75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.
 https://dl.iafastro.directory/event/IAC-2024/paper/84269/

Model Predictive Control Strategy with a Decreasing Horizon Interval for a Reusable Launcher in a Landing Scenario

Guillermo Zaragoza Prous[a]*, Leonard Felicetti[b] and Enric Grustan[c]

[a] Cranfield University, United Kingdom, g.zaragozaprous@cranfield.ac.uk
[b] Cranfield University, United Kingdom, leonard.felicetti@cranfield.ac.uk
[c] Cranfield University, United Kingdom, e.grustan@cranfield.ac.uk

* Corresponding author

Abstract

The descending and landing problem of an Reusable Launch Vehicle (RLV) concerns a wide variety of factors to overcome, such as the instability generated due to the aerodynamic forces during the descent phases and the strict requirements for accurate pinpoint landing to be met with limited control authority. In addition, the Guidance algorithm needs to be continuously and rapidly updated, in order to cope with the dynamically changing conditions that the RLV can experience during the re-entry and landing phase. One of the key technologies being studied to solve this problem is Model Predictive Control (MPC). MPC uses a linearized model of the problem to obtain a solution of the scenario, given a specific landing time in the future, called the prediction horizon (PH). In this paper, a new strategy to manage the PH of an MPC scheme is proposed for the landing scenario of an RLV. This strategy considers an offline predefined interval of PHs to obtain a valid solution instead of a single predefined PH. This strategy guarantees a wider set of feasible solutions to be searched with a Convex Optimization method, increasing the robustness of the algorithm at the guidance stage. A simulation setup is introduced for the landing scenario of an RLV, including full simulation of translational and rotational dynamics, along with the control laws to actuate each of the actuators of the vehicle. The results of the presented algorithm are then shown for the landing scenario of the first stage of a rocket.

Keywords: Model Predictive Control, Guidance, Convex Optimization, Reusable Launch Vehicle, Powered Descent Landing, Prediction Horizon

1. Introduction

Guidance and control during the propulsive landing phase of a space mission has received an increased interest during the recent years. Research of different strategies to guide a space system from an initial state to a desired state, like spacecraft *rendezvous* or a launcher landing, have been accelerated by the economical benefits that it implies. These techniques must be applied along a mission, increasing the autonomy and capabilities of a space vehicle using what is called Computational Guidance and Control (CGC) [1]. In CGC, the different loops and iterations happen in real time, by calculating online the actuation to be applied to complete successfully a mission. This calculation usually takes place within an iterative strategy called Model Predictive Control (MPC). The general concept of MPC is to solve a constrained Optimal Control Problem (OCP) repeatedly using new inputs from the Navigation subsystem [2]. As discussed in [3], there are 3 types of MPC: implicit, explicit or combined. In this paper a combined approach is followed, as there is an initial effort to tune the new variables before flight, yet the OCP is still calculated online.

One of the main parameters to consider when applying Convex Optimisation (CO) to a MPC scheme is the prediction horizon PH. The selected PH defines not only the feasibility of the problem but also the optimality of the solution, as different PH generate different propellant consumption. As mentioned in [4], the PH search for a constrained Powered Descent Guidance (PDG) problem is not trivial. The set of feasible prediction horizons has an unknown minimum, can be disconnected and can have an optimum on its boundary. In the present paper, a similar approach is taken as in [4]. An interval or range of PH are considered at each step of the MPC algorithm, being interpolated to define a specific number of PH to use during each iteration of the MPC. In Section 2 the model used for the simulation and its linearised version for the CO are presented, along with the problem statement and the constraints to be considered. One development performed in this section with respect to [5] is the update from a point-mass dynamic model to a rigid body dynamic model with translational and rotational dynamics in a three-dimensional environment. In Section 3, the Thrust Vector Control (TVC) to align the orientation of the vehicle with the thrust orientation from the Optimiser solution is presented. In Section 4 the new MPC strategy is presented and compared to the Shrinking Prediction Horizon Model Predictive Control (SPHMPC) in [5]. Later, the

numerical result of running a simulation of an example of a landing scenario with the Decreasing Horizon Interval Model Predictive Control (DHIMPC) is shown in Section 5. Finally, in Section 6 the Conclusions of the findings are discussed along with future work.

2. Problem Statement

In this section, a rigid body model is introduced along with the landing problem and the convexification of the dynamic equations and constraints. The equations of motion of the rigid model are based on two orthogonal reference frames: an Inertial Reference Frame (IRF) at the landing site and a non-inertial Body Reference Frame (BRF) that is fixed to the centre of mass (CoM) of the body. This second reference frame follows the same translational and rotational movement as the rigid body. The x, y and z axes in each reference frame are placed as follows: in the IRF, the x axis points upwards while the y and z axes are perpendicular between each other and form the horizontal plane; in the BRF, the x axis is parallel to the longitudinal axis of the body and the y and z axes define a plane parallel to the base of the body.

2.1. Rigid Body Model

The rigid body model defines the translational and rotational dynamics of the RLV in a three-dimensional model where the vehicle is subjected to the forces and torques generated by the engines and the atmospheric forces combined. The state vector of the model is characterised by the position of the vehicle (x, y, z), its orientation (ϕ, θ, ψ) with respect to the x, y and z of the inertial reference frame as Euler angles, its lineal (v_x, v_y, v_z) and angular $(\omega_x, \omega_y, \omega_z)$ velocities and the mass of the vehicle m. The Euler angles follow an E321 rotation over the x, y and z axes to translate the orientation of the rigid body between the IRF and the BRF. The state vector of the RLV is determined by the Equations 1, 2 and 3. These equations consider a non-rotating flat Earth with constant gravity for an RLV with a symmetric body around the x axis. Furthermore, there is no wind and the sloshing effect is neglected.

Equation 1 reflects the translation of the vehicle from the point of view of the IRF:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v_x} = \frac{T_x + D_x + L_x}{m} - g \\ \dot{v_y} = \frac{T_y + D_y + L_y}{m} \\ \dot{v_z} = \frac{T_z + D_z + L_z}{m} \end{cases} \tag{1}$$

where $\vec{T}$ is the thrust vector of the engines of the RLV, $\vec{L}$ and $\vec{D}$ are the lift and drag aerodynamic forces and g is the gravity force. Equation 2 describes the rotation of the vehicle, with the angular rate ω with respect to the BRF. The tensor of inertia of the rigid body is I_{RLV} and M is the moment exerted by the thrust and the aerodynamic forces on the body. The CoM of the vehicle used in the computation of the moments is defined as in [6], changing over time as the propellant is consumed.

$$\begin{cases} \dot{\phi} = (\omega_x \cos(\psi) - \omega_y \sin(\psi)) \sec(\theta) \\ \dot{\theta} = \omega_x \sin(\psi) + \omega_y \cos(\psi) \\ \dot{\psi} = \omega_z - (\omega_x \cos(\psi) - \omega_y \sin(\psi)) \tan(\theta) \\ \dot{\omega_x} = \frac{M_x - \dot{I}_{RLV,xx}\omega_x + I_{RLV,yy}\omega_y\omega_z - I_{RLV,zz}\omega_y\omega_z}{I_{RLV,xx}} \\ \dot{\omega_y} = \frac{M_y - \dot{I}_{RLV,yy}\omega_y - I_{RLV,xx}\omega_x\omega_z + I_{RLV,zz}\omega_x\omega_z}{I_{RLV,yy}} \\ \dot{\omega_z} = \frac{M_z - \dot{I}_{RLV,zz}\omega_z + I_{RLV,xx}\omega_x\omega_y - I_{RLV,yy}\omega_x\omega_y}{I_{RLV,zz}} \end{cases} \tag{2}$$

The rate of change of the mass is defined in Equation 3, where I_{sp} is the specific impulse of the rocket engine and g_0 is the gravity force at sea-level.

$$\dot{m} = -\frac{|\vec{T}|}{I_{sp} g_0} \tag{3}$$

The components of I_{RLV} in Equation 2 are defined by the shape of the rigid body, being it a cylinder with radius r and length l, and its mass. The mass is considered to be evenly distributed along the body. Equation 4 describes the allocation of mass along the longitudinal axis of the RLV, while Equation 5 illustrates the distribution along the transversal axes.

$$I_{RLV,xx} = \frac{1}{2} m r^2 \tag{4}$$

$$I_{RLV,yy} = I_{RLV,zz} = \frac{1}{12} m l^2 + \frac{1}{4} m r^2 \tag{5}$$

2.1.1. Aerodynamic Forces

The aerodynamic forces are the forces due to the interaction of the RLV with the atmosphere surrounding it. They depend on the the characteristics of the atmosphere, the shape of the vehicle and the velocity vector that defines the movement of the vehicle. The lift and drag forces are two vectors whose magnitude are defined by Equations 6 and 7, where $\rho(x)$ is the atmospheric density depending on the altitude, v is the magnitude of the velocity, S_{ref} is the reference area of the launcher and $C_D(v_{Mach})$ and $C_L(v_{Mach})$ are the drag and lift coefficients of the RLV. The values of these coefficients are obtained from [7], as well as the position of the centre of pressure (CoP) with respect to the velocity v_{Mach} in Mach.

$$D = \frac{1}{2} \rho(x) v^2 S_{ref} C_D(v_{Mach}) \tag{6}$$

$$L = \frac{1}{2} \rho(x) v^2 S_{ref} C_L(v_{Mach}) \tag{7}$$

The direction of $\vec{D}$ is defined as opposite to $\vec{v}$ in the BRF, while the direction of $\vec{L}$ is obtained as it is explained in Section 2.1.2.

2.1.2. Lift Force Direction

In order to calculate the direction of $\vec{L}$ in the BRF using analytic geometry, a generic velocity vector $\tilde{v}$ is taken into consideration. The direction of $\vec{L}$ is the intersection between the plane determined by the x axis (the orientation of the vehicle) and $\tilde{v}$, and the plane perpendicular to $\tilde{v}$. The equation of the first plane can be obtained from the unit vectors of the velocity and the orientation, and a point belonging to the plane, in this case the origin of the BRF. The equation of this plane is defined as:

$$\tilde{v}_z y - \tilde{v}_y z = 0 \tag{8}$$

The equation of the second plane is obtained from its normal vector, $\tilde{v}$, and a point, the origin of the BRF, which results in the following equation:

$$\tilde{v}_x x + \tilde{v}_y y + \tilde{v}_z z = 0 \tag{9}$$

Performing the cross product of the normal vectors of both planes, $(0, \tilde{v}_z, -\tilde{v}_y)$ and $(\tilde{v}_x, \tilde{v}_y, \tilde{v}_z)$, the equation of the line that describes the orientation of $\vec{L}$ is:

$$(\tilde{v}_y^2 + \tilde{v}_z^2)x - \tilde{v}_x\tilde{v}_y y - \tilde{v}_x\tilde{v}_z z = 0 \tag{10}$$

The direction of $\vec{L}$ in the BRF will be defined by the sign of $\tilde{v}_x$, being $(\tilde{v}_y^2 + \tilde{v}_z^2, -\tilde{v}_x\tilde{v}_y, -\tilde{v}_x\tilde{v}_z)$ when $\tilde{v}_x$ is positive and $(-\tilde{v}_y^2 - \tilde{v}_z^2, \tilde{v}_x\tilde{v}_y, \tilde{v}_x\tilde{v}_z)$ when $\tilde{v}_x$ is negative.

2.1.3. Moments

The moment exerted to the body $\vec{M}$ during the flight is composed by the moment generated by the Thrust Vector Control $\vec{M}_{TVC}$ and the aerodynamic forces $\vec{M}_A$. The sum of both moments is used in Equation 2 to define the rate of change of ω. The positions along the x axis of the RLV taken into account to calculate $\vec{M}_A$ and $\vec{M}_{TVC}$ in Equations 11 and 12 refer to the distance between that position and the base of the rocket, being x_{CoM} the position of the CoM, x_{CoP} the position of the CoP and x_{TVC} the position of the gimbal of the TVC.

$$\vec{M}_A = \begin{bmatrix} 0 \\ (x_{CoM} - x_{CoP})(L_z + D_z) \\ -(x_{CoM} - x_{CoP})(L_y + D_y) \end{bmatrix} \tag{11}$$

$$\vec{M}_{TVC} = \begin{bmatrix} 0 \\ (x_{CoM} - x_{TVC})T_z \\ -(x_{CoM} - x_{TVC})T_y \end{bmatrix} \tag{12}$$

2.2. Definition of Landing Problem

The approach to solve the landing problem of a RLV follows the steps described in [5], with the difference of considering three dimensions. It is defined as a Two Boundary Value Problem (TBVP), where the initial state at t_0 is given at the beginning of the optimization phase, in Equation 13, and the final state at t_f is placed at the origin of the coordinate system with $|\vec{v}|$ equal to 0, in Equation 14.

$$\begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \\ z(t_0) = z_0 \end{cases} \quad \begin{cases} v_x(t_0) = v_{x_0} \\ v_y(t_0) = v_{y_0} \\ v_z(t_0) = v_{z_0} \end{cases} \tag{13}$$

$$\begin{cases} x(t_f) = 0 \\ y(t_f) = 0 \\ z(t_f) = 0 \end{cases} \quad \begin{cases} v_x(t_f) = 0 \\ v_y(t_f) = 0 \\ v_z(t_f) = 0 \end{cases} \tag{14}$$

In Equation 15, the cost function considered for the TBVP minimises the required thrust, which in fact minimises the consumption of the propellant mass. At any time during flight, the thrust magnitude shall be sustained within the limits of operation of the engines. Equation 16 ensures that the thrust generated by the guidance algorithm is contained within those limits.

$$J = \int_0^{t_f} |T| \, dt \tag{15}$$

$$0 < T_{min} \leq T(t) \leq T_{max} \tag{16}$$

Moreover, it is mandatory to keep the position of the RLV above the surface of the Earth, which is enforced by the definition of the constraint in Equation 17. Meanwhile, the RLV engines should always point downwards, which leads to Equation 18, as the vertical component of the thrust must point upwards at any time of the manoeuvre.

$$0 \leq x(t) \tag{17}$$

$$0 \leq T_x(t) \tag{18}$$

The final state of the RLV is likewise restricted by Equation 19, when the final mass of the RLV must be larger than its dry mass.

$$m_{dry} \leq m(t_f) \tag{19}$$

The orientation defined by (ϕ, θ, ψ) of the RLV is constrained by Equations 20 and 21, where κ is a predefined boundary parameter that can change over time. The angle κ restricts the values of the possible orientations of the RLV at any time of the flight.

$$|\frac{\pi}{2} - \kappa(t)| \geq |\frac{\pi}{2} - \theta(t)| \tag{20}$$

$$|\frac{\pi}{2} - \kappa(t)| \geq |\frac{\pi}{2} - \psi(t)| \tag{21}$$

2.3. Convexification of the Problem

The equations and constrains presented in Section 2.2 must be convex allowing the convex optimiser to find a solution. Nevertheless, there are two sources of non-convexity, which are Equation 3 and Equation 16. The strategy to convexify these two equations followed in this paper is described in [5] and [8]. For solving the non-convexity in Equation 3, a change of variable, as in Equation 22, is introduced to take into account the non-linear decrease of the RLV mass.

$$\eta = ln(m) \tag{22}$$

By defining the fuel consumption rate as in Equation 23, the Equation 3 can be rewritten as in Equation 24. σ is defined later on in Equation 26.

$$\beta = -\frac{1}{I_{sp}\ g_0} \tag{23}$$

$$\dot{\eta} = \frac{\dot{m}}{m} = -\beta \cdot \sigma \tag{24}$$

The source of non-convexity in Equation 16 is tackled by introducing a slack variable Γ as in Equation 25. According to Lemma 1 in [8], this equation converts to $\left\|\vec{T}\right\| = \Gamma$ for the optimal solution.

$$\left\|\vec{T}\right\| \leq \Gamma \tag{25}$$

Two new variables (σ and $\vec{u}$) can be defined, in Equations 26 and 27, as the magnitude and the vector of the specific thrust, respectively. These are used to redefine Equation 25 as in Equation 28.

$$\sigma \triangleq \frac{\Gamma}{m} \tag{26}$$

$$\vec{u} \triangleq \frac{\vec{T}}{m} \tag{27}$$

$$\|\vec{u}\| \leq \sigma \tag{28}$$

The objective index defined in Equation 15 can now be reformulated with the new σ variable as in Equation 29.

$$J = \int_0^{t_f} \sigma\ dt \tag{29}$$

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 30, where the coefficient γ that multiplies v counts for part of the effect of the atmospheric drag, with the velocity not being squared to avoid non-linearities.

$$\begin{cases} x(t+\Delta t) = \frac{1}{2}(u_x(t) - g - \gamma v_x(t))\Delta t^2 + v_x(t)\ \Delta t + x(t) \\ y(t+\Delta t) = \frac{1}{2}(u_y(t) - \gamma v_y(t))\Delta t^2 + v_y(t)\ \Delta t + y(t) \\ z(t+\Delta t) = \frac{1}{2}(u_z(t) - \gamma v_z(t))\Delta t^2 + v_z(t)\ \Delta t + z(t) \\ v_x(t+\Delta t) = (u_x(t) - g - \gamma v_x(t))\Delta t + v_x(t) \\ v_y(t+\Delta t) = (u_y(t) - \gamma v_y(t))\Delta t + v_y(t) \\ v_z(t+\Delta t) = (u_z(t) - \gamma v_z(t))\Delta t + v_z(t) \\ \eta(t+\Delta t) = -\beta\sigma(t)\Delta t + \eta(t) \end{cases} \tag{30}$$

3. Thrust Vector Control

In order to follow the trajectory calculated by the convex optimiser and correct the error generated by the differences between the convexified model in Equation 30 and the non-linear model presented in Equations 1, 2 and 3, a TVC has been implemented. The TVC is based in two single input single output controllers that manage the pitch and yaw orientation of the RLV, decoupling the problem into two separated sub-problems as the roll is assumed to be negligible. The thrust magnitude given as output of the optimiser is directly applied to the non linear plant and therefore it is not controlled.

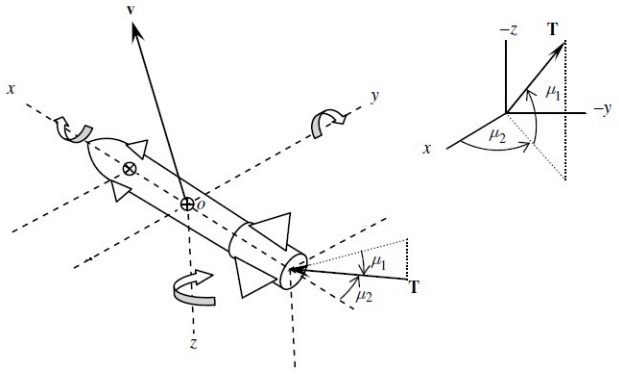


Figure 1: Definition of the TVC angles μ_1 and μ_2 [9]

3.1. Definition of the TVC angles

The control action of the Thrust Vector Control is based in the operation of a gimbal at the base of the rocket engine. The orientation of the gimbal defines the orientation of the thrust applied by the engine. This orientation is defined in the BRF by two angles, μ_1 and μ_2, which are shown as in Figure 1 [9]. The components of $\vec{T}_{TVC}$ can be obtained from these two angles as in Equation 31.

$$\vec{T}_{TVC} = \begin{bmatrix} T\cos(\mu_1)\cos(\mu_2) \\ -T\cos(\mu_1)\sin(\mu_2) \\ -T\sin(\mu_1) \end{bmatrix} \tag{31}$$

Considering that the TVC actuator has a maximum angle of deflection relatively small, and for the purpose of linearising the equations of motion to tune the PID controllers used in Section 3.2 in different points of operation, the equations to obtain the components $T_{TVC,y}$ and $T_{TVC,z}$ can be simplified as Equations 32 and 33.

$$T_{TVC,y} = -T\mu_2 \tag{32}$$

$$T_{TVC,z} = -T\mu_1 \tag{33}$$

3.2. Gain Scheduling PID Controller

Gain Scheduling is a methodology to control systems with non-linearities using a group of linear controllers. These are tuned to successfully control the system at various points of operation. This strategy has been used previously for the landing problem of a RLV in [10] and [11], employing a PID as the linear controller. Likewise, gain scheduling has been used in [12] to implement thrust vector control utilizing a Linear Quadratic Regulator (LQR) with additional integral action.

In the present paper, two PID controllers are considered to reduce the error between the orientation of the RLV (ϕ, θ, ψ) and the orientation of $\vec{T}$, one for the control of θ and the other for the control of ψ. This is achieved by setting the reference of each closed loop PID controllers to 0, which will align both orientations in the BRF. In order to tune the two PID controllers, a linearisation process must be followed to obtain a linear plant from the non-linear equations of motion (Equations 1, 2 and 3). This linearisation is implemented about $\theta = 0$ *rad* and $\psi = 0$ *rad* for each case.

The first PID controller manages the angle θ in a linear state space plan with state $state_\theta = [\theta, \omega_y, v_z, m]$, output $output_\theta = [\theta]$ and control input $control_\theta = [\mu_1]$. The second PID controller controls the angle ψ in a linear state space plan with state $state_\psi = [\psi, \omega_z, v_y, m]$, output $output_\psi = [\psi]$ and control input $control_\psi = [\mu_2]$.

Algorithm 1 Shrinking Prediction Horizon MPC Algorithm [5]

```
Define TH, PH, Δt_I, Δt_P, UF and X_0
while PH > Δt_I do
    for iteration = 0, 1, 2, ..., TH · UF do
        U = convexOptimiser(X, PH, Δt_P)
        Apply U for Δt_I
    end for
    Update TH = TH · (1 − UF)
    Update PH = PH · (1 − UF)
end while
Apply the rest of U
```

The PID Gain Scheduler employed is based in a lookup table for each gain (K_P,K_I,K_D and N) of each PID controller. The input of these lookup tables are the mass and velocity of the RLV at a specific moment. The values of the gains are obtained by tuning the PID controllers considering the following limitations: an overshoot less than 15%, a raising time less than 1 *second*, a settling time less than 10 *seconds* and a phase margin greater than 60 *degrees*.

4. Model Predictive Control with a Decreasing Prediction Horizon Interval Strategy

In [5], it was discussed why a decreasing prediction horizon strategy was more suitable for the Powered Descent Guidance problem compared to a receding prediction horizon strategy. It was hypothesised that the prediction horizon could be updated as a function of a time-scaling factor UF and the time that has passed since the last iteration of the MPC strategy. The complete algorithm is described in Algorithm 1 for reference and it is the exact same as in [5].

In this section, the authors introduce an alteration to the decision-making logic during each MPC cycle, selecting a PH from a set of possible PHs, according to two new parameters:

- *Prediction Horizon interval* ($PH_{interval}$): defines the minimum and maximum PH values to be considered at each MPC cycle.

$$PH_{min,max} = PH \pm \frac{1}{2} \cdot PH_{interval} \tag{34}$$

- *Prediction Horizon division* PH_{div}: defines the granularity of the linear interpolation between the extremes of the interval

$$PH_i = PH_{min} + \frac{(PH_{max} - PH_{min})}{PH_{div}} \cdot i \tag{35}$$

  for $i = [0, PH_{div}]$.

The DHIMPC algorithm must be tuned previously to initiate the landing scenario. This tuning phase follows the same approach as introduced in [5]. The Terminal Horizons TH and Prediction Horizon PH, the time increment between MPC iterations Δt_I and states within the optimisation problem Δt_P, the UF and the initial state X_0 shall be previously defined. Once the landing problem is specified, the DHIMPC can be started at a desired moment during the descent of the RLV. While the PH is greater than Δt_I, the algorithm will try to find a solution to the problem using the state of the vehicle at that point in time. Instead of using a single PH as in Algorithm 1, the DHIMPC creates a set of PH as specified before

and employs it to obtain feasible solutions for a period of time equal to $TH \cdot \text{UF}$. Depending on the implementation, these optimisations could be run in parallel, which minimises the difference in execution time between both algorithms. Afterwards, the best solution out of the feasible ones is selected, with a criteria that might vary from one landing problem to another, as presented in Section 4.1. Later, the TH and PH parameters are updated and when the PH is smaller than Δt_I, the last feasible solution is applied until the RLV lands.

The DHIMPC is considered to be more robust than the SPHMPC as it covers a larger PH time frame at each MPC iteration, which allows to find a feasible solution even in cases where the linear equations of motion of the optimiser and the non-linear equations of motion of the system plant diverge greatly. Moreover, the selection of the best feasible solution at the end of each MPC iteration provides a second layer of optimisation to the problem, where other variables may have a greater weight in the decision-making process than the final mass of the RLV.

Algorithm 2 Decreasing Horizon Interval MPC Algorithm

Define TH, PH, Δt_I, Δt_P, UF, $PH_{interval}$, PH_{div} and X_0
while $PH > \Delta t_I$ **do**
  Create array of PH
  for $iteration = 0, 1, 2, \ldots, TH \cdot \text{UF}$ **do**
    for $PH_i = PH_{min}, \ldots, PH_{max}$ **do**
      U_i = convexOptimiser(X, PH_i, Δt_P)
    end for
    U = selectBestFeasibleSol($[U_{min}, \ldots, U_{max}]$)
    Apply U for Δt_I
  end for
  Update $TH = TH \cdot (1 - \text{UF})$
  Update $PH = PH \cdot (1 - \text{UF})$
end while
Apply the rest of U

Rocket	Falcon 9 1st Stage
N. Engines used	1
Engine Thrust [kN]	845
Specific Impulse [s]	311
Total Propellant Mass [t]	395.7
Dry Mass [t]	25.6
Length [m]	41.2
Diameter [m]	3.66
Maximum Thrust [%]	0.8
Minimum Thrust [%]	0.1

Table 1: Falcon 9 first stage parameters used in the simulations [13].

4.1. Selection of the Best Feasible Solution

A benefit of Algorithm 2 compared to Algorithm 1 is the addition of a selection step after obtaining the optimised trajectories for the various PH considered. This selection step can be implemented in multiple ways, being the final mass the most obvious performance variable. Other ways to choose the most suitable feasible solution could be:

- the closest initial thrust magnitude compared to the current value
- the closest initial orientation, if it is important to reduce the orientation error due to an unconstrained initial orientation in the convex optimisation phase of the MPC
- the less total angular rotation over the projected trajectory
- a pondered system containing some of the previous variables and others

Variable	Initial Value
Position $X[m]$	9000
Position $Y[m]$	2500
Position $Z[m]$	2500
Euler Angle $\phi[rad]$	0
Euler Angle $\theta[rad]$	0
Euler Angle $\psi[rad]$	0
Velocity $v_x[m/s]$	-320
Velocity $v_y[m/s]$	-160
Velocity $v_z[m/s]$	-160
Angular Rate $\omega_x[rad/s]$	0
Angular Rate $\omega_y[rad/s]$	0
Angular Rate $\omega_z[rad/s]$	0
Propellant percentage [%]	2.8
Mass m [t]	36.68
Orientation κ [rad]	$\frac{\pi}{4}$

Table 2: Initial conditions for the Landing Scenario.

5. Simulation and Numerical Results

This section illustrates the simulation of a landing scenario of the first stage of the Falcon 9 launcher, assessing the suitability of the proposed guidance and control algorithms to solve the landing problem of an RLV. The RLV model used for this simulation is a representation of the first stage of the Falcon 9 rocket, which is nowadays the only orbital reusable launch vehicle being operated. The characteristics of the RLV model employed in the simulations are described in Table 1. The propellant percentage mentioned in Table 2 is multiplied by the total propellant mass of the Falcon 9 plus its dry mass to obtain the initial mass of the vehicle at the initial state vector of the simulation.

The initial state of the RLV is described in Table 2. Starting at an altitude of 9 *km* and a horizontal distance of 3.5 *km*, the total position difference to the landing pad is roughly 10 *km* and the magnitude of the velocity is about 400 *m/s*. The propellant mass available before the landing phase of the flight is 11 *tons*. The angle κ used as a boundary for the orientation of the solution calculated in the convex optimiser (Equations 20 and 21) starts at $\frac{\pi}{4}$ *radians* and increases over time within a feasible solution. It ensures that the generated trajectory lands the RLV vertically.

Parameter	Initial Value
Prediction Horizon $PH[s]$	37
Terminal Horizon $TH[s]$	70
Update Factor $UF[-]$	0.12
PH Interval $PH_{interval}[s]$	6
PH Division $PH_{div}[-]$	10
MPC time interval $\Delta t_I[s]$	2
Prediction time interval $\Delta t_P[s]$	0.1

Table 3: DHIMPC Parameter tuning for the landing scenario simulation.

Table 3 contains the tuning of the specific parameters used by the DHIMPC strategy. The interval of considered PH is 6 *seconds*, with a division of 10 elements evenly distributed within the considered values. These values produce a time gap of 0.6 *seconds* between each PH considered, which gives a good granularity over the $PH_{interval}$. The initial PH is 37 *seconds* and the TH 70 *seconds*. These values are based on the analysis performed in [5] and in previous simulations performed by the authors, where an iterative approach has been followed to find the correct values. Besides, the 2 seconds of the Δt_I replicate the real limitations of Apollo's Lunar Lander [14].

The trajectory of the RLV is presented in Figure 2 along with the thrust vector at each position. At the beginning of the simulation there is no thrust while the convex optimiser is calculating the first possible feasible solutions over the $PH_{interval}$. It is also confirmed by Figure 4, where the thrust magnitude is 0 during the first 2 *seconds*. Then, the RLV starts a phase where it decreases its velocity in the 3 axes until around *seconds* 20 to 25 of the simulation, as seen in Figure 3. Afterwards, the velocity in the x axis is maintained while the DHIMPC algorithm reduces the velocity in the y and z axes and corrects the overshoot in both axes due to a high initial velocity components v_y and v_z. This correction can be also seen in Figure 2 in the last part of the trajectory, where the orientation of the thrust of the RLV increases its angle of incidence to counteract this overshoot. In Figure 3, the v_y and v_z components seem to be equal in value, but it is due to the scale of the plots, as the state vector of the simulation shows differences between them. As soon as the y and z components of the state vector are close to 0, the v_x decreases to land close to the landing pad.

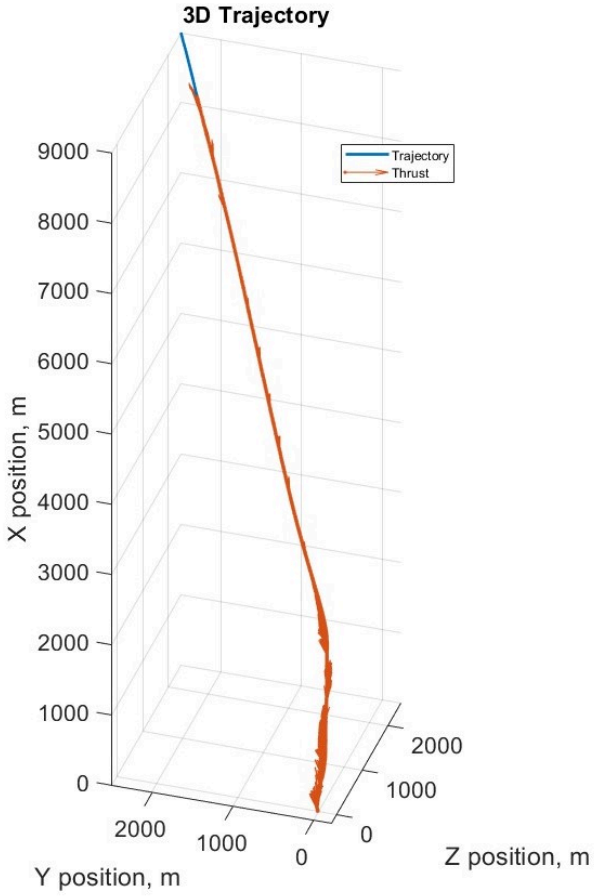
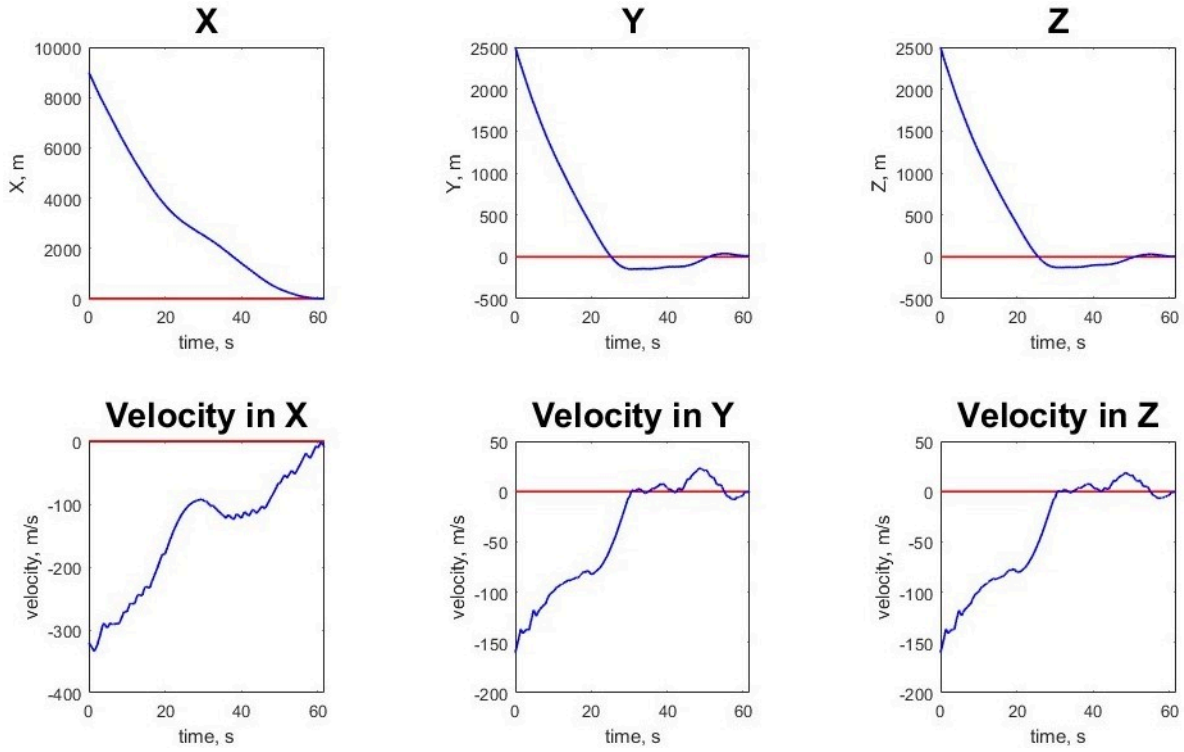


Figure 2: RLV 3D trajectory with the thrust vector represented in orange.

The final position of the RLV for this simulation is slightly moved in the y axis, with 10.88 m of error. Likewise, the vehicle is travelling vertically at 7.6 *m/s*, which is a bit high for the final landing. In fact, 1 s before landing, the velocity in the x axis is almost -2 *m/s*, but as the guidance and control algorithms try to correct the error in the y and z axes, the RLV decreases the thrust in the vertical axis, removing part of the compensation for the gravity force. The linearised equations of motion used within the optimiser do not take into account many of the non linearities of the RLV, which raises its importance close to the ground, as there is less margin to correct the deviation between both models. A solution to this problem could be implementing an specific controller for the last meters before touchdown. Therefore, assigning a higher importance to maintain a desired v_x rather than to correct the error in y and z. This approach is also followed in [15],



(a)

Figure 3: RLV Position and velocity over time.

when the hopper changes to an Auto-land system before touchdown.

Variable	Value	Value
Simulation Time $[s]$	60.8	61.7
Position $X[m]$	4.46	-0.28
Position $Y[m]$	11.24	10.88
Position $Z[m]$	5.8	5.41
Euler Angle $\phi[rad]$	-0.04	-0.03
Euler Angle $\theta[rad]$	-0.14	-0.14
Euler Angle $\psi[rad]$	0.22	0.35
Velocity $v_x[m/s]$	-2.03	-7.63
Velocity $v_y[m/s]$	-0.53	0.18
Velocity $v_z[m/s]$	-0.52	-0.26
Angular Rate $\omega_x[rad/s]$	0	0
Angular Rate $\omega_y[rad/s]$	0	-0.02
Angular Rate $\omega_z[rad/s]$	0.12	0.15
Mass m $[t]$	28.31	28.28

Table 4: Final state of the Landing Scenario simulation.

In the presented simulation, the RLV lands at time $t = 61.7$ *seconds*, which is compliant with the TH selected, i.e 70 *seconds*. When the RLV lands, the mass of the RLV is 28.3 t, which means that there is 2.8 t of propellant mass at the end of the simulation. Figure 4 shows the profile of the depletion of mass over time, with a clear relation between its rate of change and the moments in time when the thrust magnitude is at its maximum value.

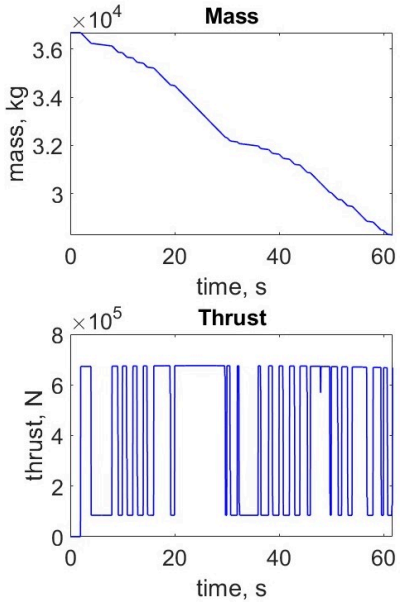


Figure 4: RLV Mass and thrust magnitude over time.

During the simulation of the landing scenario, the DHIMPC executed in parallel the 10 optimisations re-

quired by the PH_{div}. These optimisations were performed using the SeDuMi [16] library within a MATLAB/Simulink environment. Figure 5 presents the execution times of each of the iterations of the algorithm. The initial iterations are close or surpass the limitation of 2 seconds that the Δt_I imposes, but as the PH utilised decrease, the execution time is clearly within limits. This occurs as the number of constraints and variables that the optimiser has to compute is directly proportional to PH.

Figure 5: Execution time of the DHIMPC over the simulation.

6. Conclusion

The Prediction Horizon interval offers a more robust approach to ensuring a soft landing by providing multiple solutions at each MPC iteration. By leveraging a parallel approach, the iterations over the PH interval can be computed within a reasonable time frame of approximately 1.5 seconds, depending on the value of the PH. Furthermore, the algorithm presented allows for an optimal trade-off by selecting the best solution from the feasible ones obtained at each MPC iteration. The simulation and numerical results presented demonstrate the feasibility of solving the landing problem using the proposed DHIMPC.

Future work comprises the enhancement of the simulator by improving the accuracy of the RLV mass distribution, accounting for sloshing effects, and incorporating more actuators and their dynamics. A Monte Carlo analysis shall be performed to compare the robustness of the DHIMPC with respect to SPHMPC. Likewise, the linearised model used for the Gain Scheduling PID controllers shall be updated to be defined over a trajectory. Moreover, the algorithm will be implemented as part of a Processor-In-The-Loop setup to assess its performance with a processor with a similar characteristics as those currently used in space.

References

[1] P. Lu. Introducing computational guidance and control. *Journal of Guidance, Control, and Dynamics*, 40(2):193–193, 2017. doi: 10.2514/1.G002745.

[2] A. Visintini, M. Di Carlo, C. Belmonte Hernandez, L. Macchiaiolo, M. Berlin, and N. Neumann. Recent trends in computational guidance and control for space applications. *ESA GNC-ICATT 2023*, 2023.

[3] U. Eren, A. Prach, B. Koçer, S. Rakovic, E. Kayacan, and B. Açikmese. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017. ISSN 0731-5090. doi: 10.2514/1.G002507.

[4] D. P. Scharf, B. Açıkmeşe, D. Dueri, J. Benito, and J. Casoliva. Implementation and experimental demonstration of onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):213–229, 2017. doi: 10.2514/1.G000399.

[5] G. Z. Prous and L. Felicetti. A shrinking horizon model predictive control for landing of reusable launch vehicles. *73 International Astronautical Congress (IAC)*, 2022.

[6] M. R. Chiarelli, S. Carbutti, G. Mariani, G. Palaia, and K. A. Salem. Re-entry dynamics of the reusable stage of a space launcher: a first level model. In *2023 IEEE 10th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 545–550, 2023. doi: 10.1109/MetroAeroSpace57412.2023.10190019.

[7] S. Carbutti and G. Mariani. *Master Science Thesis Re-entry dynamics and state-space model for the reusable stage of a space launcher*. University of Pisa, 2022.

[8] B. Açıkmeşe and S. Ploen. A powered descent guidance algorithm for mars pinpoint landing. In *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, United States of America, Aug. 2005. ISBN 978-1-62410-056-7. doi: 10.2514/6.2005-6288.

[9] A. Tewari. *Advanced Control of Aircraft, Spacecraft and Rockets*. Wiley, 2011. doi: 10.1002/9781119971191.

[10] A. D. Oliveira and M. Lavagna. Reusable launchers reentry controlled dynamics simulator. *9 EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)*, 2022. doi: 10.13009/EUCASS2022-6193.

[11] J. Guadagnini, M. Lavagna, and P. Rosa. Model predictive control for reusable space launcher guidance improvement. *Acta Astronautica*, 193:767–778, 2022. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2021.10.014.

[12] P. dos Santos and P. Oliveira. Thrust vector control and state estimation architecture for low-cost small-scale launchers, 2023. URL `https://arxiv.org/abs/2303.16983`.

[13] Anonymous. *Falcon User's Guide*. SpaceX, 2021. URL `https://www.spacex.com/media/falcon-users-guide-2021-09.pdf`.

[14] D. Eyles. *Sunburst and Luminary: An Apollo Memoir*. Fort Point Press, 2018.

[15] D. P. Scharf, M. W. Regehr, G. M. Vaughan, J. Benito, H. Ansari, M. Aung, A. Johnson, J. Casoliva, S. Mohan, D. Dueri, B. Açikmeşe, D. Masten, and S. Nietfeld. Adapt demonstrations of onboard large-divert guidance with a vtvl rocket. In *2014 IEEE Aerospace Conference*, 2014. doi: 10.1109/AERO.2014.6836462.

[16] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999. doi: 10.1080/10556789908805766. URL `https://doi.org/10.1080/10556789908805766`.

Model predictive control strategy with a decreasing horizon interval for a reusable launcher in a landing scenario

Zaragoza Prous, Guillermo

2024-10-18

Zaragoza Prous G, Felicetti L, Grustan Gutierrez E. (2024) Model predictive control strategy with a decreasing horizon interval for a reusable launcher in a landing scenario. IAC2024 International Astronautical Congress, 14-18 October 2024, Milan, Italy
https://dl.iafastro.directory/event/IAC-2024/paper/84269/