi.org/10.2514/1.61644.

[20] Navarro-Tapia, D., Simplício, P., and Marcos, A. “Fault-Tolerant Dynamic Allocation Strategies for Launcher Systems”. en. In: *Aerospace* 12.5 (2025), p. 393. DOI: 10.3390/aerospace12050393.

[21] Simplicio, P., Marcos, A., and Bennani, S. “A Reusable Launcher Benchmark with Advanced Recovery Guidance”. English. In: 5th CEAS Conference on Guidance, Navigation & Control, EuroGNC19 ; Conference date: 03-04-2019 Through 05-04-2019. 2019. URL: https://eurognc19.polimi.it/.

[22] McNamara, S., Restrepo, C., Medina, E., Whitley, R., Proud, R., and Madsen, J. “Gain Scheduling for the Orion Launch Abort Vehicle Controller”. en. In: *AIAA Guidance, Navigation, and Control Conference*. Portland, Oregon: American Institute of Aeronautics and Astronautics, 2011. DOI: 10.2514/6.2011-6259. URL: https://arc.aiaa.org/doi/10.2514/6.2011-6259.

[23] Zagalo, I., Rosa, P., and Lemos, J. M. “Reinforcement Learning based adaptive control of landing manoeuvres in uncertain environments”. In: *Proceedings of the 2024 CEAS EuroGNC conference*. Bristol, UK, 2024. DOI: 10.82124/CEAS-GNC-2024-043.

[24] Cook, M. V. *Flight dynamics principles*. en. 2nd ed. Elsevier aerospace engineering series. Amsterdam Boston: Elsevier/Butterworth-Heinemann, 2007.

[25] Du, W. “Dynamic modeling and ascent flight control of Ares-I Crew Launch Vehicle”. en. Doctor of Philosophy. Ames: Iowa State University, Digital Repository, 2010, p. 2807776. DOI: 10.31274/etd-180810-1029. URL: https://lib.dr.iastate.edu/etd/11540/.

[26] Simplicio, P., Marcos, A., and Bennani, S. “A Reusable Launcher Benchmark with Advanced Recovery Guidance”. English. In: (2019). 5th CEAS Conference on Guidance, Navigation & Control, EuroGNC19 ; Conference date: 03-04-2019 Through 05-04-2019. URL: https://eurognc19.polimi.it/.

[27] Siouris, G. M. *Missile Guidance and Control Systems*. en. 1st ed. 2004. New York, NY: Imprint: Springer, 2004. DOI: 10.1007/b97614.

[28] Barrows, T. M. and Orr, J. S. *Dynamics and simulation of flexible rockets*. en. Amsterdam: Academic Press, 2021.

[29] Greensite, A. L. “Analysis and design of space vehicle flight control systems”. In: 1970. URL: https://api.semanticscholar.org/CorpusID:16781792.

[30] Macés-Hernández, J. A., Sagliano, M., Heidecker, A., Seelbinder, D., Schlotterer, M., Farì, S., Theil, S., Woicke, S., and Dumont, E. “Ascent Flight Control System for Reusable Launch Vehicles: Full Order and Structured H∞ Desings”. In: 2021.

[31] MathWorks. *Aerospace Toolbox User’s Guide*. https://www.mathworks.com/products/aerospace-toolbox.html. The MathWorks, Inc. 2025.

[32] Sagliano, M., Lu, P., Seelbinder, D., and Theil, S. “Analytical Treatise on Endo-Atmospheric Fuel-Optimal Rocket Landings”. In: *Journal of Guidance, Control, and Dynamics* 48.3 (2025), pp. 450–469. DOI: 10.2514/1.G008547. eprint: https://doi.org/10.2514/1.G008547. URL: https://doi.org/10.2514/1.G008547.

[33] Malyuta, D., Yu, Y., Elango, P., and Açıkmeşe, B. “Advances in trajectory optimization for space vehicle control”. en. In: *Annual Reviews in Control* 52 (2021), pp. 282–315. DOI: 10.1016/j.arcontrol.2021.04.013.

[34] Sagliano, M. “Pseudospectral Convex Optimization for Powered Descent and Landing”. en. In: *Journal of Guidance, Control, and Dynamics* 41.2 (2018), pp. 320–334. DOI: 10.2514/1.G002818.

[35] Kamien, M. and Schwartz, N. *Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management*. Dover books on mathematics. Dover Publications, 2012. URL: https://books.google.de/books?id=0IoGUn8wjDQC.

[36] Benson, D. “A Gauss Pseudospectral Transcription for Optimal Control”. en. In: ().

[37] Patterson, M. A. and Rao, A. V. *GPOPS-II: A General-Purpose MATLAB Software for Solving Multiple-Phase Optimal Control Problems*. http://www.gpops2.com. 2016.

[38] Jung, K.-W., Lee, S.-D., Jung, C.-G., and Lee, C.-H. “Model Predictive Guidance for Fuel-Optimal Landing of Reusable Launch Vehicles”. en. In: *International Journal of Control, Automation and Systems* 23.8 (2025). arXiv:2405.01264 [eess], pp. 2198–2218. DOI: 10.1007/s12555-024-0628-3.

[39] Ma, L., Wang, K., Xu, Z., Shao, Z., and Song, Z. “Trajectory optimization for powered descent and landing of reusable rockets with restartable engines”. In: 2018.

[40] Lu, P. and Davami, C. “Rethinking Propellant-Optimal Powered Descent Guidance”. en. In: *Journal of Guidance, Control, and Dynamics* 47.10 (2024), pp. 2016–2028. DOI: 10.2514/1.G008343.

[41] Lewis, F. *Optimal Control*. John Wiley & Sons Australia, Limited, 2005. URL: https://books.google.pt/books?id=FGGIPwAACAAJ.

[42] *Optimal and Robust Control: Advanced Topics with MATLAB*. en. 0th ed. CRC Press, 2012. DOI: 10.1201/b11660. URL: https://www.taylorfrancis.com/books/9781466503151.

[43] Kálmán, R. E. “Contributions to the Theory of Optimal Control”. In: 1960. URL: https://api.semanticscholar.org/CorpusID:845554.

[44] Barry, P., Zheng, A., Caillet, K., and Reynolds, D. “Development of an LQR Controller for a Jet Vanes Thrust-Vectored Rocket”. en. In: *2025 Regional Student Conferences*. Multiple Locations: American Institute of Aeronautics and Astronautics, 2025. DOI: 10.2514/6.2025-99074. URL: https://arc.aiaa.org/doi/10.2514/6.2025-99074.

[45] Bryson, A. E. “Applied Optimal Control: OPTIMIZATION, ESTIMATION, AND CONTROL”. en. In: ().

[46] University of Zielona Góra, I. o. C., Engineering, C., Witczak, M., and Pazera, M. “Fault Tolerant-Control: Solutions and Challenges”. en. In: *Pomiary Automatyka Robotyka* 20.1 (2016), pp. 5–16. DOI: 10.14313/PAR_219/5.

[47] Horn, R. A. and Johnson, C. R. *Matrix Analysis*. Cambridge University Press, 1985.

[48] Golub, G. and Van Loan, C. *Matrix Computations*. Johns Hopkins Studies in Atlantic History Culture. Johns Hopkins University Press, 1983. URL: https://books.google.de/books?id=g4gQAQAAIAAJ.

[49] Pei, J., Choueiri, M. N., Elandt, R., Peck, M. A., and Finch, P. “An Innovative Control Allocation Scheme to Address Reaction Thruster Interactions on a 3U CubeSat”. en. In: *2018 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, Florida: American Institute of Aeronautics and Astronautics, 2018. DOI: 10.2514/6.2018-2096. URL: https://arc.aiaa.org/doi/10.2514/6.2018-2096.

[50] Virnig, J. and Bodden, D. “Multivariable control allocation and control law conditioning when control effectors limit”. In: *Guidance, Navigation, and Control Conference*. DOI: 10.2514/6.1994-3609. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.1994-3609. URL: https://arc.aiaa.org/doi/abs/10.2514/6.1994-3609.

[51] Shi, J., Zhang, W., Li, G., and Liu, X. “Research on allocation efficiency of the redistributed pseudo inverse algorithm”. en. In: *Science China Information Sciences* 53.2 (2010), pp. 271–277. DOI: 10.1007/s11432-010-0032-x.

[52] Bodson, M. “Evaluation of Optimization Methods for Control Allocation”. In: *Journal of Guidance, Control, and Dynamics* 25.4 (2002), pp. 703–711. DOI: 10.2514/2.4937. eprint: https://doi.org/10.2514/2.4937. URL: https://doi.org/10.2514/2.4937.

[53] MURATA, R., THIOULOUSE, L., MARZAT, J., PIET-LAHANIER, H., GALEOTTA, M., and FARAGO, F. “Optimal reconfigurable allocation of a multi-engine cluster for a reusable launch vehicle”. en. In: (2022), 12 pages. DOI: 10.13009/EUCASS2022-4382.

[54] Miller, L. J. and Pei, J. “A Comparison of Control Allocation Methods in the Presence of Parametric Model Uncertainty”. en. In: *AIAA SCITECH 2025 Forum*. Orlando, FL: American Institute of Aeronautics and Astronautics, 2025. DOI: 10.2514/6.2025-2240. URL: https://arc.aiaa.org/doi/10.2514/6.2025-2240.

[55] Grant, M. and Boyd, S. *CVX: MATLAB Software for Disciplined Convex Programming*. http://cvxr.com/cvx. CVX Research, Inc. Los Angeles, CA, 2024.

[56] MathWorks. *Optimization Toolbox User’s Guide*. https://www.mathworks.com/products/optimization.html. The MathWorks, Inc. 2025.

[57] The MathWorks, I. *Exact Linearization Algorithm — Simulink Control Design Documentation*. https://www.mathworks.com/help/slcontrol/ug/exact-linearization-algorithm.html. [Online; accessed 17-Dec-2025]. 2025.