



Article

A Decreasing Horizon Model Predictive Control for Landing Reusable Launch Vehicles

Guillermo Zaragoza Prous ^{*}, Enric Grustan-Gutierrez  and Leonard Felicetti 

Faculty of Engineering and Applied Sciences, Cranfield University, Bedford MK43 0AL, UK; e.grustan@cranfield.ac.uk (E.G.-G.); leonard.felicetti@cranfield.ac.uk (L.F.)

* Correspondence: g.zaragozaprous@cranfield.ac.uk

Abstract: A novel approach to model predictive control (MPC) with a decreasing horizon is analysed for guiding and controlling reusable launch vehicles (RLVs) during powered descent phases. Conventional MPC methods typically use receding horizons, where optimal control inputs are computed over fixed time intervals. However, when applied directly, these methods can cause a hovering-like behaviour, preventing the vehicle from reaching the landing platform, as the landing time is continually deferred at each iteration. The proposed solution addresses this problem by adjusting the prediction horizon dynamically, reducing its length over time. This dynamic adjustment is driven by a time-scaling factor and the time elapsed since the previous MPC iteration. Optimal control solutions are derived through convex optimization techniques. To evaluate the algorithm's robustness against initial conditions, a Monte Carlo analysis is performed by varying initial position, velocity and mass. This method can also be used as a viable methodology for selecting tuning parameters for the MPC to ensure a successful and safe landing for a wide range of initial conditions.

Keywords: powered descend guidance; convex optimization; model predictive control; reusable launch vehicle



Academic Editors: Pedro Simplicio, Paul Acquatella and Diego Navarro-Tapia

Received: 1 December 2024

Revised: 26 January 2025

Accepted: 27 January 2025

Published: 31 January 2025

Citation: Zaragoza Prous, G.; Grustan-Gutierrez, E.; Felicetti, L. A Decreasing Horizon Model Predictive Control for Landing Reusable Launch Vehicles. *Aerospace* **2025**, *12*, 111. <https://doi.org/10.3390/aerospace12020111>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development of technologies that allow for vertical descent and landing of rockets has proven to be a game changer by reducing the cost barrier to space access. Thanks to this technology, new companies, e.g., SpaceX, can offer competitive prices for launch services to new space actors and enterprises [1]. Likewise, they enable planetary missions to touch down on previously selected landing sites deemed interesting for their scientific value. Examples of this are the Mars Science Laboratory mission [2], the MARS 2020 mission, and the Tianwen-1 Mars mission [3].

Guidance and control during the powered descent and landing of reusable launchers is one of the most complex challenges that characterise such systems, involving innovative solutions from a technical and practical perspective. The challenge of achieving a soft landing for a fast free-falling launcher lies in the precise control of its dynamics. This involves a complex interplay of forces, where the thrust must be modulated and re-oriented to counteract the effects of gravity and atmospheric drag [1]. By adjusting the thrust vector, the descent can be stabilised, allowing for adjustments to be made in real-time to ensure a gentle touchdown. This process requires advanced algorithms and control systems capable of responding to the rapidly changing conditions of a free fall, ensuring that the launcher can navigate to the desired landing site and achieve a controlled and safe landing [4].

The development of powered descent guidance algorithms based on optimal control theory began in the 1960s during the Apollo era as part of the lunar landing program [5–7]. More recently, the space sector has renewed its interest in this type of technology as part of the Mars exploration missions at the end of the last century and beginning of the 2000s through the utilisation of constrained optimisation methods. Such algorithms were further optimised by the application of convex programming [8,9], which demonstrated that a second-order cone programming (SOCP) set of equations can be used to solve the powered descent problem through interior point methods. Other guidance methods have been successfully applied to perform smooth landings on planetary bodies. Among other examples are the Mars Science Laboratory Entry, Descent, and Landing System, which followed a 3D polynomial trajectory computed at the beginning of the landing phase and performed a powered approach to the Martian surface [2] and the polynomial guidance strategy used by the Tianwen-1 mission to perform a diversion manoeuvre during the powered descent phase [3]. Nevertheless, whereas polynomial powered descent guidance is computationally efficient and sufficient for small diversions, constrained optimisation methods solve for globally fuel-optimal and constrained diversions that allow for the inclusion of the particularities of the spacecraft in the guidance strategy [9,10]. For the particular case of landing a reusable launch vehicle (RLV), polynomial guidance might not be suitable due to rapid changes in the flight conditions, as mentioned before.

The last decade has seen a significant surge in the development of advanced guidance and control systems for launch vehicles. The incorporation of convex optimisation models has revolutionised the precision with which these vehicles can be manoeuvred, accounting for both translational and rotational dynamics. This progress in six degrees-of-freedom (6 DoF) modelling enhances the accuracy and reliability of launchers, marking a substantial advancement in aerospace engineering and space exploration capabilities. In [11], the methodology of successive convexification, a form of sequential convex programming (SCP) was used for solving the powered descent guidance problem. Within that study, the algorithm employed virtual control and trust region modifications to facilitate convergence of the problem. In [12], a similar approach was used by integrating aerodynamic controls within a two-dimensional framework, simulating conditions over a flat Earth model. Additionally, in [13], the control optimiser leveraged the atmospheric drag energy to enhance efficiency, representing another innovative application of this technique. These advancements underscore the growing reliance on convex models to optimise the performance and safety of reusable launch vehicles and landers during the final phases of descent and landing. A more detailed survey on optimization-based space vehicle control was performed in [14], not only for a powered landing scenario but also for other guidance and control challenges such as rendezvous and proximity operations, constrained reorientation, and orbit transfer and injection. In [15], an assessment of the coupling between the guidance and control techniques was carried out for different RLV recovery strategies and controllability challenges. Additionally, alternative guidance strategies are currently being studied such as evolutionary optimisation, tree searches, and machine learning [16,17]. For instance, a trained neural network was used in [18] to solve fuel-optimal powered descent guidance in lunar pinpoint landing, and closed-loop MPC guidance with deep neural networks was used for landing a rocket's first stage [19].

Nevertheless, the challenges of Earth descent and landing for RLVs are indeed complex, with atmospheric drag, winds, and structural flexibility posing significant risks to the integrity of the manoeuvres. Open-loop control schemes, which rely solely on preplanned trajectories, are often insufficient due to their inability to adapt to dynamic environmental conditions. In contrast, closed-loop control systems, such as model predictive control (MPC), offer a more robust solution by continuously adjusting the vehicle's

trajectory in response to real-time sensor data. MPC can predict future states of the RLV and make necessary adjustments to ensure a safe and accurate landing [20,21]. MPC had been widely used for managing the complex processes of industries like chemical and oil processing [22,23], where changes occur over longer periods. However, applying MPC to RLVs during powered descent involves rapid dynamics and requires real-time computational speed, robustness against disturbances, and precise handling of constraints. Typically, MPC uses a receding horizon strategy where the optimal control is always calculated for a fixed period of time shifting at each iteration step [22]. However, the direct implementation of such schemes in a powered descent scenario is not trivial, and it can lead to unexpected behaviours such as hovering-like trajectories with the vehicle never reaching the landing target. This is shown in [24], where an optimal strategy for setting the prediction horizon duration was developed as a solution to counter the impossibility of finding a fixed receding horizon for the guidance and control of RLVs. In addition, Scharf et al., in [25,26], used an onboard time-of-flight search algorithm to obtain the best prediction horizon for the optimisation problem at each iteration of the G-FOLD algorithm.

This paper introduces a new algorithm designed to simplify the time-length update phase within an MPC algorithm. It details an iterative method that adjusts the horizon length using a specific update factor. The resulting solution, while sub-optimal, might demand significantly less computational power than previously optimised algorithms, such as those mentioned in [24], as it predefines the prediction horizon to use at each iteration instead of including it as a variable to optimize in the cost function. Other earlier methods aimed to determine the optimal horizon at the beginning of each MPC iteration [26] require multiple problem-solving instances per iteration and thus are not computationally efficient. The proposed method markedly reduces the iterations needed to establish a viable time horizon, thereby reducing the computational time. This feature is particularly advantageous for real-time embedded system applications. This paper completes the work presented in [27], varying the initial conditions of the numerical simulations and completing the study with a Monte Carlo analysis of the robustness of the algorithm with respect to the variability of the initial state vector, which benefits the proposed method to accurately tune the MPC algorithm and prevents possible issues during flight.

The paper is organised into the following sections: Section 2 defines the set of equations and constraints of the guidance problem, and a convexification of this set of equations is proposed in Section 3 so that an MPC algorithm can be developed later in Section 3.2. Next, the simulation results and a Monte Carlo analysis in Section 4 prove the robustness of the algorithm against different initial and environmental conditions. Finally, Section 5 provides the concluding remarks and future work.

2. Problem Statement

In this section, the rocket plant used for the simulation is presented along with a mathematical definition of the landing problem. This section has been written in an abridged format, as the main theoretical derivation was presented in [27].

2.1. Model

The model used for the simulations is a point-mass two-dimensional model and is subjected to the forces generated by the thrust T of its engines and atmospheric drag D (see Figure 1):

$$\begin{cases} \dot{x} = V_x \\ \dot{z} = V_z \\ \dot{V}_x = \frac{T_x - D \cos(\gamma)}{m} \\ \dot{V}_z = \frac{T_z - D \sin(\gamma)}{m} - g \\ \dot{m} = -\frac{|T|}{I_{sp} g_0} \end{cases} \quad (1)$$

where $\gamma = \tan^{-1}(\frac{V_z}{V_x})$ is the ascent/descent angle of the RLV, I_{sp} is the specific impulse of its engines, and g_0 is the gravity force at sea level. The orientation of the RLV is not explicitly represented in the set of Equation (1), although the tangent of the thrust vector is considered to be the orientation of the RLV, $\theta = \tan^{-1}(\frac{T_z}{T_x})$, fixing the thrusters to the body.

$$D = \frac{1}{2} \rho(z) \cdot V^2 \cdot S(\alpha) \cdot C_D(\alpha) \quad (2)$$

$$\alpha = |\theta - \gamma| \quad (3)$$

The aerodynamic drag D is obtained from Equation (2), where ρ is the air density at a given altitude, V is the magnitude of the velocity relative to the atmosphere, which includes the RLV velocity and the wind velocity, S is the area of the launcher exposed to atmospheric flow, and C_D is the drag coefficient. The angle of attack α is defined as the difference between θ and γ (see Equation (3)). This model is further developed in [27].

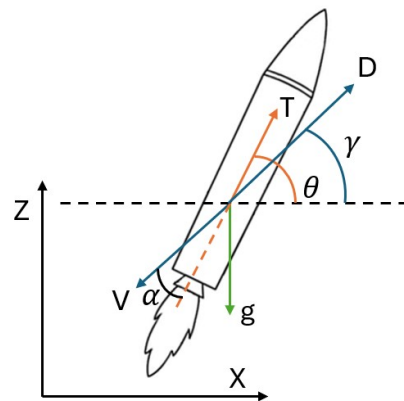


Figure 1. RLV model variables as presented in Equation (1).

2.2. Definition of Landing Problem

A set of constraints defines the problem (a two-boundary value problem (TBVP)) to be solved, including the initial and desired final values of the state vector:

$$\begin{cases} x(t_0) = x_0 \\ z(t_0) = z_0 \\ V_x(t_0) = V_{x_0} \\ V_z(t_0) = V_{z_0} \end{cases} \quad (4)$$

$$\begin{cases} x(t_f) = 0 \\ z(t_f) = 0 \\ V_x(t_f) = 0 \\ V_z(t_f) = 0 \end{cases} \quad (5)$$

where t_0 and t_f are the initial and final time. The landing pad is assumed to be fixed in the origin of the reference frame (Constraint (5)) and needs to be reached by the RLV with zero velocity, in order to have a soft landing.

The TBVP assumes a cost function that minimises the required thrust:

$$J = \int_0^{t_f} |T| dt \quad (6)$$

The list of considered constraints is derived in [8,27] and contains the following equations:

$$0 < T_{min} \leq T(t) \leq T_{max} \quad (7)$$

$$z(t) \geq 0 \quad (8)$$

$$T_z(t) \geq 0 \quad (9)$$

$$m(t_f) \geq m_{dry} \quad (10)$$

$$\left| \frac{\pi}{2} - \theta_B(t) \right| \geq \left| \frac{\pi}{2} - \theta(t) \right| \quad (11)$$

where $\theta_B(t)$ is an orientation boundary that changes over time.

3. Model Predictive Control

The MPC strategy used is illustrated in Figure 2, where the RLV's physical model (Equation (1)) is controlled by an optimal thrust profile derived from a convex optimisation solver (SeDuMi [28]). The solver employs the convexified set of constraints and equations of motion outlined in Section 3.1. The implementation of the decreasing horizon model predictive algorithm is then detailed in Section 3.2.

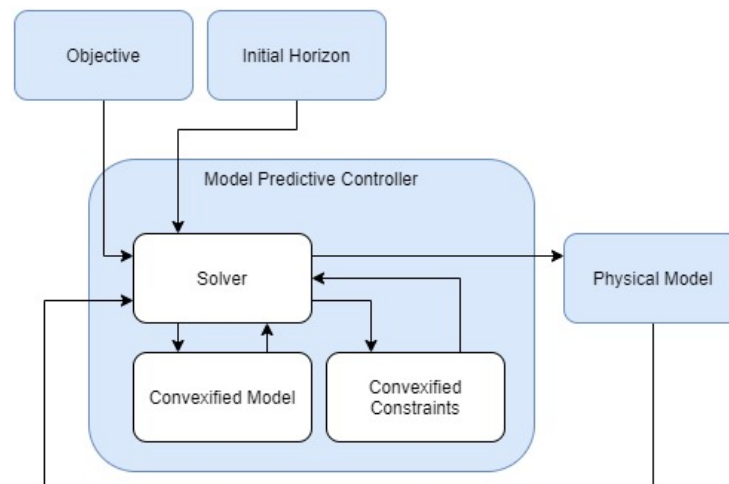


Figure 2. Flowchart of MPC Strategy [27].

3.1. Convexification of the Problem

Two steps must be performed to turn the landing problem into a semidefinite programming (SDP) problem [8], which can be solved by a convex solver. The first step is to transform the non-convex constraints into a convex form following the process described in [8]. In the landing problem presented in Section 2.2, the sources of non-convexity are (1) the change of mass in the last row of Equation (1), which is tackled by a change of variable as in Equation (12), and (2) the thrust limit in Constraint (7), which is dealt with by introducing a slack variable Γ as in Equation (13) [8].

$$\eta = \ln(m) \quad (12)$$

$$\|\vec{T}\| \leq \Gamma \quad (13)$$

After a series of steps followed in [27], the last equation of Equation (1) can be rewritten as Equation (14).

$$\dot{\eta} = \frac{\dot{m}}{m} = -\frac{\Gamma}{I_{sp} g_0 m} \quad (14)$$

The objective index defined in Equation (6) can now be reformulated with the new Γ variable as Equation (15).

$$J = \int_0^{t_f} \frac{\Gamma}{m} dt \quad (15)$$

Thus, the state variables can be updated at each iteration step of the prediction phase in the MPC by using the linearised set of equations of motion in Equation (16), where the atmospheric drag is not included and therefore it is considered as an external disturbing factor for the controller.

$$\begin{cases} x(t + \Delta t) = \frac{1}{2}u_x(t) \Delta t^2 + V_x(t) \Delta t + x(t) \\ z(t + \Delta t) = \frac{1}{2}(u_z(t) - g) \Delta t^2 + V_z(t) \Delta t + z(t) \\ V_x(t + \Delta t) = u_x(t) \Delta t + V_x(t) \\ V_z(t + \Delta t) = (u_z(t) - g) \Delta t + V_z(t) \\ \eta(t + \Delta t) = -\frac{\Gamma(t)}{I_{sp} g_0 m(t)} \Delta t + \eta(t) \end{cases} \quad (16)$$

3.2. Decreasing Horizon MPC

The proposed modification to the MPC method is based on the adjustment of the horizon length to solve the issues when a receding horizon is applied. The main issue with the receding horizon strategy for the TBVP is that the landing time is always postponed, and therefore the guidance algorithm will generate the guidance law to land in the future until the problem becomes infeasible due to the lack of propellant (see [27] for a comparison between both strategies) and to allow the RLV to land. For this purpose, it is necessary to define several time-related parameters, which will define how and when the prediction horizon is updated:

- (a) *Terminal Horizon time (TH)*: which serves for setting the global desired time to be landed. This represents the upper-bound limit for touch down, that in any case can happen anytime before.
- (b) *Prediction Horizon time (PH)*: time used in the prediction step by the optimisation solver.
- (c) Δt_I : time between each MPC iteration when the optimiser recalculates the optimal solution by getting the updated state of the RLV.
- (d) Δt_P : time between each of the prediction states computed within the same prediction horizon.
- (e) *Update Factor (UF)*: determines when and by which rate *TH* and *PH* are decreased. Its value must be set between 0 and 1.

Algorithm 1 and Figure 3 outline the operation of this new strategy. In the first step of the algorithm, the initial values for the new parameters are set. Two nested loops are then executed. The outer loop (line 2) updates *TH* and *PH* at each iteration and continues until the prediction horizon *PH* becomes smaller than the iteration time step Δt_I . The inner loop (line 3) handles the prediction and calculates the optimized control for the RLV. The prediction step is carried out by the function *convexOptimiser(X, PH, Δt_P)*, which uses the given *PH* and Δt_P , solving the optimization problem in Equation (15) and implementing the equations and constraints described in Section 3.1. Once the outer loop completes, the algorithm applies the final segment of the control signal *U* to the RLV to complete the landing manoeuvre (line 10).

Algorithm 1 Decreasing Horizon MPC Algorithm

```

1: Define  $TH, PH, \Delta t_I, \Delta t_p, UF$  and  $X_0$ 
2: while  $PH > \Delta t_I$  do
3:   for  $iteration = 0, 1, 2, \dots, TH \cdot UF$  do
4:      $U = \text{convexOptimiser}(X, PH, \Delta t_p)$ 
5:     Apply  $U$  for  $\Delta t_I$ 
6:   end for
7:   Update  $TH = TH \cdot (1 - UF)$ 
8:   Update  $PH = PH \cdot (1 - UF)$ 
9: end while
10: Apply the rest of  $U$ 

```

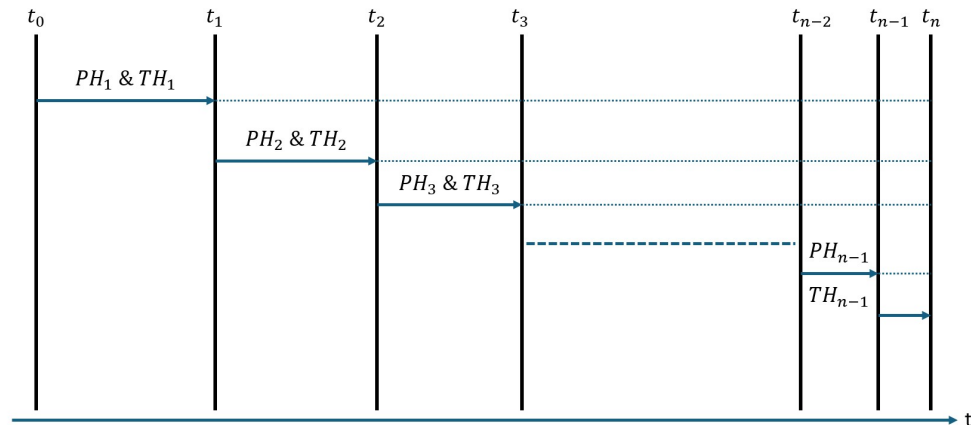


Figure 3. Generalisation of the decreasing MPC algorithm.

Figure 3 illustrates how the update of TH and PH progresses over time within the algorithm. Starting with the initial values $TH_1, PH_1, \Delta t_I, \Delta t_p$ and UF , the horizons TH_1 and PH_1 are applied over a time t_1 equal to $TH \cdot UF$. At this point, TH_2 and PH_2 are calculated by multiplying TH_1 and PH_1 by $(1 - UF)$, as shown in lines 7 and 8 of Algorithm 1. The algorithm then proceeds with these updated horizons for the next iteration. It is important to note that only the terminal and prediction horizons are adjusted during the execution of the MPC algorithm. The process continues until the prediction horizon PH becomes shorter than the iteration step Δt_I , represented as PH_{n-1} in Figure 3. At this point, the RLV uses the remaining portion of the predicted control generated by the MPC. As discussed in Section 4, the selection of horizon times and initial parameters is critical to the algorithm's performance, and these values vary depending on the problem's initial conditions.

4. Numerical Results

The performance of the proposed MPC algorithm for reusable launch vehicle (RLV) scenarios is evaluated through numerical simulations. This section presents several test cases simulating landings on Earth using the first stage of SpaceX's Falcon 9 rocket [29]. The methodology follows the approach outlined in [27], with the addition of a Monte Carlo analysis to assess the robustness of the MPC algorithm with respect to varying initial conditions of the RLV. The aim is to further validate the capability of the proposed algorithm in achieving a successful RLV landing in a realistic scenario, complementing the results presented in [27]. The chosen scenario models the landing of the Falcon 9's first stage, beginning from an initial position of $x_0 = 2500$ m, $z_0 = 5000$ m and initial velocities of $V_{x_0} = -170$ m/s and $V_{z_0} = -250$ m/s. These values are used as the initial state vector for solving the problem.

The relevant parameters for the rocket and the environment are provided in Tables 1 and 2, respectively. For the simulations, only one engine of the rocket is con-

sidered. The initial mass of the RLV in each scenario is determined by the remaining fuel mass at the start of the landing phase. The simulations are conducted using the lower atmosphere of Earth as the environmental model.

Table 1. Falcon 9 first stage parameters used in the simulations.

Rocket	Falcon 9 1st Stage
N. Engines	9
Engine Thrust [kN]	845
Specific Impulse [s]	311
Propellant Mass ¹ [t]	395.7
Dry Mass [t]	25.6
Length [m]	41.2
Diameter [m]	3.7
Maximum Thrust [%]	0.8
Minimum Thrust [%]	0.1

¹ Initial propellant mass at launch.

Table 2. Earth parameters used in the simulations. Data from [30].

Environment	Earth
Air density at sea level [kg/m ³]	1.217
Scale Height [m]	85,000
Gravity acceleration at sea level [m/s ²]	9.81
Speed of Sound [m/s]	340

4.1. Understanding the Feasibility Region of the Optimisation Problem

Before analysing the performance of the proposed MPC algorithm, a study is carried out to understand the suitability of the convex optimiser to solve the problem. This is achieved through a series of Monte Carlo and parametric analyses that allow a better comprehension of the feasible region with respect to the prediction horizon PH and the initial conditions. A sensitivity analysis regarding the initial conditions and prediction horizon duration was performed in Section 4.1.1 of [27], with a range of initial conditions that still apply for those used for the simulations in this study (see Table 3).

Table 3. Monte Carlo minimum and maximum boundaries for the initial conditions of the problem [27].

Variable	Min. Value	Max. Value
Position X [m]	2000	3500
Position Z [m]	4000	6000
Velocity V_x [m/s]	−250	−50
Velocity V_z [m/s]	−350	−200
Fuel percentage [%]	1.5	3.5
Mass m [t]	31.5	39.5
Orientation θ_B [deg]	45	45

4.1.1. Finding the Applicable Ranges of the Prediction Horizon for Different Initial Velocities

The second step of the feasibility study is an analysis of the effect of the initial magnitude in the two components of the velocity V_x and V_z [27]. A parametric analysis is executed to understand the feasible range of PH values for each V considered. Table 4 describes the initial conditions of the analysis while the velocities are obtained from Table 3.

Table 4. Initial conditions for the parametric analysis of the prediction horizon of the problem.

Variable	Initial Value
Position X [m]	2500
Position Z [m]	5000
Fuel percentage [%]	2.8
Mass m [t]	36.7
Orientation θ_B [deg]	45

The parametric analysis results can be seen in Figure 4. In Figure 4a, the values for the problem show a global minimum of 32 s around $V = (-150, -250)$ m/s and a radius of 30 to 40 m/s. The propagation of these initial velocities starting at the initial position (2500, 5000) m allows for a clear understanding of the physical interpretation of the location of such minimum. These velocity values bring the RLV close to the landing site (0, 0) m without the need for thrust, and therefore they allow for minimisation of the eventual corrections of the trajectory.

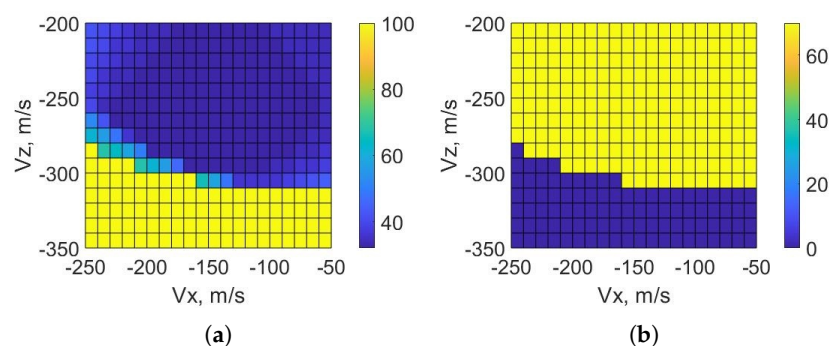


Figure 4. Parametric Analysis of the initial velocity for the optimisation problem: (a) minimum PH and (b) maximum PH in seconds. The values in yellow in (a) and dark blue in (b) define the infeasible region of the problem.

In Figure 4b, the region with the maximum feasible PH values is in the upper-right corner. Such a region of maximum feasible PH is due to the lower thrust needed to overcome the initial velocities, being these with smaller magnitudes compared to the other cases, and therefore a higher quantity of fuel still available to descend slowly or even hover before touching the ground. Moreover, the values in yellow in Figure 4a and in dark blue in Figure 4b when V_z is close to -350 m/s represent the cases in which the problem is unfeasible.

4.2. Analysis of the Performance of the Proposed MPC

After understanding the applicability of the presented optimiser for the landing scenario, the performance of the proposed MPC algorithm is analysed. Firstly, a parametric analysis is carried out to understand where the feasible region is with respect to the parameters of the MPC to be tuned. These parameters are the prediction horizon PH , the terminal horizon TH , and the update factor UF . Afterwards, an optimal trajectory obtained within the feasible region is discussed.

4.2.1. Finding the Applicable Ranges of Terminal Horizon and Update Factor

The feasible ranges for TH and UF need to be identified based on the findings in Section 4.1. To this end, a parametric test with 1188 cases was conducted across a broad range of values, see Table 5. It was determined that a $TH \approx 2 \cdot PH$ and a UF within the range $[0.01, 0.20]$ are suitable for solving the problem. The feasibility of the problem is assessed by verifying that the constraints outlined in Section 2.2 are satisfied and by

comparing the state of the RLV at the end of each MPC run with the final objective, which is $[0 \text{ m}, 0 \text{ m}, 0 \text{ m/s}, 0 \text{ m/s}]$. The problem is considered solved if the position and velocity errors fall within specified margins: 2 m and 2 m/s, respectively.

Table 5. Range of values for the TH , UF , and PH for the parametric analysis of the TH and UF based on the result obtained in Section 4.1.

MPC Tuning			
Variable	Min. Value	Max. Value	Step Size
PH [s]	26	42	2
TH [s]	58	78	2
UF [-]	0.01	0.02	0.01
UF [-]	0.04	0.40	0.04

The execution of the set of scenarios revealed a zone in which the problem is feasible, as seen in Figure 5. From the set containing 1188 cases, 348 were feasible. The first detail seen in Figure 5 is that $PH = 28 \text{ s}$ is the minimum feasible PH . This is explained by the fact that the MPC algorithm will not solve the problem if the convex optimiser does not. For $PH = 28 \text{ s}$, the problem is feasible for almost any TH selected and a UF greater or equal to 0.12. As the PH increases, the feasible region is restricted to a lower UF , i.e., between 0.04 and 0.20 for $PH > 28 \text{ s}$ and $PH \leq 32 \text{ s}$. The higher boundary of UF continues decreasing until there is only $UF = 0.04$ for $PH = 42 \text{ s}$. In a similar manner, as the UF increases, the TH increases. For the cases with lower PH , there are feasible solutions for all the values of TH with various values of UF , while in the upper part of Figure 5 the main feasible region is constrained to $UF = 0.04$ and $UF = 0.08$ with $TH \geq 74 \text{ s}$. The scenarios comprised by $UF = 0.12$ are a good example of the tendency of the feasible region with respect to the selected value of TH . Initially, it covers the full spectrum of selected values of TH when $PH \leq 30 \text{ s}$ and ends with only one feasible solution with $PH = 42 \text{ s}$ and $TH = 78 \text{ s}$. There were other feasible cases seen in Figure 5, but as they were isolated, they were not considered to be part of the main feasibility region.

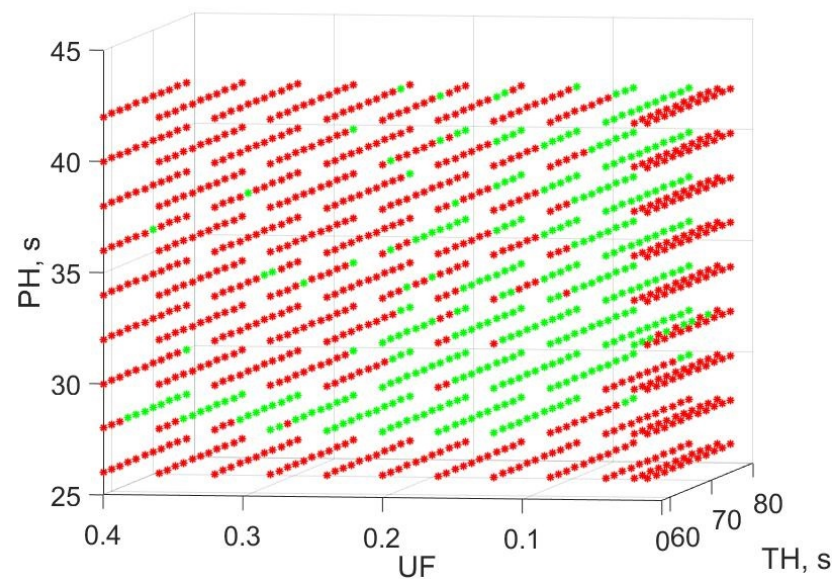


Figure 5. Feasible (green) and unfeasible (red) cases of the parametric analysis of PH , TH , and UF .

4.2.2. Optimal Trajectory

The final step to discuss is the performance of the MPC algorithm in a feasible scenario. The selected case, part of the set analysed in Section 4.2.1, considers parameters of $PH = 32$ s, $TH = 64$ s, and $UF = 0.08$.

Figure 6 illustrates the trajectory followed by the RLV during the powered descent. The red vectors represent the thrust applied at each moment, with the largest thrust exerted at the start and end of the landing. This is further confirmed by Figure 7, which displays the state and control vectors for the landing case. In Figure 7e, the thrust follows a bang–bang profile, reaching its maximum for extended periods at both the beginning and end of the simulation. Meanwhile, Figure 7f shows that the thrust angle starts at 65 degrees and gradually converges towards 90 degrees for a vertical landing.

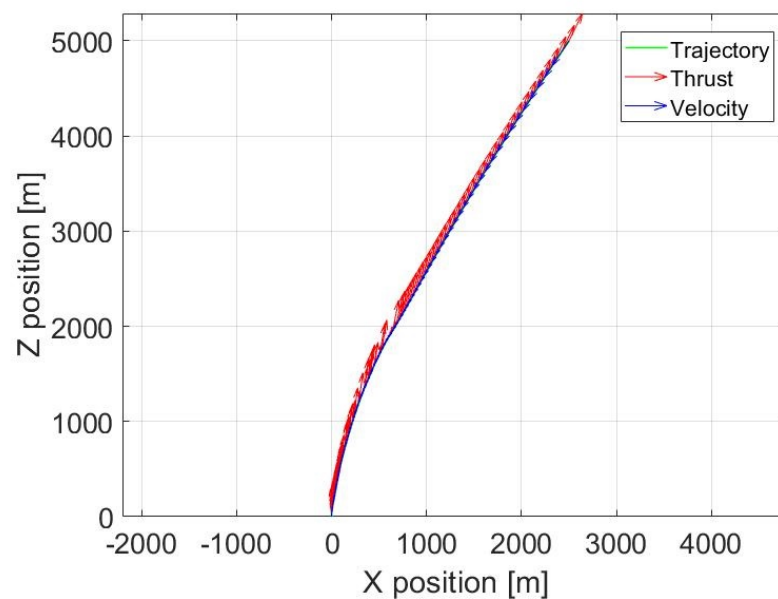


Figure 6. MPC: Landing trajectory with thrust vector.

In Figures 6 and 7a, the altitude drops rapidly during the first 3000 m, while the proposed MPC algorithm simultaneously reduces the V_z , as shown in Figure 7b, counteracting the initial velocity. Notably, between $t \approx 20$ s and $t \approx 42$ s, the V_z remains nearly constant to allow the RLV to approach the target on the X-axis, as seen in Figure 7a–d, resulting in a slower decrease in altitude. During the final phase of the descent, the MPC reduces the magnitudes of both the V_x and V_z components, achieving a safe landing at the target position. The final state vector of the RLV is $[-0.001$ m, 1.718 m, -0.001 m/s, -1.925 m/s], which satisfies the specified margins. The final mass is 28183 kg, greater than the dry mass of the first stage of the Falcon 9 rocket.

Although not shown in Figure 7f, some cases exhibit instantaneous changes in the thrust angle θ due to the absence of a constraint on the initial control actions. As a result, at each step of the MPC algorithm, the initial θ may differ from its previous value. Future research activities will focus on including this constraint in the algorithm to ensure that the optimizer takes the RLV's current orientation into account. Additionally, the model in Equation (1) does not consider the rotational dynamics of the RLV, meaning that the computed θ is assumed to be achieved instantaneously. Nonetheless, these instantaneous jumps in θ are limited to ± 10 degrees. It is reasonable to expect that if attitude dynamics were included in the simulation model, the angle would be reached over a shorter period than the characteristic time of the simulation.

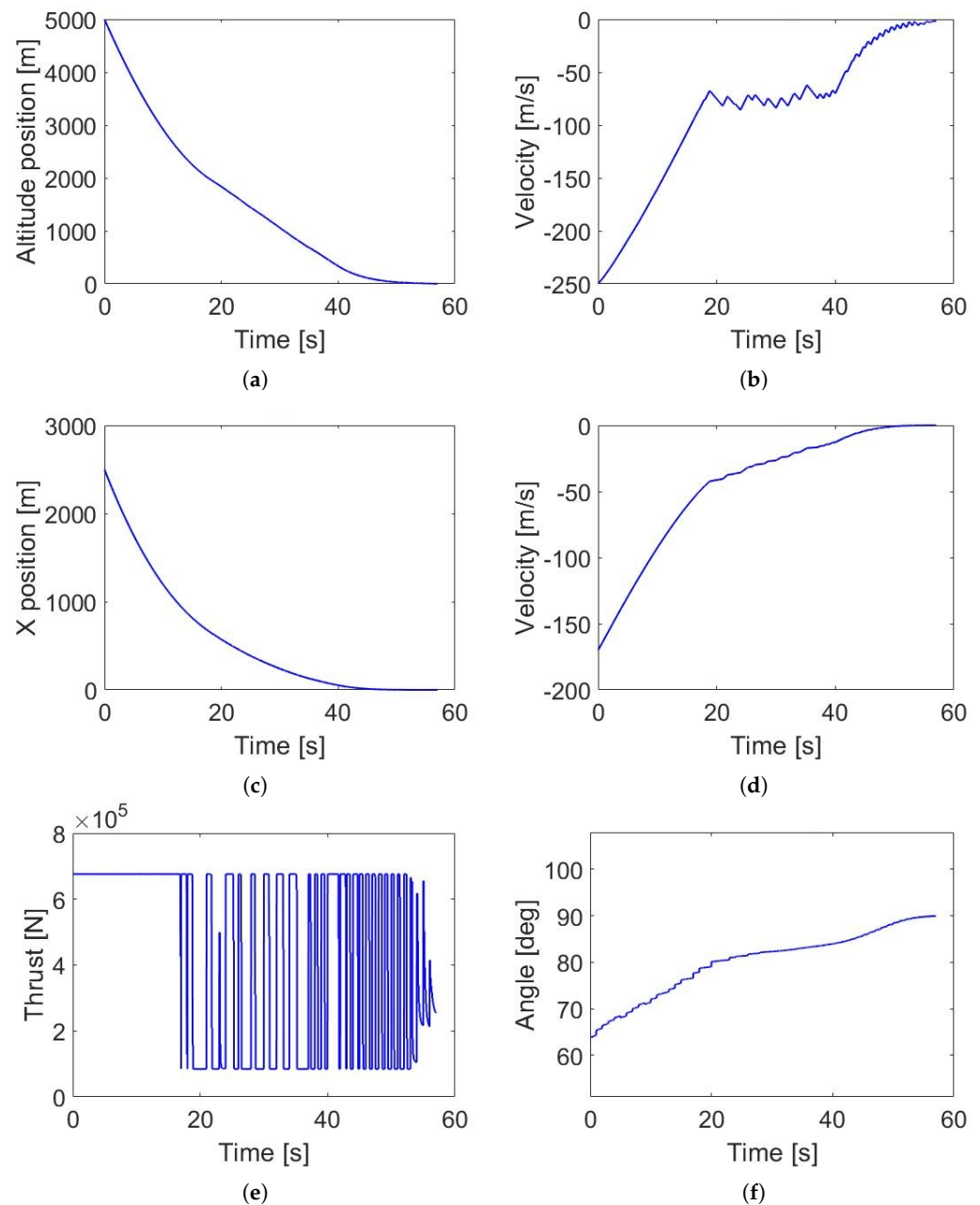


Figure 7. MPC: Nominal scenario landing variables for the RLV. Subfigure (a) shows its altitude against time, (b) shows its V_z over time, (c) shows its X position against time, (d) shows its V_x over time, (e) shows the used thrust norm, and (f) shows the thrust angle γ .

4.3. Monte Carlo Analysis of the Nominal Scenario Boundaries for the Selected PH, TH and UF

Once a nominal scenario has been studied and the values of the PH, TH, and UF have been selected, the last step is to understand how robust the algorithm is with respect to uncertainties in the initial state of the vehicle. Using the chosen values of the PH, TH, and UF, a Monte Carlo analysis is performed to find the boundaries of the feasibility region in each of the elements of the state vector.

Table 6 shows the minimum and maximum values for the uniformly distributed random points characterising the analysis. The range of values considered in each component of the position is 1000 m, from 2000 m in X and 4500 m in Z, up to 3000 m and 5500 m, respectively. The velocity components are constrained to $[-215, -135]$ m/s in V_x and $[-300, -200]$ m/s in V_z , which means a range of 90 m/s and 100 m/s. The initial mass

studied covers from a minimum of 33.5 tonnes up to 39.8 tonnes, which means that the percentage of the fuel goes from 0.02 to 0.036.

Table 6. Boundaries of the Monte Carlo analysis of the selected *PH*, *TH*, and *UF*.

Variable	Minimum	Maximum
<i>X</i> [m]	2000	3000
<i>Z</i> [m]	4500	5500
<i>V_x</i> [m/s]	−215	−135
<i>V_z</i> [m/s]	−300	−200
Mass [kg]	33,500	39,845
Fuel Mass percentage [%]	0.02	0.036

4.3.1. Monte Carlo Analysis with Fixed Initial Velocity

The first case studied is the sensibility of the proposed MPC algorithm to variations of the initial position and mass. Figure 8 presents the results of the 500 simulations. Figure 8a compares the variability of the problem with respect to the initial position. It displays a feasible region for all $X \leq 2700$ m for all the range of altitudes considered, but going up to $X = 3000$ m when the altitude is higher. Additionally, it presents an unfeasible region in the lower-right corner when the initial horizontal distance is greater and the altitude is lower, which shows the relation between initial position and velocity. As the velocity is fixed, there is a boundary in which the problem becomes unfeasible for the MPC algorithm to solve. This relates to what was stated previously in Section 4.1.1. The unfeasible cases in the feasible region are caused by an initial mass too small or too big, as seen in Figure 8b when $X \leq 2500$ m.

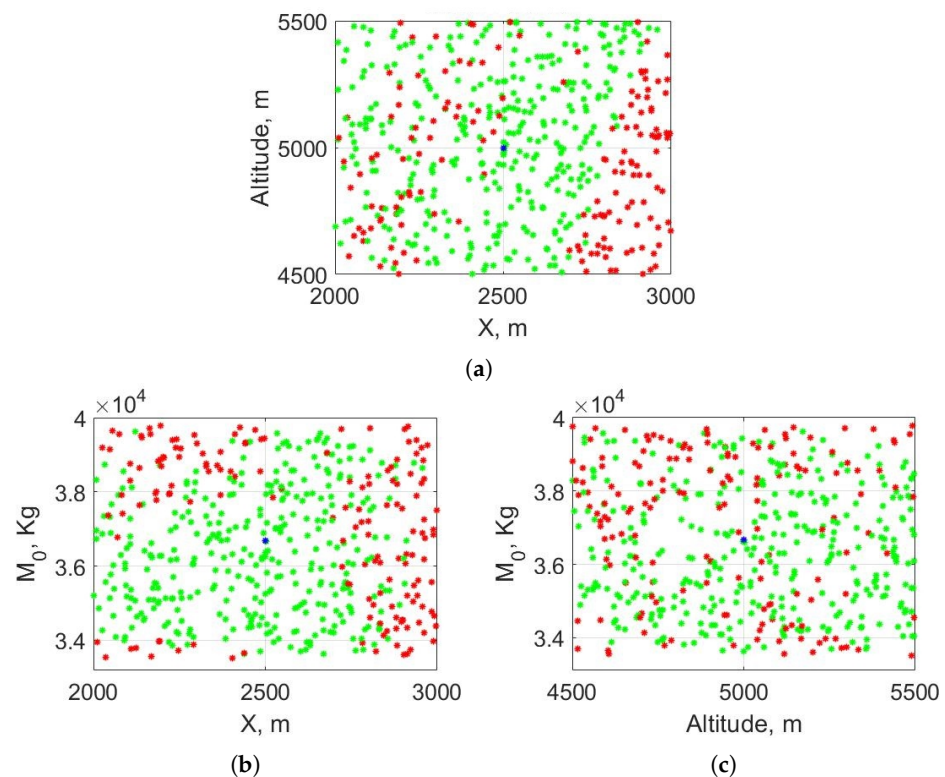


Figure 8. Monte Carlo results with variable initial position and initial mass. Subfigure (a) compares initial altitude vs horizontal range, Subfigure (b) compares initial mass vs horizontal range, Subfigure (c) compares initial mass vs. altitude. Results in green are feasible, in red are unfeasible, and the blue points mark the initial conditions of the nominal scenario. These three subfigures represent a three-dimensional figure considering *X*, altitude, and initial mass M_0 . Feasible and unfeasible close points have different values in the missing axis of each subfigure or are close to a feasibility boundary.

In Figure 8b,c, the initial mass is compared to each position component. Analysing both subfigures, it can be seen that, for a fixed initial velocity, the component that affects the feasibility the most is X . For $X \leq 2800$ m, there is a clear feasible zone, except for two regions where the mass is too high ($m_0 \geq 38$ tonnes) or too low ($m_0 \leq 34$ tonnes) and $X \leq 2500$ m. Figure 8c contains feasible solutions in the range of considered altitudes with a higher difficulty in landing the RLV correctly when the mass is close to the upper limit.

4.3.2. Monte Carlo Analysis with Fixed Initial Position

The second Monte Carlo test maintains the initial position as in the nominal scenario and varies the initial velocity and mass according to Table 6. The results obtained after 500 iterations are represented in Figure 9.

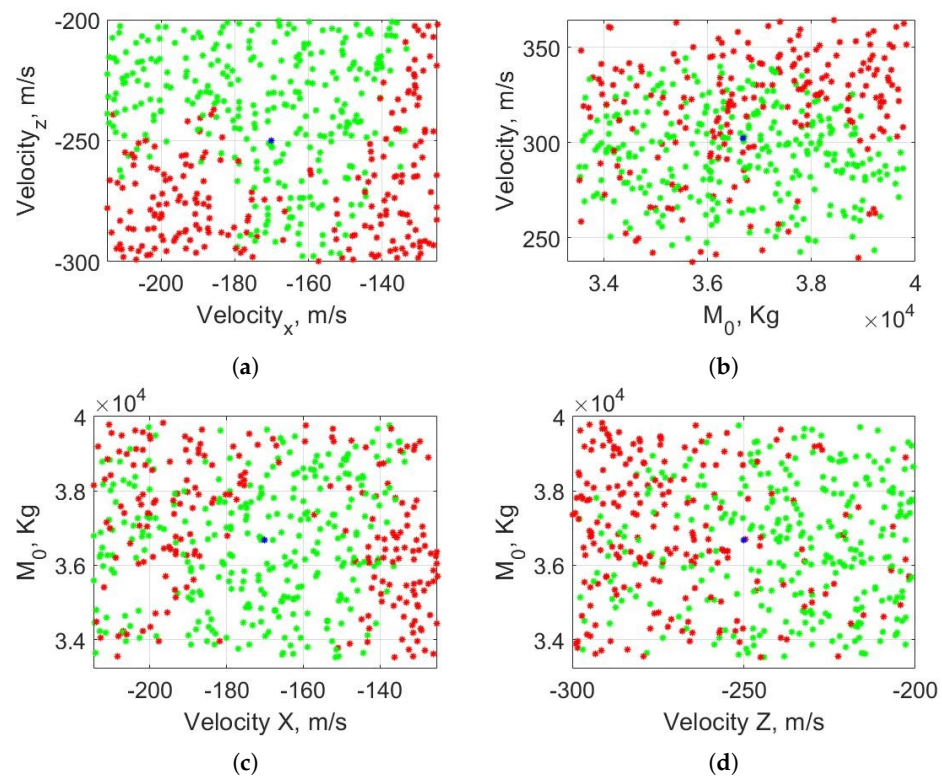


Figure 9. Monte Carlo results with variable initial velocity and initial mass. Results in green are feasible, results in red are unfeasible, and the blue points mark the initial conditions of the nominal scenario. Subfigure (a) shows the results with respect to the velocity components (V_x , V_z) and Subfigure (b) shows the results with respect to the velocity norm V and the initial mass M_0 . Subfigures (a,c,d) represent three-dimensional figure considering V_x , V_z and M_0 .

On Figure 9a, the feasible region represents all the cases in which the MPC algorithm was able to overcome the initial velocity. When the initial magnitude of V_x is small ($|V_x| < 150$ m/s), the RLV is not able to reach the landing target, especially as the initial $|V_z|$ becomes higher. Moreover, there is an unfeasible region when $|V_x| > 180$ m/s and $|V_z| > 250$ m/s, when $|V| > 310$ m/s, as seen in Figure 9b. This subfigure shows an unstable region around the nominal scenario conditions, but its feasibility is better explained in Figure 9c,d, where the initial mass m_0 is compared with each velocity component. The proposed MPC algorithm is robust for the zone around the nominal condition in Figure 9c, particularly when $|V_x| > 150$ m/s and independently of m_0 . As $|V_x|$ increases, the problem becomes more difficult to solve, in the same manner as in Figure 9d when $|V_z|$ reaches 300 m/s.

4.3.3. Monte Carlo Analysis with Fixed Initial Mass

The last Monte Carlo analysis for assessing the robustness of the MPC algorithm consists of 500 cases in which the initial mass was fixed and the position and velocity variate. Figure 10 presents the results obtained and reinforce the lessons already learned in the previous Sections 4.3.1 and 4.3.2. Specifically, in Figure 10a, the problem becomes harder to solve when the horizontal distance X with respect to the landing pad increases and, in Figure 10b the problem cannot be consistently solved in the bottom right region when the initial $|V_x|$ is smaller and the initial $|V_z|$ rises.

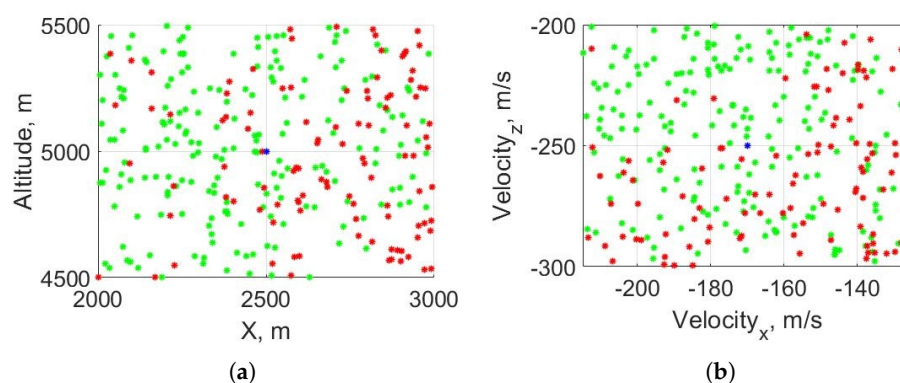


Figure 10. Monte Carlo results with variable initial position (a) and initial velocity (b). Results in green are feasible, those in red are unfeasible, and the blue points mark the initial conditions of the nominal scenario.

5. Conclusions

This paper further analyses the robustness of a proposed MPC strategy for the powered descent of an RLV. The strategy adjusts the prediction horizon using an update factor, allowing the RLV to land within a specified terminal horizon. The paper highlights the advantages of this approach over receding horizon MPC methods, reducing the need for an online search of a suitable prediction horizon when it is not fixed [10]. It can be tuned by properly setting the initial PH , TH , and UF values.

The numerical simulations and the tuning carried out within this paper show that this approach is viable and promising. The presented tuning strategy can be easily automatized and applied to systems with different RLVs and initial conditions, leading to different optimal sets of PH , TH and UF . The robustness of the tuning strategy, as well as the overall MPC algorithm, was assessed by analysing the performances and the feasibility regions by varying, under a Monte Carlo approach, the initial conditions of the vehicle. The qualitative interpretation of the results leads to stress the importance of the initial horizontal distance between the RLV and the landing site, and the initial vertical component velocity. A key parameter considered within these analyses was the propellant mass, which is strongly affecting the capability of the rocket to land. Sharp feasibility boundaries were indeed identified when initial parameters were pushed out of reasonable ranges. In the presented case of the landing of a Falcon 9 stage, the maximum horizontal distance could not be bigger than 2800 m and the initial vertical descent velocity could not exceed approx. 280 m/s. In such cases, the RLV could run out of fuel or could not be able to slow sufficiently fast the motion before touching the ground.

Preliminary results of the current work by the authors have been presented in [31], describing further research of a similar strategy on a six-DoF rigid body model with translational and rotational dynamics. Future work will test possible failure scenarios, i.e., an engine failure, and how the proposed MPC algorithm copes with them. Alternative future works will explore the possibilities of using artificial intelligence for the selection of the prediction horizon, i.e., tree search algorithms, as described in [16].

Author Contributions: Conceptualization, G.Z.P. and L.F.; methodology, G.Z.P. and L.F.; software, G.Z.P.; validation, G.Z.P. and L.F.; investigation, G.Z.P. and L.F.; writing—original draft preparation, G.Z.P.; writing—editing, G.Z.P.; writing—review, L.F. and E.G.-G.; visualization, G.Z.P.; supervision, L.F. and E.G.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Supporting data is available upon request to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Blackmore, L. Autonomous precision landing of space rockets. *Bridge* **2016**, *46*, 15–20.
2. Prakash, R.; Burkhart, P.D.; Chen, A.; Comeaux, K.A.; Guernsey, C.S.; Kipp, D.M.; Lorenzoni, L.V.; Mendeck, G.F.; Powell, R.W.; Rivellini, T.P.; et al. Mars Science Laboratory Entry, Descent, and Landing System Overview. In Proceedings of the IEEE Aerospace Conference Proceedings, Big Sky, MT, USA, 1–8 March 2008. [[CrossRef](#)]
3. Huang, X.; Xu, C.; Hu, J.; Li, M.; Guo, M.; Wang, X.; Zhao, Y.; Hua, B.; Wang, Y. Powered-descent landing GNC system design and flight results for Tianwen-1 mission. *Astrodynamics* **2022**, *6*, 3–16. [[CrossRef](#)]
4. Gallaher, M.; Coughlin, D.; Krupp, D. A guidance and control assessment of three vertical landing options for RLV. In Proceedings of the Guidance, Navigation, and Control Conference, San Diego, CA, USA, 29–31 July 1996. [[CrossRef](#)]
5. Lawden, D.F. *Optimal Trajectories for Space Navigation*; Butterworths: London, UK, 1963.
6. Meditch, J.S. On the problem of optimal thrust programming for a lunar soft landing. *IEEE Trans. Autom. Control* **1964**, *9*, 477–484. [[CrossRef](#)]
7. Eyles, D. *Sunburst and Luminary: An Apollo Memoir*; Fort Point Press: Boston, MA, USA 2018.
8. Açıkmeşe, B.; Ploen, S. A Powered Descent Guidance Algorithm for Mars Pinpoint Landing. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, USA, 15–18 August 2005. [[CrossRef](#)]
9. Ploen, S.; Acikmese, B.; Wolf, A. A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006. [[CrossRef](#)]
10. Trawny, N.; Benito, J.; Tweddle, B.E.; Bergh, C.F.; Khanoyan, G.; Vaughan, G.; Zheng, J.; Villalpando, C.; Cheng, Y.; Scharf, D.P.; et al. Flight testing of terrain-relative navigation and large-divert guidance on a VTVL rocket. In Proceedings of the AIAA SPACE 2015 Conference and Exposition, Pasadena, CA, USA, 31 August–2 September 2015. [[CrossRef](#)]
11. Szmuk, M.; Reynolds, T.P.; Açıkmeşe, B. Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints. *J. Guid. Control Dyn.* **2020**, *43*, 1399–1413. [[CrossRef](#)]
12. Liu, X. Fuel-Optimal Rocket Landing with Aerodynamic Controls. *J. Guid. Control Dyn.* **2019**, *42*, 65–77. [[CrossRef](#)]
13. Sagliano, M.; Mooij, E. Optimal drag-energy entry guidance via pseudospectral convex optimization. *Aerosp. Sci. Technol.* **2021**, *117*, 106946. [[CrossRef](#)]
14. Malyuta, D.; Yu, Y.; Elango, P.; Açıkmeşe, B. Advances in trajectory optimization for space vehicle control. *Annu. Rev. Control* **2021**, *52*, 282–315. [[CrossRef](#)]
15. Simplício, P.; Marcos, A.; Bennani, S. Reusable Launchers: Development of a Coupled Flight Mechanics, Guidance, and Control Benchmark. *J. Spacecr. Rocket.* **2020**, *57*, 74–89. [[CrossRef](#)]
16. Izzo, D.; Märten, M.; Pan, B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics* **2019**, *3*, 287–299. [[CrossRef](#)]
17. Visintini, A.; Di Carlo, M.; Belmonte Hernandez, C.; Macchiaiolo, L.; Berlin, M.; Neumann, N. Recent Trends in Computational Guidance and Control for Space Applications. In Proceedings of the ESA 12th International Conference on Guidance Navigation and Control and 9th International Conference on Astrodynamics Tools and Techniques, Sopot, Poland, 12–16 June 2023. [[CrossRef](#)]
18. Wang, K.; Chen, Z.; Li, J. Fuel-optimal powered descent guidance for lunar pinpoint landing using neural networks. *Adv. Space Res.* **2024**, *74*, 5006–5022. [[CrossRef](#)]
19. Li, W.; Song, Y.; Cheng, L.; Gong, S. Closed-loop deep neural network optimal control algorithm and error analysis for powered landing under uncertainties. *Astrodynamics* **2023**, *7*, 211–228. [[CrossRef](#)]
20. Pascucci, C.A.; Bennani, S.; Bemporad, A. Model predictive control for powered descent guidance and control. In Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015; pp. 1388–1393. [[CrossRef](#)]
21. Yang, L.; Liu, X.; Chen, W.; Zhou, H. Autonomous entry guidance using Linear Pseudospectral Model Predictive Control. *Aerosp. Sci. Technol.* **2018**, *80*, 38–55. [[CrossRef](#)]
22. Maciejowski, J.M. *Predictive Control with Constraints*; Prentice Hall: Hoboken, NJ, USA, 2000.
23. Miotto, P.; LePome, R.C. Design of a Model Predictive Control Flight Control System for a Reusable Launch Vehicle. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, USA, 11–14 August 2003.

24. Guadagnini, J.; Lavagna, M.; Rosa, P. Model Predictive Control for Reusable Space Launcher Guidance Improvement. In Proceedings of the 71 International Astronautical Congress (IAC), Online, 12–14 October 2020.
25. Scharf, D.P.; Açıkmese, B.; Dueri, D.; Benito, J.; Casoliva, J. Implementation and Experimental Demonstration of Onboard Powered-Descent Guidance. *J. Guid. Control Dyn.* **2017**, *40*, 213–229. [[CrossRef](#)]
26. Scharf, D.P.; Regehr, M.W.; Vaughan, G.M.; Benito, J.; Ansari, H.; Aung, M.; Johnson, A.; Casoliva, J.; Mohan, S.; Dueri, D.; et al. ADAPT Demonstrations of Onboard Large-Divert Guidance with a VTVLRocket. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2014.
27. Zaragoza Prous, G.; Felicetti, L. A Shrinking Horizon Model Predictive Control for Landing of Reusable Launch Vehicles. In Proceedings of the 73 International Astronautical Congress (IAC), Paris, France, 18–22 September 2022.
28. Sturm, J.F. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optim. Methods Softw.* **1999**, *11*, 625–653. [[CrossRef](#)]
29. SpaceX. *Falcon User's Guide*; Space Exploration Technologies Corp.: Hawthorne, CA, USA, 2021. Available online: <https://www.spacex.com/media/falcon-users-guide-2021-09.pdf> (accessed on 3 November 2022).
30. Williams, D.R. Earth Fact Sheet. Available online: <https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html> (accessed on 15 September 2020).
31. Zaragoza Prous, G.; Felicetti, L.; Grustan Gutierrez, E. Model predictive control strategy with a decreasing horizon interval for a reusable launcher in a landing scenario. In Proceedings of the 75 International Astronautical Congress (IAC), Milan, Italy, 14–18 October 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

A decreasing horizon model predictive control for landing reusable launch vehicles

Zaragoza Prous, Guillermo

2025-02-01

Attribution 4.0 International

Zaragoza Prous G, Grustan-Gutierrez E, Felicetti L. (2025) A decreasing horizon model predictive control for landing reusable launch vehicles. *Aerospace*, Volume 12, Issue 2, February 2025, Article number 111

<https://doi.org/10.3390/aerospace12020111>

Downloaded from CERES Research Repository, Cranfield University