

Onboard Guidance for Reusable Rockets: Aerodynamic Descent and Powered Landing

Marco Sagliano¹, Ansgar Heidecker¹, José Macés Hernández¹, Stefano Farì¹, Markus Schlotterer¹, Svenja Woicke¹, David Seelbinder¹, and Etienne Dumont¹

⁽¹⁾*German Aerospace Center, Robert Hooke Str. 7, 28359, Bremen, Germany*

This paper describes a novel general on-board guidance strategy which can be applied to both the aerodynamically-controlled descent and the powered landing phase of reusable rockets. The proposed guidance method is based on sequential convex optimization applied to a Cartesian representation of the equations of motion. The contributions are an exploitation of convex and non-convex contributions, which are processed separately to maximize the computational efficiency of the approach, the inclusion of highly nonlinear terms represented by aerodynamic accelerations, a complete reformulation of the problem based on the use of Euler angle rates as control means, an improved transcription based on the use of a generalized hp pseudospectral method, and a dedicated formulation of the aerodynamic guidance problem for reusable rockets. The problem is solved for a 40 kN-class reusable rocket. Results show that the proposed technique is a very effective methodology able to satisfy all the constraints acting on the system, and can be potentially employed online to solve the entire descent phase of reusable rockets in real-time.

I. Introduction

The second decade of the new millennium has shown that we are at the beginning of a second space race. While the first one was peremptorily fought by United States and Soviet Union to establish a global technical and military supremacy between the capitalist and the communist doctrines, all the clues let think that the second one will have a much sparser structure, with multiple players acting on both the exploration and the exploitation aspects of space activities, and for the first time, with an active role of the private industry, led by SpaceX [1]. The declared objective of the company, according to the words of the founder and CEO Elon Musk, is to "Make Humans a Multi-Planetary Species" [2].

SpaceX is one of the main industrial players, but not the only one. In fact, Blue Origin made impressive steps forward with the New Shepard [3], and aims now at landing astronauts on the Moon by 2024 [4]. In this compound scenario agencies and intergovernmental institutions are required to update and revisit their plans. NASA revealed its strategy for coming back to the Moon and preparing the conquest of Mars in the '30s [5]. China is successfully continuing the robotic exploration of our natural satellite with its Chang'e program, with the last mission of Chang'e 4 which focused on the far side side of the Moon [6].

With the aim to develop strategic technologies in the frame of a wider reusability-focused program the German Aerospace Center (DLR), the Japan Aerospace Exploration Agency (JAXA), and the French National Centre for Space Studies (CNES) joined in a trilateral agreement to develop and demonstrate the technologies that will be needed for future reusable launch vehicles. In the joint project CALLISTO (Cooperative Action Leading to Launcher Innovation in Stage Toss back Operations) [7] a demonstrator is developed and built. As long-term objective this project aims at paving the way to develop a rocket that can be reused, and the joint efforts of the three agencies will culminate in a series of flights that will performed from the Kourou Space Center (KSC), in French Guiana.

Within the trilateral agreement two lines of development of Guidance and Control (G&C) subsystems take place in parallel for CALLISTO. Specifically, DLR and JAXA decided to strengthen their synergy and proceed with the development of a unique, fully integrated G&C subsystem [8]. The mission consists of multiple flight phases, which correspond to different aerodynamic configurations of the vehicle. Specifically four main phases of flight can be defined to better frame the problem: the ascent phase, the boostback maneuver, the aerodynamic phase, and the powered descent and landing phase. Therefore, a plethora of modern methods is needed to successfully and autonomously complete such an ambitious mission.

This paper focuses on the last two phases, namely the aerodynamic descent and the powered landing. It is well-known

that the non-powered, aerodynamically guided phase of the descent is particularly important for the management of the error in terms of position and velocity [9]. Moreover, this phase is characterized by large uncertainties due to both the atmosphere and the aerodynamic properties of the vehicle. Finally, there might be errors to be compensated for coming from previous phases of flight. All these aspects make the aerodynamic guidance a complex problem, which requires the capability to generate valid solutions rapidly, in reliable way, and in case significant off-nominal conditions are experienced during the mission, these must be taken into account. On the other hand the landing phase requires high accuracy and a perfect coordination of thrust, position, velocity and attitude to meet the strict requirements allowing for a safe and accurate touchdown, the so-called *pinpoint landing* [10].

Given the aforementioned reasons, the problem of generating valid guidance solutions in the frame of Entry, Descent, and Landing (EDL) has gained great attention, and multiple research groups and companies have worked on the subject. In many different solutions the key technology is represented by the use of Convex Optimization [11], a sub-branch of Optimization characterized by several benefits, including the guarantee to find a solution if there exists one, the reduced dependency on initial guesses, and the rapidness of computation, due to state-of-the-art interior point primal-dual solvers [12]. In the specific frame of EDL large attention was dedicated to the application of Second-Order Conic Programming (SOCP), a specific subset of convex optimization, in which all the inequality constraints are formulated in linear or conic form.

Among the proposed techniques there is the formulation of the entry problem in the energy domain [13]. In this case the non-convex constraints were transformed into upper and lower bounds on the altitude, by expressing the speed as a function of the energy. Moreover, to overcome the non-convexity intrinsically associated with the bank angle σ , two new controls, defined as the sine and the cosine of the bank angle, were adopted. The substitution was then made valid by ensuring that the identity $\sin^2 \sigma + \cos^2 \sigma = 1$ was satisfied. Partially inspired by the Space Shuttle Entry Guidance [14] a drag-energy approach based on the application of pseudospectral methods and convex optimization was developed in [15], where a valid drag-energy profile was computed by reformulating the problem in terms of inverse of drag acceleration, and the solution was mapped against longitudinal states by using an iterative approach. An interesting approach was also formulated by Wang and Grant by exploiting semidefinite programming [16] and second-order conic programming [17], respectively. In this case the problem was directly transcribed in the time domain by using a direct linearization approach of the nonlinear equations underlying the problem. Wang and Lu further improved the method [18] by means of line-search and trust region techniques that were introduced to speed up the convergence process. The previous approaches were mainly applied to VTHL (Vertical Take-off, Horizontal Landing) vehicles.

For what regards VTVL (Vertical Take-off, Vertical Landing) rockets, the landing phase was extensively treated in the last years, starting from 2007 [19], and this area is still very active now, with multiple applications of Convex Optimization [10], Successive Convex Optimization [20–22], and Pseudospectral Convex Optimization in its standard and generalized forms [23, 24]. Moreover, a first, successful attempt to combine aerodynamic and propulsive control was also proposed by Xinfu Liu [25], where the problem was reduced to two dimensions, and a new set of variables, needed to convexify the subproblem, was introduced. Yang and Liu also proposed to use altitude as independent variable to be able to deal with free-final time powered descent problems [26], through the corresponding manipulation of the equations of motion.

As pointed out by Yang and Liu this class of methods can be mainly divided into *Direct Linearization Approach* and *Nonlinearity-Kept and Linearization Approach*. In the former the equations of motion, as well as the constraints and the cost function are directly obtained by linearizing the problem around the solution found at the end of the previous iteration. The validity of the approach is ensured by an ad-hoc choice of the static trust regions. The weak point in this case might be a slow convergence, and further strategies might be needed to improve the quality of the process [16, 27]. In the latter strategy some variable transformations are introduced to try to keep in the limits of the constraints of the problem some features of the nonlinear formulation. The problems arise in this case because of the peculiarity of the transformations needed (which strongly depend on the nature of the problem), the assumptions required to ensure that some simplifications and transformations are valid, and the difficulty to generalize the method (for example it can be hard to move from the 2-D case to the general 3-D solution).

In this work we propose a third approach, which can be seen as a middle-ground between the two aforementioned methodologies. There are four main novelties associated with this work. First, we endorse an ad-hoc formulation of the equations of motion which minimizes the presence of non-convex terms. For example, by using Cartesian representation for position and velocity we avoid the trigonometric terms appearing in the equations of motion for longitude and latitude. This approach is to the best of our knowledge applied for the first time to the pure aerodynamic guidance problem of a VTVL vehicle.

Second, we introduce a different parametrization of controls, including Euler angle rates defined with respect to the

target-centered Altitude-Crossrange-Downrange directions, in addition to the thrust-rate already adopted in literature. This choice allows to maximize the presence the linear terms in our differential equations while having the controls appearing in affine form, a property which simplifies the convergence process, as pointed out by Liu et Al. [28]. Moreover the proposed approach gives us the chance to explicitly limit control rates as well, leading to smooth solutions, and therefore to trajectories which can be more easily tracked by the attitude controller.

The third aspect to be considered is that with the proposed approach the nonlinearities are entirely conveyed into the terms appearing in the differential equations representing the accelerations for the aerodynamic phase and the mass rate, and only for these terms numerical differentiation is employed, leading to an hybrid computation of the matrix representing the equation of motion.

The fourth and last aspect is the introduction of a systematic transcription based on generalized hp pseudospectral methods, already adopted for the powered landing problem [24], but here extended to deal with the aerodynamic phase problem too. This choice benefits from the properties of pseudospectral methods [29], such as the quasi-exponential (or *spectral*) convergence, and the easiness of implementation. Moreover, its unique mapping is exploited to formulate the free-final time problem, leading to an increased capability of handling initial dispersions, since we don't need any a-priori knowledge or estimate of the flight time for off-nominal cases.

It is worth mentioning that in related works it is reasonable to consider some simplifications for what regards the aerodynamic accelerations (e.g., constant drag coefficient) given their relative importance. In this work we attempt instead to test the method in conditions which are as close as possible to what the vehicle in a real scenario will experience. Therefore we reject simplifications typically used for this class of methods, (e.g., constant gravity, no lift, or constant drag coefficient). Instead, we use a full-blown aerodynamic database, which includes lift, drag, side-force, as well as aerodynamic torques, which depend on the 3-D attitude of the vehicle as well as on the Mach number [30]. These assumptions are required given the centrality of the aerodynamic accelerations for this type of problems.

Numerical results are shown for a CALLISTO-class rocket. The paper is organized as follows: in Sec. II the mission and the vehicle are briefly described, while Sec. III focuses on the problem formulation in continuous form. Sections IV and V describe the convexified approach and its corresponding pseudospectral transcription, respectively, while numerical results are described in Sec. VI. Finally, we draw some conclusions and describe the future outlook for this work in Sec. VII.

II. Mission and Vehicle

The vehicle considered in this work is a 40-kN class rocket. Its thrust can be reduced up to 40%, and is equipped with four steerable fins mounted on top, able to provide full aerodynamic control for the attitude during the phase of shut-down of the engine, while during the ascent and the landing phases the engine, mounted on a gimbaled system, is able to provide thrust vector control capability around pitch and yaw axes. A reaction control system, made by eight thrusters, is included to generate roll control. The set of actuators is able to provide full control capability during every phase of flight. An illustration of the vehicle used as example is depicted in Fig. 1(a).

Due to the mass consumption, the vehicle experiences during the mission a significant variation of both the center of mass and the inertia matrix. In this work they are considered constant during the aerodynamic descent, and time-varying parameters in the powered phase, with their value that is stored as lookup-table depending on the current mass. The axis-symmetry of the vehicle has been exploited to compute aerodynamic coefficients as function of an horizontal and a vertical angle of attack α_1 and α_2 , as illustrated in Fig. 1(c).

The aerodynamic force coefficients with respect to the body axes \mathbf{C}_{BODY}^{aero} are provided as multidimensional look-up tables, which depend on Mach number M , and angles of attack α_1 and α_2 , and on the fin deflections $\delta_1, \delta_2, \delta_3, \delta_4$.

$$\mathbf{C}_{BODY}^{aero} = \begin{bmatrix} C_x(M, \alpha_1, \alpha_2, \delta_1, \delta_2, \delta_3, \delta_4) \\ C_y(M, \alpha_1, \alpha_2, \delta_1, \delta_2, \delta_3, \delta_4) \\ C_z(M, \alpha_1, \alpha_2, \delta_1, \delta_2, \delta_3, \delta_4) \end{bmatrix} \quad (1)$$

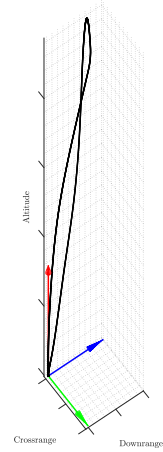
From these coefficients the aerodynamic force can be computed as

$$\mathbf{F}_{BODY,i}^{aero} = \frac{1}{2} \rho V^2 S \mathbf{C}_{BODY,i}^{aero} \quad , \quad i = x, y, z \quad (2)$$

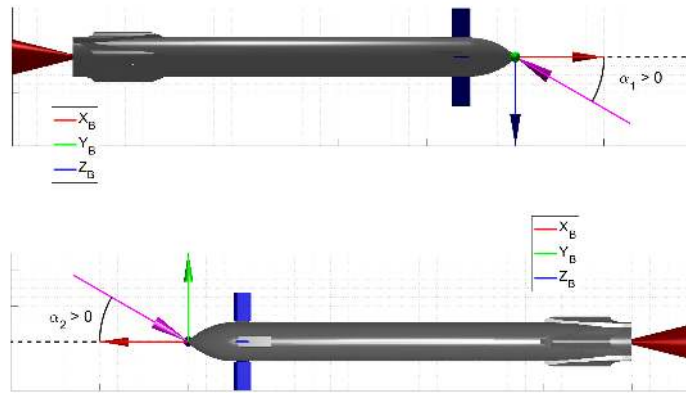
with the reference surface S equal to 0.95 m, and ρ representing the atmospheric density, which depends on altitude, and V is the speed of the vehicle with respect to the air. Throughout this work we use a real atmospheric model coming from



(a) CALLISTO experimental vehicle



(b) Return-to-Launch-Site mission profile



(c) Vertical and horizontal angles of attack.

Fig. 1 Mission and vehicle overview: (a) CALLISTO rocket, (b) Reference mission profile, (c) Longitudinal and lateral angles of attack.

experimental measures, and provided as look-up tables, where the geodetic altitude is the independent variable. This choice confirms once more that no strong simplifications are considered for the proposed method.

Remark 1: Note that, although this solution is inherently based on a 3-DOF model we are interested to generate solutions which we define *6-DOF capable*, which means the the generated trajectory has to provide an attitude which can be trimmed by the aerodynamic fins. This aspect is currently included in the present work, as it will be shown in Sec. VI.

The flight will take place in 2022 at the Guiana Space Center, the European Spaceport in French Guiana, and the mission will foresee a sequence of flights following the Return-to-Launch-Site (RTLS) profile, depicted in Fig. 1(b). This sequence will give indications on the level of refurbishment required between two consecutive flights performed with the same vehicle, and provide first-hand data to all the partners to further enhance the knowledge of reusable technologies and some of the related critical technologies, especially in terms of Guidance, Navigation and Control.

III. Problem Formulation

In this section we describe in detail the problem formulation for both the aerodynamic descent and the powered landing phases of the flight.

Aerodynamic Descent

A. Equations of Motion

The problem of the aerodynamic guidance is described in a target-centered Downrange-Crossrange-Altitude (DCA) reference frame as depicted in Fig. 1(b). Since no thrust is available during this phase only the aerodynamic forces can be used to control the vehicle. The forces are determined by the relative attitude of the body axes with respect to the airflow. Therefore the attitude represents the main way to control the system, which is modeled as a 3-DOF point mass. The desired attitude will then define the reference signals to be tracked by the attitude controller when it is time to include a feedback law, which is realized by using the fins to generate the desired torques. The translational dynamics can be therefore described by using the following equations of motion,

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a}^{grav} + \mathbf{a}^{aero} - 2\boldsymbol{\omega} \times \mathbf{v} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \\ \dot{\theta} &= u_\theta \\ \dot{\psi} &= u_\psi\end{aligned}\tag{3}$$

where \mathbf{r} is the position of the center of mass of the vehicle expressed in DCA reference frame, and \mathbf{v} the corresponding velocity. Note that the reference frame indication is omitted for brevity. Non-inertial terms due to the rotation of the Earth $\boldsymbol{\omega}$ are also considered, while θ and ψ are the pitch and yaw angle of the vehicle with respect to the target-centered Downrange-Crossrange-Altitude reference frame. The roll angle is kept constant during the descent to maximize the decoupling between pitch and yaw axes. Note that the controls we effectively use are the pitch rate u_θ and the yaw rate u_ψ . This choice is twofold beneficial: first, it decouples the control matrix from the states, as we will see. Second, it allows to impose explicit bounds on the control rates, making the solution smoother. The sources of non-convexity are in this case two: the gravity, which non-linearly depends on position, and the aerodynamic accelerations, which are nonlinear functions of velocity, altitude, and attitude.

For the gravity we can assume a central-body model.

$$\mathbf{a}^{grav} = -\mu_\oplus \frac{\mathbf{r} + \mathbf{r}_{T,DCA}}{\|\mathbf{r} + \mathbf{r}_{T,DCA}\|_2^3}\tag{4}$$

with μ_\oplus representing the gravitational parameter of the Earth, while \mathbf{r}_T is the position vector of the target site with respect to the center of the Earth. Note that with this formulation more accurate models, like the one based on the World Geodetic System 84 [31], could be adopted. However, this more advanced modeling is kept for future development, and a simpler choice was here preferred. For what regards the aerodynamic accelerations they represent a nonlinear combination of the states. In fact, the aerodynamic accelerations are, for the rocket under analysis, expressed in body-reference frame as

$$\mathbf{a}_{BODY}^{aero} = \frac{1}{m} \mathbf{F}_{BODY}^{aero}(\alpha_1, \alpha_2, M, q_{dyn})\tag{5}$$

where m is the mass of the vehicle, whereas α_1 and α_2 are the vertical and horizontal angle of attack (introduced in Fig. 1(c) and used to exploit the axis-symmetry of the vehicle). M represents the Mach number, while q_{dyn} is the dynamic pressure, function of the altitude (through the atmospheric density ρ) and the velocity magnitude V .

$$q_{dyn} = \frac{1}{2}\rho V^2 \quad (6)$$

Since we are interested to express the aerodynamic accelerations in DCA coordinates it is necessary to transform the outcome of Eq. (5) as

$$\mathbf{a}_{DCA}^{aero} = \mathbf{R}_{BODY}^{DCA} \cdot \mathbf{a}_{BODY}^{aero} \quad (7)$$

where \mathbf{R}_{BODY}^{DCA} represents the rotation matrix from $BODY$ to DCA . This matrix is composed by two different contributions.

$$\mathbf{R}_{BODY}^{DCA} = \mathbf{R}_{UEN}^{DCA} \cdot \mathbf{R}_{BODY}^{UEN} \quad (8)$$

with UEN representing the target-fixed Up-East-North reference frame. Note that the first term of the right hand side of Eq. (8) is only function of the target position $\mathbf{r}_{T,DCA}$

$$\mathbf{R}_{UEN}^{DCA} = \mathbf{R}_{UEN}^{DCA}(\mathbf{r}_{T,DCA}) \quad (9)$$

and is constant, while the second contribution is a direct function of the attitude of the body. In fact, we can write

$$\mathbf{R}_{BODY}^{UEN} = \mathbf{R}_{BODY}^{UEN}(\theta, \psi) \quad (10)$$

The derivation of the aerodynamic accelerations in DCA is not only necessary for the formulation of the equations of motion, but is also useful because it gives us an indication of the dependencies to be considered when the linearized equations of motion are derived.

B. Boundary Conditions

The problem will have fixed initial and final conditions, coming from previous and successive phases of flight,

$$\begin{bmatrix} \mathbf{r}_{DCA}(t_0) \\ \mathbf{v}_{DCA}(t_0) \\ \theta_{DCA}(t_0) \\ \psi_{DCA}(t_0) \\ \mathbf{r}_{DCA}(t_F) \\ \mathbf{v}_{DCA}(t_F) \\ \theta_{DCA}(t_F) \\ \psi_{DCA}(t_F) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{DCA,0} \\ \mathbf{v}_{DCA,0} \\ \theta_{DCA,0} \\ \psi_{DCA,0} \\ \mathbf{r}_{DCA,F} \\ \mathbf{v}_{DCA,F} \\ \theta_{DCA,F} \\ \psi_{DCA,F} \end{bmatrix} \quad (11)$$

as we want, at the end of the aerodynamic descent (happening at the free time t_F) to be in conditions of correctly switching to the powered phase in optimal conditions for the pinpoint landing.

Remark 2: Note that the Euler angles adopted throughout this work are built on the Up-East-North convention, and not on the traditional North-East-Down. This choice is motivated by the need to avoid the classical singularity of the pitch angle at 90 deg, which is what would happen for a vertical descending vehicle. With the adopted convention we ensure to be away from the singularity, as the vertical descent is associated with a pitch angle $\theta = 0$ deg.

C. Constraints

As previously mentioned, to ensure the structural integrity of the vehicle during the flight we might want to impose further constraints in our problem, i.e., an upper bound on the dynamic pressure or on the load factor that can be tolerated by the vehicle's structure.

$$\begin{bmatrix} q_{dyn} \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho V^2 \\ \frac{\|\mathbf{F}_n\|_2}{mg_\oplus} \end{bmatrix} \leq \begin{bmatrix} q_{dyn,max} \\ n_{z,max} \end{bmatrix} \quad (12)$$

The inequality sign in Eq. (12) meant as component-wise, and \mathbf{F}_n represents the normal component of the aerodynamic force expressed in the body reference frame as defined in Eq. 5. m is the mass of the vehicle during the descent phase, and g_\oplus is the gravity acceleration at sea level, assumed equal to 9.8067 m/s^2 .

D. States and controls bounds

Finally, we want to have a meaningful upper bound and lower bound on the states,

$$\begin{bmatrix} r_{xL} \\ r_{yL} \\ r_{zL} \\ v_{xL} \\ v_{yL} \\ v_{zL} \\ \theta_L \\ \psi_L \end{bmatrix} \leq \begin{bmatrix} r_x \\ r_y \\ r_z \\ v_x \\ v_y \\ v_z \\ \theta \\ \psi \end{bmatrix} \leq \begin{bmatrix} r_{xU} \\ r_{yU} \\ r_{zU} \\ v_{xU} \\ v_{yU} \\ v_{zU} \\ \theta_U \\ \psi_U \end{bmatrix} \quad (13)$$

which derive mainly from flight-safety studies. Moreover, to limit the closed-loop bandwidth associated with the attitude controller, and generate a smooth solution, upper and lower bounds are also assigned to the rates of pitch and yaw angles, with $u_{\theta,max} = u_{\psi,max} = 2 \text{ deg/s}$.

$$\begin{bmatrix} -u_{\theta,max} \\ -u_{\psi,max} \end{bmatrix} \leq \begin{bmatrix} u_\theta \\ u_\psi \end{bmatrix} \leq \begin{bmatrix} u_{\theta,max} \\ u_{\psi,max} \end{bmatrix} \quad (14)$$

E. Cost function

Finally, for this problem we are interested to minimize the control activity, therefore we simply express the cost function as

$$J = w_u \int_{t_0}^{t_F} \left[u_\theta^2(t) + u_\psi^2(t) \right] dt \quad (15)$$

where w_u is a user-defined positive weight, whose value is formally irrelevant in Eq. (15), but becomes important in the construction of an augmented cost function that takes also other effects into account, as it will be shown in Secs. IV and V.

The problem to be solved is therefore the following: we aim at minimizing Eq. (15) with the system subject to the differential equations defined in Eq. (3). The solution has to satisfy the boundary conditions given by Eqs. (11), as well as the constraints of Eq. (12). Finally, states and control are bounded according to Eqs. (13) and (14).

Powered Landing

A. Equations of motion

We can extend the previous formulation to the powered landing problem. Several elegant formulations have been proposed over the years to deal with the problem of powered landing. Here we aim at including the presence of realistic effects, which in order of relevance are 1) the thrust-aerodynamic forces interaction, 2) the minimization of aerodynamic torques that could prevent the 6-DOF feasibility of the trajectory, 3) the motion of the center of mass while descending, 4) the effect of the pressure on the effective thrust generated, and 5) other effects, like non-constant gravitational acceleration and non-inertial forces due to the rotation of the Earth.

While in the previous subsection we were using the aerodynamic forces as means of control in this case their effect is combined with the force exerted by the engine to dominate the motion of the rocket. The corresponding model is a

3-DOF point having variable mass, and its evolution is described by the following set of equations:

$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathbf{a}^{thr} + \mathbf{a}^{grav} + \mathbf{a}^{aero} - 2\boldsymbol{\omega} \times \mathbf{v} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \\
\dot{m} &= -\frac{T_{vac}}{I_{sp}g_0} \\
\dot{\phi} &= u_\phi \\
\dot{\theta} &= u_\theta \\
\dot{\psi} &= u_\psi \\
\dot{T}_{atmo} &= u_T
\end{aligned} \tag{16}$$

Note the presence of the roll angle ϕ , the mass m and the atmospheric thrust T_{atmo} , with the last two terms linked to the vacuum thrust T_{vac} through the equation

$$T_{atmo} = T_{vac} - A_{nz}p \tag{17}$$

with A_{nz} indicating the nozzle area, p the atmospheric pressure, and T_{vac} the thrust generated in vacuum. The controls are for this scenario the Euler angle rates u_ϕ, u_θ, u_ψ . Moreover, to limit the instantaneous change of thrust, not compatible with the physical rocket engine, we include the thrust rate u_T . This choice gives the chance to decouple the control matrix from the states as done in for the aerodynamic descent, and at the same time to obtain solution physically realizable by the rocket's actuators. For this problem there are four different sources of non-convexity: in addition to the the aforementioned gravity and aerodynamic accelerations we have now the acceleration caused by the thrust $\mathbf{a}_{thr} = \mathbf{T}_{thr}/m$. Moreover, there is an exponential dependence on the massflow from the altitude through the pressure in virtue of Eq. (17).

The aerodynamic accelerations can be computed again by invoking Eq. (5) and we can convert them into their DCA representation exactly as done through Eq. (7)-(9). The only difference is represented by the modification of Eq. (10), which, in virtue of the dependence on the roll angle ϕ becomes

$$\mathbf{R}_{BODY}^{UEN} = \mathbf{R}_{BODY}^{UEN}(\phi, \theta, \psi) \tag{18}$$

B. Boundary Conditions

The boundary conditions described in Eqs. (11) are still valid. We augment them with some further conditions coming from the new variables included in the problem

$$\begin{bmatrix} m(t_0) \\ \phi_{DCA}(t_0) \\ T(t_0) \\ \phi_{DCA}(t_F) \end{bmatrix} = \begin{bmatrix} m_0 \\ \phi_{DCA,0} \\ T_0 \\ \phi_{DCA,F} \end{bmatrix} \tag{19}$$

Note that we omitted the final value of mass and thrust, as they are determined by the algorithm. Moreover, we include initial and final conditions for the attitude to ensure that the vehicle lands with its x -body axis being normal to the local horizontal plane.

C. Constraints

Three types of constraints are included here: first, we introduce the classical glideslope constraint to enforce the vehicle to impose a controlled ratio between reduction of horizontal and vertical distance with respect to the landing spot.

$$\frac{r_A}{\|\mathbf{r}_{D,C}\|} \geq \tan \gamma_{gs} \tag{20}$$

with glideslope angle equal to 70 deg. To further enforce a vertical motion towards the end of the pinpoint landing sequence it is imposed that in the last segment of the trajectory, approximately corresponding to the last 5 seconds of flight, both side-components of position and velocity are bounded, and specifically

$$\begin{aligned}
\|\mathbf{r}_{D,C}(t \leq t_F - t^*)\| &\leq r_{D,C,max} \\
\|\mathbf{v}_{D,C}(t \leq t_F - t^*)\| &\leq v_{D,C,max}
\end{aligned} \tag{21}$$

with t^* equal to 5 s, $r_{D,C,max}$ equal to 1 m, and $v_{D,C,max}$ defined as 0.1 m/s.

Note that all these constraints can be modeled as second-order conic constraints, and therefore do not require linearization.

D. States and controls bounds

While Eqs. (13)-(14) still hold for the powered phase too, we augment them according to the problem definition by adding the following box constraints for the states,

$$\begin{bmatrix} m_L \\ \phi_L \\ T_L \end{bmatrix} \leq \begin{bmatrix} m \\ \phi \\ T \end{bmatrix} \leq \begin{bmatrix} m_u \\ \phi_U \\ T_U \end{bmatrix} \quad (22)$$

and the controls

$$\begin{bmatrix} -u_{\phi,max} \\ -u_{T,max} \end{bmatrix} \leq \begin{bmatrix} u_{\phi} \\ u_T \end{bmatrix} \leq \begin{bmatrix} u_{\phi,max} \\ u_{T,max} \end{bmatrix} \quad (23)$$

with $u_{\theta,max} = u_{\psi,max} = 5$ deg/s, while the roll rate is limited to 0.1 deg/s. Finally, a scaled maximum thrust rate $u_{T,max}$ equal to 14.55 is included in the formulation.

E. Cost function

The cost function we build for this problem is made of different contributions: first, we are interested to maximize the final mass of the vehicle, which corresponds to the minimization of the fuel required to perform the landing maneuver. Moreover, we introduce a penalization of the control rates to ensure that the solution we obtain is smooth enough.

$$J = -w_m m(t_F) + w_u \int_{t_0}^{t_F} (\mathbf{u}^T \cdot \mathbf{R} \cdot \mathbf{u}) dt \quad (24)$$

The vector \mathbf{u} embeds all the four controls included in the formulation through \mathbf{R} , defined as a unitary diagonal matrix, while the terms w_m and w_u measure the relative importance of the two terms in the optimization process, and are assumed equal to 100 and 10, respectively. This choice is motivated by the fact that while we are interested to optimize the fuel consumption, we also want to discourage through the presence of the term w_u large variations of the angular rates. In fact larger control variations could lead to hectic control profiles, which might be slightly more efficient from the fuel-consumption perspective, but less safe. We have completely defined the problem to be solved: we aim at minimizing Eq. (24) with the system subject to the differential equations defined in Eq. (16). The solution has to satisfy the boundary conditions given by Eqs. (11),(19) and (13)-(14),(22)-(23) as well as the constraints of Eqs. (20) and (21).

IV. Convex Formulation

In this section we will transform the two continuous problems described in Sec. III into a sequence of convex problems, to be solved iteratively.

Aerodynamic Descent

A. Equations of motion

For what regards the equations of motion during the aerodynamic phase we can decompose the system described in Eq. (3) in a convex part, and a non-convex part. Defined the state vector as

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{DCA} & \mathbf{v}_{DCA} & \theta & \psi \end{bmatrix}^T \quad (25)$$

we can write the equations of motion as

$$\dot{\mathbf{x}} = \mathbf{f}_{nc}(\mathbf{x}) + \mathbf{f}_c(\mathbf{x}) + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{v} \quad (26)$$

with

$$\mathbf{u} = \begin{bmatrix} u_{\theta} & u_{\psi} \end{bmatrix}^T \quad (27)$$

representing the physical controls used to manipulate the attitude of the vehicle, and consequently, the aerodynamic forces generated, while the vector $\boldsymbol{v} \in \mathbb{R}^{n_s}$, defined as

$$\boldsymbol{v} = \begin{bmatrix} v_{r_x} & v_{r_y} & v_{r_z} & v_{v_x} & v_{v_y} & v_{v_z} & v_\theta & v_\psi \end{bmatrix}^T \quad (28)$$

represents the virtual controls, required to avoid artificial infeasibility [20]. The matrix \mathbf{C} is a design parameter to decide which and how many virtual controls will be used to help the convergence process. For this work the matrix is defined as

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{6 \times 6} & \mathbf{O}_{6 \times 2} \\ \mathbf{O}_{2 \times 6} & \mathbf{O}_{2 \times 2} \end{bmatrix} \quad (29)$$

which implies that virtual controls are only applied to the translational states, acting as synthetic accelerations and velocities affecting the differential equations of \boldsymbol{v} and \boldsymbol{r} , respectively. This choice is due to the nature of the problem, given that including virtual controls affecting the attitude states would not provide any physical improvement to the convergence process.

For what regards the convex terms, they are represented by

$$\mathbf{f}_c(\mathbf{x}) = \begin{bmatrix} \mathbf{v}_{DCA} \\ -2\boldsymbol{\omega} \times \mathbf{v}_{DCA} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{DCA}) \\ \mathbf{O}_{2 \times 1} \end{bmatrix} = \mathbf{A}_c \cdot \mathbf{x} \quad (30)$$

with

$$\mathbf{A}_c \triangleq \begin{bmatrix} \mathbf{O}_{[3 \times 3]} & \mathbf{I}_{[3 \times 3]} & \mathbf{O}_{[3 \times 2]} \\ -\boldsymbol{\omega} \times (\boldsymbol{\omega} \times) & -2\boldsymbol{\omega} \times & \mathbf{O}_{[3 \times 2]} \\ \mathbf{O}_{[3 \times 3]} & \mathbf{O}_{[3 \times 3]} & \mathbf{O}_{[3 \times 2]} \end{bmatrix} \quad (31)$$

The matrix \mathbf{A}_c only contains constant terms, and is therefore computed only once during the initialization of the algorithm, and kept constant through the process.

The non-convex term can be convoyed into the fourth, fifth and sixth elements of the vector

$$\mathbf{f}_{nc}(\mathbf{x}) = \begin{bmatrix} \mathbf{O}_{3 \times 1} \\ \mathbf{a}_{DCA}^{grav} + \mathbf{a}_{DCA}^{aero} \\ \mathbf{O}_{2 \times 1} \end{bmatrix} \quad (32)$$

and this contribution represents the only term that requires linearization. Finally, the control matrix \mathbf{B} is

$$\mathbf{B} = \begin{bmatrix} \mathbf{O}_{6 \times 2} \\ \mathbf{I}_{2 \times 2} \end{bmatrix} \quad (33)$$

We can see that the system is affine in control, which is a very important property of the problem to be iteratively solved by using sequential convex programming, as demonstrated by Liu et Al. [28]. Moreover, the structure chosen to represent the problem suggests us that we can apply a partial linearization and perform sequential convex programming by exploiting the distinction between convex and non-convex terms. On this purpose, suppose we have solved the problem k times, with $k = 0, \dots, k_{max}$. The solution with $k = 0$ can either be a propagation of dummy controls, or a linear interpolation between initial and final states and controls. To solve the $(k + 1)^{th}$ sub-problem we linearize the nonlinear terms of equations of motion around the sub-solution k . The subscript k will indicate the terms computed by using the corresponding k^{th} solution. We can therefore rewrite the system described in Eq. (26) as

$$\dot{\mathbf{x}} = \mathbf{A}_c \mathbf{x} + \mathbf{A}_k \mathbf{x} + \mathbf{B} \mathbf{u} + \mathbf{C} \boldsymbol{v} + \mathbf{G}_k \quad (34)$$

where

$$\mathbf{A}_k \triangleq \left. \frac{\partial \mathbf{f}_{nc}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_k} \quad (35)$$

and

$$\mathbf{G}_k \triangleq \mathbf{f}_{nc}(\mathbf{x}_k) - \mathbf{A}_k \mathbf{x}_k \quad (36)$$

Note that, as highlighted by multiple authors [18, 20, 32] it is necessary that the new solution does not largely differ from the previous one. This condition is needed to ensure that the nonlinear behavior of the system is well captured by the first two terms of the Taylor expansions underlying the linearization. To have a meaningful linearization process trust region constraints are adopted. The way to implement trust region constraints has been widely treated in literature in multiple forms. Some researchers prefer to express the trust region radius as a user-defined vector [25, 28]. This approach has the advantage to reduce the size of the problem, since the trust region size is an input to the subproblem to be solved, rather than a variable to be optimized. Other relevant works include update rules for shrinking or enlarging their size depending on some metrics measuring the validity of the linearization at each iteration [33, 34]. Finally, a further approach consists in introducing dynamic upper bounds for the trust region as part of the subproblem formulation [21]. In a similar fashion to this last approach we introduce trust region upper bounds on the difference between the new solution and the previous iteration, used to build the current subproblem to be solved.

$$\|\tilde{\mathbf{X}}(t) - \tilde{\mathbf{X}}_k(t)\| \leq \zeta(t) \quad (37)$$

with

$$\tilde{\mathbf{X}} \triangleq \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}, \quad \tilde{\mathbf{X}}_k \triangleq \begin{bmatrix} \mathbf{x}_k(t) \\ \mathbf{u}_k(t) \end{bmatrix} \quad (38)$$

and ζ representing an upper bound that limits the excursion between two consecutive iterations, to be penalized as well through a corresponding slack variable

$$\|\zeta\|_2 \leq s_\zeta, \quad s_\zeta \in \mathbb{R} \quad (39)$$

Note that the problem must be scaled in order to have the construction of the norm in Eq. (37) to be a legitimate operation.

B. Constraints

A linearization is also required to handle the constraints of Eq. (12), which are reformulated as

$$\begin{bmatrix} q_{dyn} \\ n_z \end{bmatrix} \Big|_{\mathbf{x}=\mathbf{x}_k} + \begin{bmatrix} \mathbf{G}_{q_{dyn}} \\ \mathbf{G}_{n_z} \end{bmatrix} (\mathbf{x} - \mathbf{x}_k) \leq \begin{bmatrix} q_{dyn,U} \\ n_{z,U} \end{bmatrix} \quad (40)$$

with the matrices $\mathbf{G}_{q_{dyn}}$ and \mathbf{G}_{n_z} computed with a numerical approach, in virtue of the presence of Eq. (2) appearing in the last of the terms in Eq. (12), and the presence of a look-up table-based atmospheric model. Upper and lower bounds on states and controls expressed by Eqs. (13)-(14) can be easily expressed as linear inequalities. Finally, the last constraint to be considered is represented by the virtual controls. They are only used to avoid *artificial obstructions*, and need to be reduced to a negligible value when the process tends towards its final converged solution. For this reason, as already proposed in literature ([20]) every virtual control vector is bounded by a corresponding slack variable η_v

$$\|v\|_2 \leq \eta_v, \quad \eta_v \in \mathbb{R}^{n_p} \quad (41)$$

and to ensure that the virtual controls are removed over the process, a further slack variable is introduced as upper bound for the norm of η_v :

$$\|\eta_v\|_2 \leq s_\eta \quad (42)$$

with the term s_η included in the cost function, and scaled by a positive value w_η .

C. Augmented Cost function

To include the penalization of trust regions and virtual controls in the formulation the augmented cost function for the subproblem is defined as

$$J_{aug}(\mathbf{x}, \mathbf{u}, v, \zeta, s_\eta, s_\zeta) = w_u J + w_\eta \cdot s_\eta + w_\zeta \cdot s_\zeta \quad (43)$$

with the weights w_η and w_ζ measuring the relative importance of the penalization of virtual control and trust region with respect to the true cost function J defined in Eq. (15), which is weighted by w_u .

In conclusions during the aerodynamic phase we are interested to optimize at each iteration Eq. (43), subject to Eqs. (34) and (40) while ensuring proper penalization of both the trust region size through Eqs. (37) and (39), and a shrinkage of virtual controls through Eqs. (41) and (42).

Powered Landing

A. Equations of motion

By extending the logic of the previous section, we expand the state vector for the powered landing phase in the following manner,

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{DCA} & \mathbf{v}_{DCA} & m & \phi & \theta & \psi & T \end{bmatrix}^T \quad (44)$$

with the corresponding dynamics that in vector-form remains the same as Eq. (26) but where the control vector is now the following.

$$\mathbf{u} = \begin{bmatrix} u_\phi & u_\theta & u_\psi & u_T \end{bmatrix}^T \quad (45)$$

The virtual control vector $\mathbf{v} \in \mathbb{R}^{n_s}$, is in this case defined as

$$\mathbf{v} = \begin{bmatrix} v_{rx} & v_{ry} & v_{rz} & v_{vx} & v_{vy} & v_{vz} & v_m & v_\phi & v_\theta & v_\psi & v_T \end{bmatrix}^T \quad (46)$$

and the matrix \mathbf{C} defined in Eq. (29) is augmented accordingly,

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{6 \times 6} & \mathbf{O}_{6 \times 5} \\ \mathbf{O}_{5 \times 6} & \mathbf{O}_{5 \times 5} \end{bmatrix} \quad (47)$$

such that also in this case virtual controls affect the translational motion. The convex terms are common to those defined during the aerodynamic descent, modified only to take the different size of the state vector into account.

$$\mathbf{f}_c(\mathbf{x}) = \begin{bmatrix} \mathbf{v}_{DCA} \\ -2\boldsymbol{\omega} \times \mathbf{v}_{DCA} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{DCA}) \\ \mathbf{O}_{5 \times 1} \end{bmatrix} = \mathbf{A}_c \cdot \mathbf{x} \quad (48)$$

with

$$\mathbf{A}_c \triangleq \begin{bmatrix} \mathbf{O}_{[3 \times 3]} & \mathbf{I}_{[3 \times 3]} & \mathbf{O}_{[3 \times 5]} \\ -\boldsymbol{\omega} \times (\boldsymbol{\omega} \times) & -2\boldsymbol{\omega} \times & \mathbf{O}_{[3 \times 5]} \\ \mathbf{O}_{[5 \times 3]} & \mathbf{O}_{[5 \times 3]} & \mathbf{O}_{[5 \times 5]} \end{bmatrix} \quad (49)$$

The non-convex terms are grouped into the fourth, fifth, sixth and seventh differential equations coming from Eq. (16).

$$\mathbf{f}_{nc}(\mathbf{x}) = \begin{bmatrix} \mathbf{O}_{3 \times 1} \\ \mathbf{a}_{DCA}^{thr} + \mathbf{a}_{DCA}^{grav} + \mathbf{a}_{DCA}^{aero} \\ -\frac{T_{vac}}{I_{sp}g_0} \\ \mathbf{O}_{4 \times 1} \end{bmatrix} \quad (50)$$

As previously done, numerical linearization is applied to these terms only. Finally, the control matrix \mathbf{B} is

$$\mathbf{B} = \begin{bmatrix} \mathbf{O}_{7 \times 4} \\ \mathbf{I}_{4 \times 4} \end{bmatrix} \quad (51)$$

The procedure is therefore exactly the same as the one highlighted in the previous section. We apply to this augmented formulation Eqs. (34)-(39),(41),(42) at each iteration. The only differences reside in the size of states and controls, and in the corresponding non-convex contributions. Moreover, we have different constraints and cost function, described in the next subsections.

B. Constraints

As aforementioned the constraints included in this work, and described by Eqs. (20) and (21) can be exactly implemented as Second-order conic constraints. For the glide-slope constraint we impose

$$\|\mathbf{A}_{gs}\mathbf{r} + \mathbf{b}_{gs}\| \leq \mathbf{c}_{gs}\mathbf{r} + d_{gs} \quad (52)$$

with

$$\mathbf{A}_{gs} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b}_{gs} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (53)$$

$$\mathbf{c}_{gs} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad d_{gs} = 0$$

For the limitations of horizontal position and velocity at the end of the landing phase, we can derive similar expressions:

$$\begin{aligned} \|\mathbf{A}_r\mathbf{r} + \mathbf{b}_r\| &\leq \mathbf{c}_r\mathbf{r} + d_r \\ \|\mathbf{A}_v\mathbf{v} + \mathbf{b}_v\| &\leq \mathbf{c}_v\mathbf{v} + d_v \end{aligned} \quad (54)$$

where the corresponding matrices are defined as

$$\begin{aligned} \mathbf{A}_r = \mathbf{A}_v &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b}_r = \mathbf{b}_v = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ \mathbf{c}_r = \mathbf{c}_v &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad d_r = 1, \quad d_v = 0.1 \end{aligned} \quad (55)$$

C. Augmented Cost function

The augmented cost is formally the same as Eq. (43). However some practical terms will differ due to the application since the true cost J , defined respectively by Eq. (15) and (24) are clearly distinct.

To summarize the landing convexified problem we want to minimize at each iteration Eq. (43). The solution must satisfy Eqs. (34), (20), and (21) while ensuring proper penalization of both the trust region size through Eqs. (37) and (39), and of the virtual controls through Eqs. (41) and (42).

In the next section we will transcribe the problem through the use of hp generalized pseudospectral methods.

V. Sequential Pseudospectral Convex Programming

The transcription proposed here is conceptually common to both the aerodynamic descent and the powered landing problems, and therefore we will first focus on the aspects common to both, while emphasizing later the differences between the two specific problems, especially in terms of constraints, cost function, and initialization strategy.

More specifically, we propose to adopt an hp generalized pseudospectral transcription based on the use of flipped Legendre-Gauss-Radau (fLGR) method. The proposed method represents a valid alternative to the more traditional Euler and trapezoidal discretization schemes, given its higher accuracy. Moreover, it was shown that the larger CPU time can be mitigated by adopting its hp variant [24]. In the remainder of this section we will identify the steps of transcription according to the Sequential Pseudospectral Convex Programming (SPCP) method here proposed.

A. Discretization

Inspired by the good results obtained in our previous works [23, 24] we extend the methodology to the problem formulated in Sec. IV. Specifically, we propose to use n segments, and in each of them perform a local collocation using $p + 1$ nodes coming from the p roots of the corresponding fLGR polynomial, defined in the domain $(-1,1]$, and initial non-collocated node at $\tau = -1$. A visualization of the domain is visible in Fig. 2.

Note that since the domain is broken into segments, some linking conditions connecting them are needed. They will explicitly be defined in this section, and form, together with the equations of motion and the boundary conditions the set of linear equations underlying the transcription.

Another benefit is associated with the possibility to have an open final-time formulation of the guidance problem. In fact, to come up with a free final-time discretization some researchers prefer to reformulate the problem by using a different independent variable, known to be monotonically changing, and with known initial and final values [15, 26].

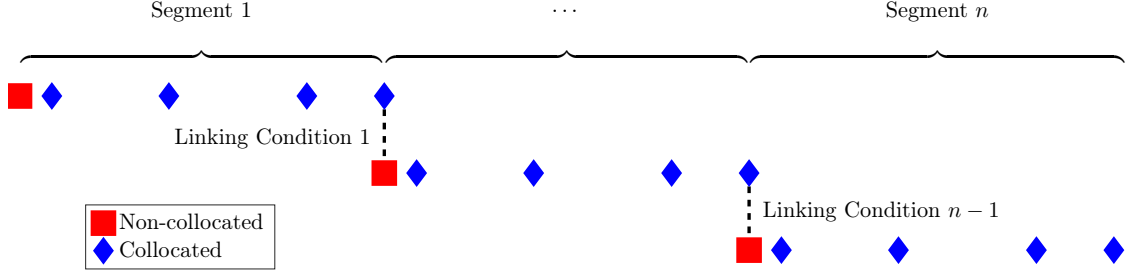


Fig. 2 Domain of the hp flipped Legendre-Gauss-Radau method.

A different approach was the use of a stretching term $\hat{\sigma}$ in the equations of motion [22]. This term can be in fact re-interpreted as the typical mapping between physical time and pseudospectral time k_t , defined as

$$k_t \triangleq \frac{t_F - t_0}{2n} \quad (56)$$

with t_0 and t_F initial and final times of the physical problem to be solved, and n the number of phases to be used in the hp framework we are proposing. This observation helps in rewriting the transcription in presence of free final time as follows. We assume to have the state vector $\mathbf{x} \in \mathbb{R}^{n_s}$, the control vector $\mathbf{u} \in \mathbb{R}^{n_c}$, the virtual control vector $\mathbf{v} \in \mathbb{R}^{n_v}$, and a total of $n(p+1)$ discrete time steps, corresponding to the n segments and the $p+1$ discrete points in each of the segments. Computed the $p+1$ nodes τ_i corresponding to a single segment and defined between -1 and 1 we can, for every couple t_0 , and t_F , identify the single i^{th} discrete timestep associated with the j^{th} segment as

$$t_i^j = t_0 + \frac{t_F - t_0}{n} \left(j - \frac{1}{2} \right) + \frac{t_F - t_0}{2n} \tau_i, \quad j = 1, \dots, n-1 \quad (57)$$

Defined the discrete time domain, we can introduce the augmented discrete decision vector $\mathbf{X} \in \mathbb{R}^{n_{var}}$ as

$$\mathbf{X} = \left[\mathbf{x}_0^1 \quad \mathbf{u}_0^1 \quad \mathbf{v}_0^1 \quad \dots \quad \mathbf{x}_p^n \quad \mathbf{u}_p^n \quad \mathbf{v}_p^n \quad \eta_0^1 \quad \dots \quad \eta_p^n \quad \zeta_0^1 \quad \dots \quad \zeta_p^n \quad \mu_0^1 \dots \mu_p^n \quad s_\eta \quad s_\zeta \quad t_F \right]^T \quad (58)$$

The first $[n_s + n_c + n_v]$ $[n(p+1)]$ elements correspond to states, control and virtual controls, respectively. The set of data associated with $\eta_0^1, \dots, \eta_p^n$ represents the upper bounds on virtual controls, while the variables identified as $\zeta_0^1, \dots, \zeta_p^n$ constrain the size of the pointwise trust regions. The variables μ_0^1, \dots, μ_p^n are associated with the cost function. We can see the presence of the slack variables s_η and s_ζ , penalizing virtual controls and trust regions, as they appear in Eq. (43). Finally, since the problem has open final time the variable t_F appears as last element of the augmented vector, by assuming, without compromising any possibility of general application of the proposed method, that $t_0 = 0$. This assumption is always applicable by simply shifting the time vector by the initial time of the aerodynamic descent or the powered landing sequence.

B. Dynamics

Let us consider the dynamics of our system as convexified in Eq. (34). By adopting the time mapping of Eq. (56) we can rewrite it as follows:

$$\dot{\mathbf{x}} = k_t [\mathbf{A}_c \mathbf{x} + \mathbf{A}_k \mathbf{x} + \mathbf{B} \mathbf{u} + \mathbf{C} \mathbf{v} + \mathbf{G}_k] \quad (59)$$

where the equation now describes the evolution of the states with respect to a new independent variable τ , defined between -1 and 1, and chosen because it represents the domain of definition of Legendre-Gauss-Radau polynomials [29, 35]. We can use this expression to derive linear system of equations representing Eq. (59) in discrete form.

By keeping in mind that only \mathbf{A}_k and \mathbf{G}_k change at each iteration we can perform an expansion of the right-hand side of Eq. (59) with respect to the variables \mathbf{x} , \mathbf{u} , ν , and t_F . Let us define the following quantities.

$$\begin{aligned}
\tilde{\mathbf{A}} &\triangleq k_t [\mathbf{A}_k + \mathbf{A}_c] \\
\tilde{\mathbf{B}} &\triangleq k_t \mathbf{B} \\
\tilde{\mathbf{C}} &\triangleq k_t \mathbf{C} \\
\tilde{\mathbf{E}} &\triangleq \frac{1}{2n} [\mathbf{f}_{nc}(\mathbf{x}_k) + \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{C}\nu_k] \\
\tilde{\mathbf{G}} &\triangleq k_t \mathbf{G}_k - \frac{t_{F,k}}{2n} [\mathbf{f}_{nc}(\mathbf{x}_k) + \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{C}\nu_k]
\end{aligned} \tag{60}$$

It is straightforward to verify that the equations of motion can be expressed as

$$\dot{\mathbf{x}} = \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{B}}\mathbf{u} + \tilde{\mathbf{C}}\nu + \tilde{\mathbf{E}}t_F + \tilde{\mathbf{G}} \tag{61}$$

This expression needs to be tailored for the specific domain of choice. In this work we choose to apply the hp-methods as a series of n equally spaced segments, and in each of them we can collocate the differential equations using $p + 1$ nodes. This choice is motivated by the fact that we are interested to real-time-capable methods, and therefore we do not focus on refinement methods which iteratively adapt the size and the distribution of the meshes.

The final step is the inclusion of the pseudospectral differential operator. Note that the derivative of the state in the discrete points \mathbf{x}_i , $i = 1, \dots, n$ can be approximated by a matrix \mathbf{D} in the form

$$\dot{\mathbf{x}} \cong \mathbf{D} \cdot \mathbf{x} \tag{62}$$

Equation (62) tells us that the derivative in one of the discrete points of the domain can be approximated by a linear combination of the values that the variable \mathbf{x} assumes over all the discretized points through the coefficients provided by the columns of \mathbf{D} .

We can exploit this property to finally build the linear matrix representing the equations of motion as follow: defined $\mathbf{D}_{i,\dots}$ as the i^{th} row of \mathbf{D} we have that for each segment $j \in [1, \dots, n]$ and node $i \in [0, \dots, p]$

$$\mathbf{D}_{i,\dots} \cdot \mathbf{x}_{0,\dots,p}^j - \tilde{\mathbf{A}}\mathbf{x}_i^j - \tilde{\mathbf{B}}\mathbf{u}_i^j - \tilde{\mathbf{C}}\nu_i^j - \tilde{\mathbf{E}}t_F = \tilde{\mathbf{G}} \tag{63}$$

which in matrix form can be assembled as

$$\mathbf{A}_{EoM}\mathbf{X} = \mathbf{b}_{EoM} \tag{64}$$

C. Boundary Conditions

We can augment the previous linear system of Eq. (64) by including hard constraints to comply with initial and final states definitions. The corresponding matrix will simply by

$$\mathbf{A}_{bc}\mathbf{X} = \mathbf{b}_{bc} \tag{65}$$

corresponding to

$$\mathbf{A}_{bc} = \begin{bmatrix} I_{n_s} & O_{n_s \times (n_t)(n)(p-2)} & O_{n_s \times n_s} & O_{n_s \times (n_{var} - [(n_t)(n)(p-2) + 2(n_s)])} \\ O_{n_s \times n_s} & O_{n_s \times (n_t)(n)(p-2)} & I_{n_s} & O_{n_s \times (n_{var} - [(n_t)(n)(p-2) + 2(n_s)])} \end{bmatrix} \tag{66}$$

where I_{n_x} and $O_{n_y \times n_z}$ are the identity matrix and the zero matrix of size n_x and $n_y \times n_z$, respectively, while $n_t = n_s + n_c + n_\nu$. The vector \mathbf{b}_{bc} is intuitively defined as

$$\mathbf{b}_{bc} = \begin{bmatrix} \mathbf{x}(t_0) \\ \mathbf{x}(t_F) \end{bmatrix} \tag{67}$$

In case some initial and / or final conditions are left free the corresponding rows in Eqs. (66) and (67) can simply be deleted.

D. Linking conditions

The discretization introduced in Fig. 2 requires some extra constraints known as *Linking conditions*, needed to enforce continuity of the states and controls defined on the edge of the segments. These conditions are represented by equality constraints in the form

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \nu \end{bmatrix}_p^j = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \nu \end{bmatrix}_0^{j+1}, \quad j \in [1, n-1] \quad (68)$$

It is immediate to see that these conditions can be built by assigning the matrices I_{ns} and $-I_{ns}$ to the proper indices of a matrix \mathbf{A}_{lc} with the corresponding vector $\mathbf{b}_{lc} \triangleq O_{n_t \times 1}$.

The overall system of differential equations is therefore given by

$$\mathbf{A}_{eq} \mathbf{X} = \mathbf{b}_{eq} \quad (69)$$

with

$$\mathbf{A}_{eq} = \begin{bmatrix} \mathbf{A}_{EoM} \\ \mathbf{A}_{bc} \\ \mathbf{A}_{lc} \end{bmatrix}, \quad \mathbf{b}_{eq} = \begin{bmatrix} \mathbf{b}_{EoM} \\ \mathbf{b}_{bc} \\ \mathbf{b}_{lc} \end{bmatrix} \quad (70)$$

E. Cost

For the cost function by recovering Eq. (43), and remembering the quadrature expression associated with the hp fLGR method and the continuous expression we can rewrite it as

$$J_{aug} = -w_m m_p^n + w_u \frac{t_F - t_0}{2n} \mathbf{w}_{fLGR}^T \mu + w_\nu \cdot s_\nu + w_\zeta \cdot s_\zeta + w_{t_F} s_t \quad (71)$$

where the weights w_{fLGR} are dictated by the fLGR theory [35, 36], and w_m equal to 0 if we refer to the aerodynamic descent, or larger than 0 if referred to the powered landing case. Finally, note that a penalization on the final time variation is included through a corresponding slack variable s_t . This variable acts as upper bound on the variation of the final time with respect to the previous one, and is also modeled as conic constraint. Its construction is trivial and skipped to avoid excess of redundancy in the equations.

Remark 3: Note that since we are dealing with open final time problems a formal linearization of the second term in Eq. (71), bilinear in the variables μ_i^j and t_F , is needed. For easiness of implementation this linearization is not carried out, and approximated by $\frac{t_F - t_0}{2n} \mathbf{w}_{fLGR}^T \mu$. This approximation is valid as long as the condition $(t_F - t_{F,k}) \mathbf{w}_{fLGR}^T \mu_k \ll (t_F - t_0) \mathbf{w}_{fLGR}^T \mu$ is satisfied. This aspect is omitted here for brevity, but numerically verified during the simulations, and therefore the approximation is valid.

The role of the variables μ is to act as upper bound for the true elements appearing in the cost function of Eq. (15). Given the selected cost function it is immediate to observe that it can be cast in a second-order cone constraint as follows.

$$\left\| \begin{bmatrix} u_\theta \\ u_\psi \end{bmatrix} \right\|_{2,i}^j \leq \mu_i^j, \quad \begin{array}{l} i \in [0, p] \\ j \in [1, n] \end{array} \quad (72)$$

The transcription is completed by applying point-wise the second-order conic constraints representing the upper bound on virtual controls, i.e.,

$$\left\| \nu_i^j \right\|_2 \leq \eta_{\nu,i}^j, \quad \begin{array}{l} i \in [0, p] \\ j \in [1, n] \end{array} \quad (73)$$

and on the trust region.

$$\left\| \tilde{\mathbf{X}} - \tilde{\mathbf{X}}_k \right\|_{2,i}^j \leq \zeta_i^j \quad (74)$$

$$\left\| \zeta \right\|_{2,i}^j \leq s_{\zeta,i}^j, \quad s_{\zeta} \in \mathbb{R}^{np} \quad (75)$$

F. Constraints - Aerodynamic Descent

Finally, for what concerns the constraints detailed in Eqs. (13) and (40), they are expressed as pointwise linear inequalities which complete the transcription.

$$\begin{bmatrix} \mathbf{x}_L \\ \mathbf{u}_L \end{bmatrix} \leq \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}_i^j \leq \begin{bmatrix} \mathbf{x}_U \\ \mathbf{u}_U \end{bmatrix} \quad (76)$$

and

$$\begin{bmatrix} q_{dyn} \\ n_z \end{bmatrix} \Big|_{\mathbf{x}=\mathbf{x}_k} + \begin{bmatrix} \mathbf{G}_{q_{dyn}} \\ \mathbf{G}_{n_z} \end{bmatrix} \Big|_{\mathbf{x}=\mathbf{x}_k} (\mathbf{x} - \mathbf{x}_k) \leq \begin{bmatrix} q_{dyn,U} \\ n_{z,U} \end{bmatrix} \quad (77)$$

G. Constraints - Powered Landing

In addition to Eq. 72 (opportunistically augmented to consider the different number of controls) the constraints of Eqs. (20) and (21) are also modeled as second-order conic constraints. Therefore, they can simply be applied to each discrete node representing position and velocity.

$$\begin{aligned} \left\| \mathbf{A}_{gs} \mathbf{r}_i^j + \mathbf{b}_{gs} \right\|_2 &\leq \mathbf{c}_{gs} \mathbf{r}_i^j + d_{gs} & i \in [0, p] \\ \left\| \mathbf{A}_r \mathbf{r}_i^j + \mathbf{b}_r \right\|_2 &\leq \mathbf{c}_r \mathbf{r}_i^j + d_r, & j \in [1, n] \\ \left\| \mathbf{A}_v \mathbf{v}_i^j + \mathbf{b}_v \right\|_2 &\leq \mathbf{c}_v \mathbf{v}_i^j + d_v \end{aligned} \quad (78)$$

H. Initialization

For the aerodynamic descent the initial guess for the states is built by using linear interpolation between the desired initial and final states, while the controls were kept equal to 0. For the landing phase a more sophisticated strategy is adopted. A scheme illustrating the SPCP algorithm is depicted in Fig. 3. Inspired by the idea of Simplicio et Al. [37] we adopt an educated guess, obtained by solving the problem with the method described in [24]. Given the reference scenario the simplified problem is solved with hard constraints for initial and final conditions, as well as for the time of flight, assumed to be fixed. In this algorithm neither aerodynamic effects, nor control rates are considered, and the gravity is assumed to be constant. Then, the obtained solution is converted into a format compatible with the SPCP transcription illustrated in this section, and utilized as k^{th} solution, with $k = 0$ to start the sequential pseudospectral convex optimization procedure.

The solution obtained contains the thrust vector in Cartesian coordinate, which is converted into the corresponding Euler angle representation by assuming that the x -body axis coincides with the thrust vector. This is a valid assumption since we are dealing with a 3-DOF model. The control rates are then obtained by numerically differentiating the Euler angles and the thrust magnitude profile, completing the information required to build the very first solution according to the format described by Eqs. (44) and (45). It is then possible to start the sequential pseudospectral convex optimization, and at the end of each iteration the algorithm checks whether convergence has been reached. In negative case the current solution is used to build the new subproblem and iterate the procedure. In the opposite case the procedure is concluded. To validate our solution we perform a feedforward propagation of the full nonlinear equations of motion driven by the computed controls, and compare the obtained solution with the optimized one, which is the outcome of the proposed algorithm.

Remark 4: Note that this further step is not considered to be part of the on-board guidance strategy, but it is only meant as validation tool to measure the effectiveness and the accuracy of the proposed algorithm.

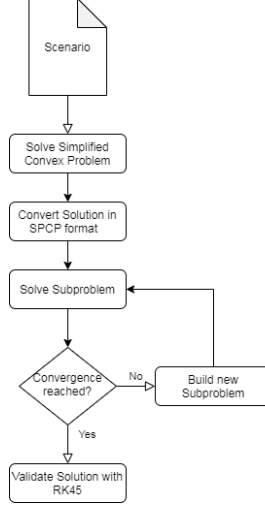


Fig. 3 Sequential Pseudospectral Convex Programming (SPCP) scheme.

VI. Numerical Results

This section illustrates results obtained with the proposed method for both types of scenarios. First, we will describe the aerodynamic descent scenario, followed by the powered landing results.

A. Aerodynamic Descent

For the aerodynamic descent initial and final conditions are described in Table 1. Note that all the positions have been scaled with respect to the initial altitude, the scaled gravity is equal to 1, and all the other variables have been scaled consistently with these two assumptions.

Table 1 Aerodynamic Descent - Initial and Final conditions

Initial State	Value	Final State	Value
$r_A(t_0)$	1.0000	$r_A(t_f)$	0.0714
$r_C(t_0)$	-0.0013	$r_C(t_f)$	0.0006
$r_D(t_0)$	0.0855	$r_D(t_f)$	0.0158
$v_A(t_0)$	-1.1056	$v_A(t_f)$	-0.4464
$v_C(t_0)$	0.00143	$v_C(t_f)$	-0.00021
$v_D(t_0)$	-0.0294	$v_D(t_f)$	-0.0619
$\theta(t_0)$	-0.07356	$\theta(t_f)$	-0.2369
$\psi(t_0)$	-0.00113	$\psi(t_f)$	0.0836

The weights w_{vc} and w_{lr} are equal to 10^4 and 1, while w_u is equal to 100. Finally, the weight penalizing variations of final time w_{T_f} is equal to 10^{-6} . The process is stopped once that the variation between two consecutive values of augmented cost function becomes less than 10^{-5} . Upper bounds for the dynamic pressure and the load factor are here shown in normalized form (i.e., the maximum allowed value is equal to 1).

Results are shown in Figs. 4 through 10. By looking at the states (Fig. 4) we see that the solution shows a smooth behavior while satisfying initial and final conditions. The same holds for the attitude (Fig. 5(a)), where the specific upper and lower bounds for pitch and yaw are met, and the attitude rates, correctly bounded between -2 and 2 deg (Fig. 5(b)). The overall trajectory is depicted in Fig. 6, where the body axes (in RGB convention) depict the corresponding attitude while performing the aerodynamic descent. The associated aerodynamic behavior is visible in Fig. 7(a), showing the forces in body axes. Note that to further enhance the 6-DOF feasibility of the solution the aerodynamic

forces are computed by dynamically trimming the vehicle. To verify the correct behavior of the solution the resulting aerodynamic torques with respect to the center of mass are stored and observed. The torque profiles are depicted in Fig. 7(b), and are close to the zero-machine. The trimming is realized by deflecting the four fins, which ensure that the aero-torque disturbances are nullified while satisfying the maximum allowed fin deflections (shown in normalized coordinates in Fig. 7(c)). We can also observe that the limitations given by the constraints are met, as observed in Figs. 7(d) and 7(e). Note that the upper limit, especially for the dynamic pressure was selected in order to avoid an intrinsic formulation-infeasibility due to the final conditions, where the value of q_{dyn} is implicitly defined. Both constraints are fully satisfied.

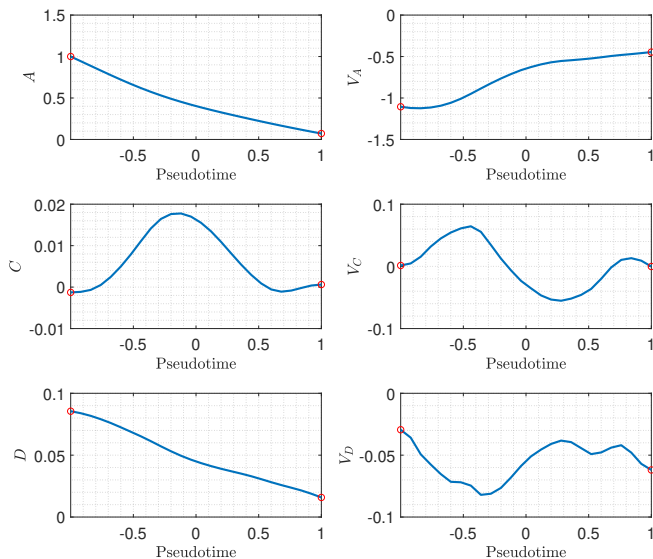


Fig. 4 Aerodynamic Guidance Solution - Translational States.

For what regards the accuracy of the solution a full propagation of the trajectory by using the full set of nonlinear equations is performed, and depicted in Fig. 8, with the mismatch between the two profiles visible in Fig. 9. The two solutions agree very well, with a relative error in the order of $5 \cdot 10^{-5}$ for the position and $4 \cdot 10^{-4}$ for the velocity. In full scale these results correspond to sub-meter error for the position, and less than 0.2 m/s for the velocity. The convergence behavior is depicted in Figs. 10(a) through 10(f). The first thing to observe is the behavior of the cost and the augmented cost, shown in Figs. 10(a) and 10(b). At the beginning of the process larger variations between solutions are experienced. From iteration 4 to the end the algorithm converges to a specific solution, and therefore the upper bounds on trust regions and virtual control fade away. As a consequence the two profiles converge to very similar values. The reduction of virtual controls and trust regions can be seen in Figs. 10(c) and 10(d). We can see that starting from the second iteration the method does not really rely on virtual controls, meaning that the actual controls are sufficient to solve the convex subproblems. The trust regions upper bound becomes smaller than 10^{-6} from the 6th iteration, and besides the small increase in the last iteration, no sensitive variations of the augmented cost function and of the solution are observed. Finally, we can observe that the final time variations between consecutive solutions rapidly decreases too (Fig. 10(e)). Note that as further test the initialized final time is given as the converged final time + 10 s to observe whether the algorithm was able to come back to the optimal value. This behavior is confirmed by looking at the first iteration, where a variation of about 9 s is observed, followed by smaller variations along the successive iterations. The last plot on the bottom right (Fig. 10(f)) shows the decisions of keeping or rejecting the subsolution along the iterations. A subsolution is rejected in two cases: first, when the SOCP solver returns an infeasibility status, or if the maximum number of iterations is reached without finding a valid solution, with the iteration limit for the SOCP solver set equal to 100. This issue did not occur in the results shown here, as confirmed by Fig. 10(f).

To give an intuitive idea of the convergence process we plot the several subsolutions obtained over the iterations in Figs. 11 and 12, showing translational and attitude states, respectively. The process is initialized with a trivial linear interpolation between initial and final desired states (in blue). After three iterations the solution is already resembling

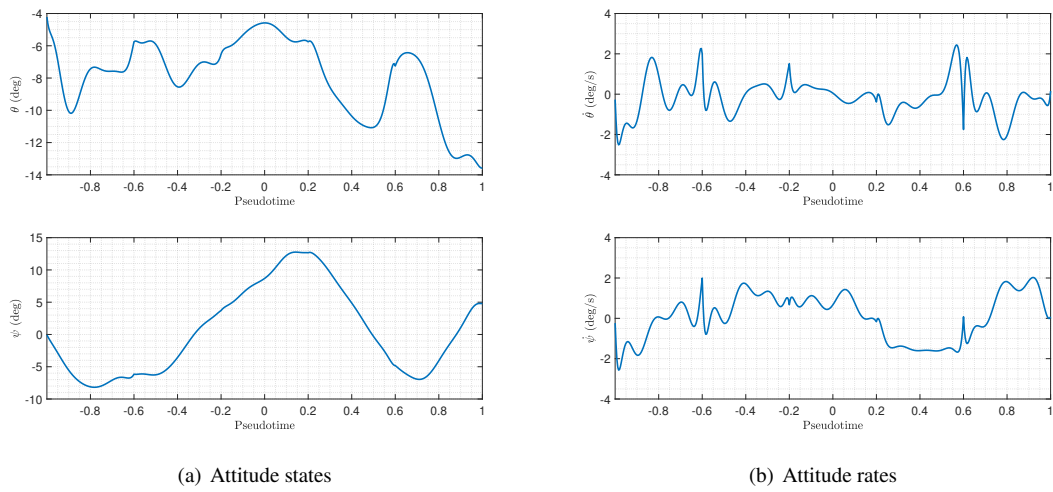


Fig. 5 Aerodynamic Guidance Solution: (a) Attitude states, and (b) Attitude rates.

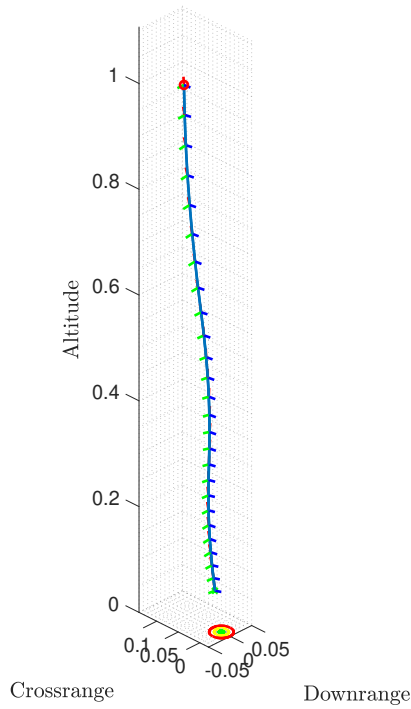
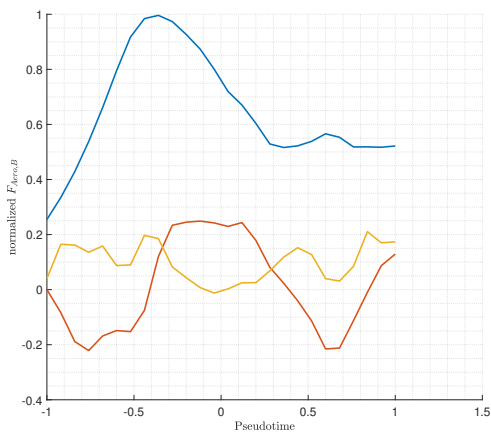
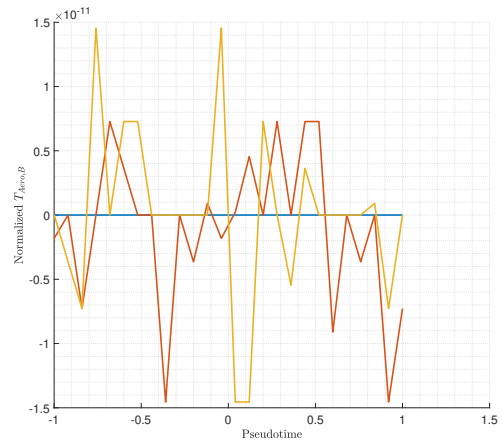


Fig. 6 Aerodynamic Guidance Solution - Trajectory: the body axes are depicted in red (X), green (Y), and blue (Z).

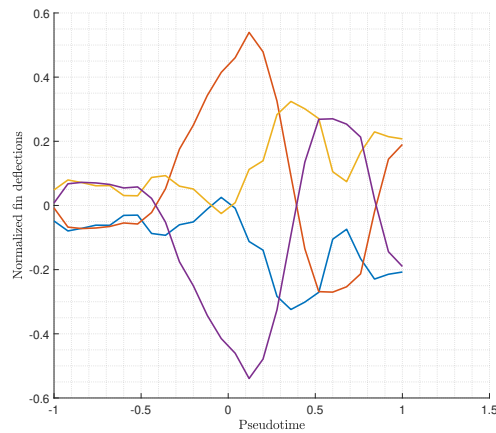
the final one (in red), that is only refined in the remaining iterations. This is a consequence of having variable trust region upper boundaries, which allow larger variations at the beginning if needed, and are dynamically reduced, making, iteration after iteration, the linearized dynamics more and more able to capture the behavior of the nonlinear differential equations underlying the problem.



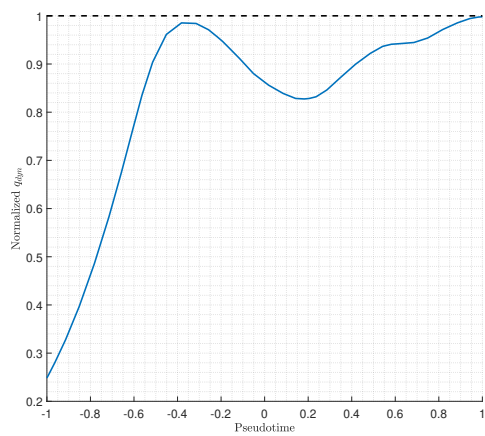
(a) Aerodynamic forces



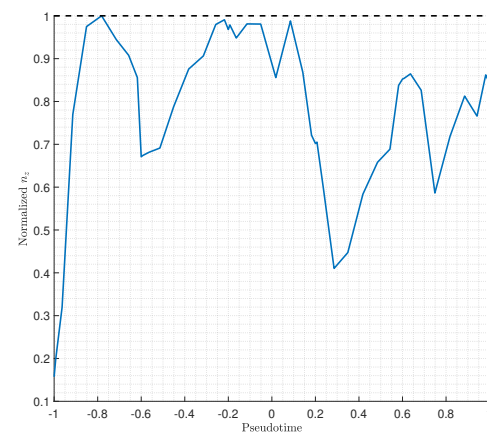
(b) Aerodynamic torques



(c) Fin deflections



(d) Dynamic pressure



(e) Load factor

Fig. 7 Aerodynamic Guidance Solution: (a) Aerodynamic forces, (b) Aerodynamic torques, (c) Fin deflections, (d) Normalized dynamics pressure, and (e) Normalized load factor.

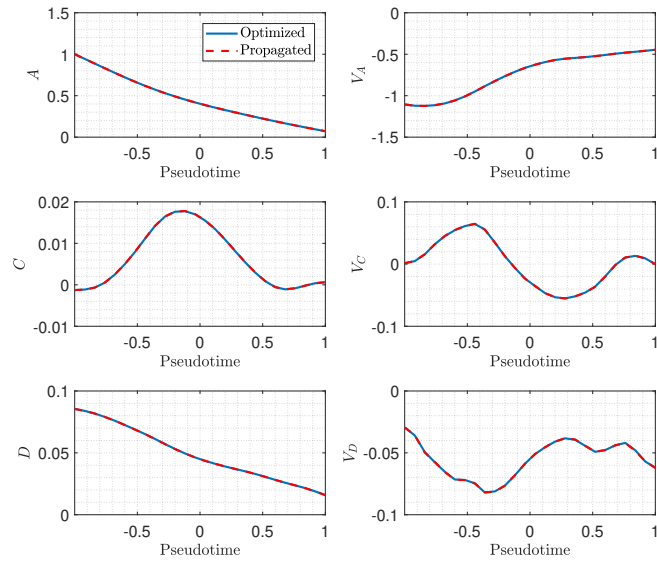


Fig. 8 Aerodynamic Guidance Solution - Comparison with Runge-Kutta Propagation.

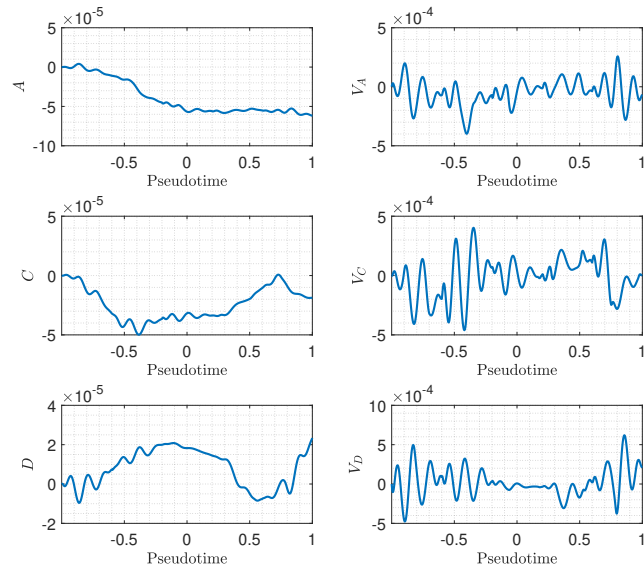
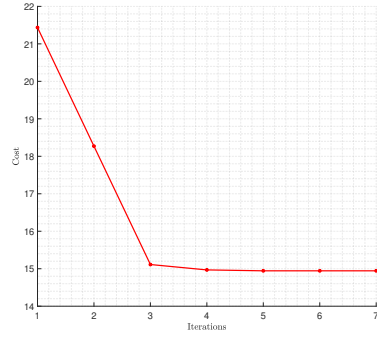
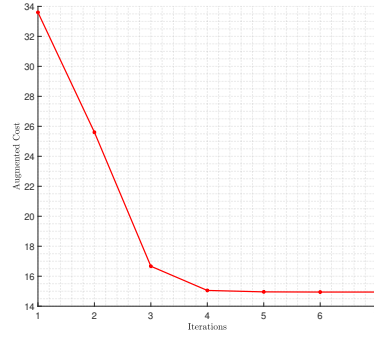


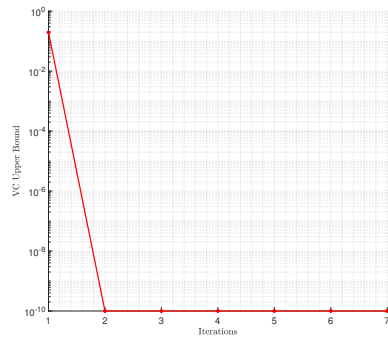
Fig. 9 Aerodynamic Guidance Solution - Error with respect Runge-Kutta Propagation.



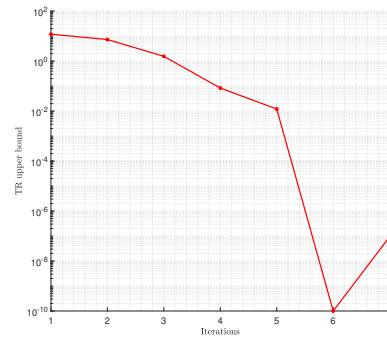
(a) Cost function



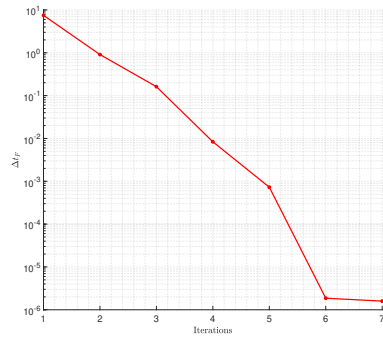
(b) Augmented cost function



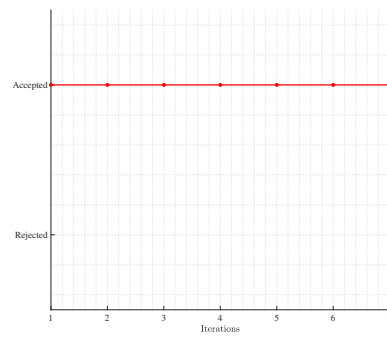
(c) Virtual control upper bound



(d) Trust region upper bound



(e) t_F variations



(f) Acceptance / Rejection sequence

Fig. 10 Convergence behavior: (a) Cost function, (b) Augmented cost function, (c) virtual control upper bound, (d) trust region upper bound, (e) final time variations, and (f) acceptance / rejection decision.

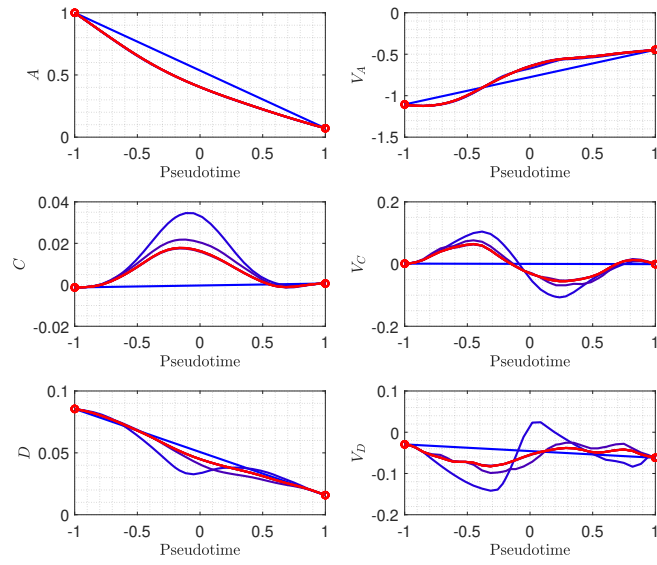


Fig. 11 Convergence behavior: Evolution of translational states.

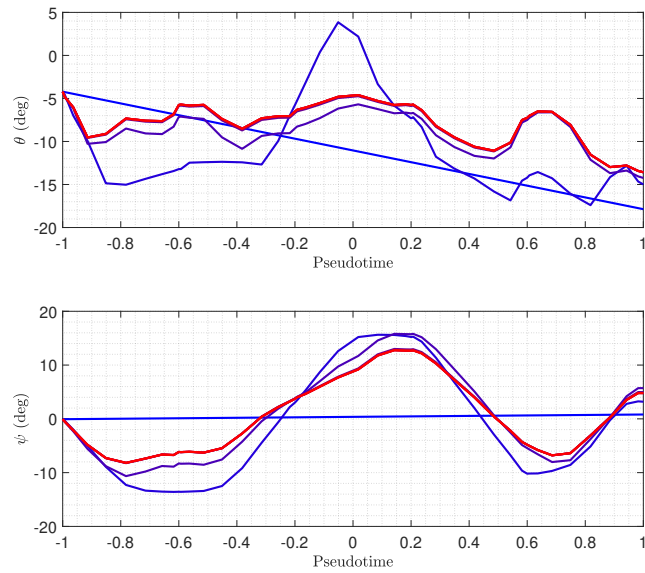


Fig. 12 Convergence behavior: Evolution of attitude states.

B. Powered Landing

The prescribed initial and final conditions for the powered landing scenario we are dealing with are described in Table 2. The initial weights w_{vC} and w_{tF} are equal to 500 and 10. We keep the same penalization of final-time variations

Table 2 Powered Landing - Initial and Final conditions

Initial State	Value	Final State	Value
$r_A(t_0)$	1.0000	$r_A(t_f)$	0.0000
$r_C(t_0)$	0.0086	$r_C(t_f)$	0.0000
$r_D(t_0)$	0.2216	$r_D(t_f)$	0.0000
$v_A(t_0)$	-1.6732	$v_A(t_f)$	-0.0038
$v_C(t_0)$	-0.0010	$v_C(t_f)$	0.0000
$v_D(t_0)$	-0.2365	$v_D(t_f)$	0.0000
$m(t_0)$	1.0000	$m(t_F)$	-
$\phi(t_0)$	3.1381	$\phi(t_f)$	3.1415
$\theta(t_0)$	-0.3118	$\theta(t_f)$	0.0000
$\psi(t_0)$	0.0140	$\psi(t_f)$	0.0000
$T(t_0)$	0.8686	$T(t_F)$	-

as for the aerodynamic descent. Moreover, the same stopping criterion is used. States, and controls are depicted in Figs. 13, 14(a) and 14(b). Besides a smooth solution also in this case with initial and final boundaries fully satisfied we can see that in the last phase of landing the horizontal components of position and velocity are correctly constrained too, and so is the glideslope constraint (here omitted for brevity). Moreover, the attitude rates always lie in the prescribed boundaries, and the vehicle shows a vertical attitude when landing.

The thrust profile, with its corresponding thrust rate and mass profiles are shown in Fig. 15. Note that the thrust rate bound is also satisfied, and so does the thrust, whose maximum allowed scaled value is equal to 2.39. The correct attitude is visible also in Fig. 16, while the corresponding aerodynamic forces and torques are depicted in Figs. 17(a) and 17(b). Note that the forces are computed also in this case by taking the attitude controllability into account, such that part of the torque is reduced by using the fins (Fig. 17(c)) and the remaining disturbance torque can be compensated by the thrust-vector control (as visible in Fig. 17(d)). Finally, as depicted at the end of the scheme of Fig. 3, a validation through Runge-Kutta 45 is performed to verify that the obtained solution satisfies the full nonlinear equations of motion. The results are visible in Figs. 18 and 19. Note that the solution perfectly matches the propagated one, with an error that in full scale is in the order of 2-3 meters for the position components, and below 0.2 m/s for the velocity components. We can have a look at the convergence properties of the algorithm: (Figs. 20(a)-20(f)). First, by looking at the cost function and the augmented cost function they do not significantly change after iteration 6, and given the constant reduction of the consecutive subsolutions, they also converge to the same value, approximately equal to 0.875. The final mass is larger than the one obtained with iteration 1, solved without any aerodynamic effect. This means that the algorithm effectively exploits the lift and drag, rather than fights them to save fuel, and therefore integrates the aerodynamic information to come up with a more efficient solution.

The upper bounds on virtual controls (Fig. 20(c)) become negligible after iteration 5 as well, and therefore have no practical consequences on the physical consistency of the solution. The fact that the convergence process is quite fast is confirmed by the variation of the upper bounds on trust regions (Fig. 20(d)), which at the end of the iterative process is in the order of $2 \cdot 10^{-6}$. The same behavior is visible for the variations of t_F , shown in Fig. 20(e). After iteration 2 the variations on the final time are always smaller than 1 s, and become negligible after iteration 5.

The convergence process in terms of acceptance / rejection is depicted in Fig. 20(f). Only the third iteration is rejected as the maximum number of iteration was reached before an optimal sub-solution was found. In this case the algorithm rejects the new subsolution, and the trust region weight is relaxed to give more search space to the algorithm. This causes the jumps visible in iteration 4 in terms of cost and augmented cost, which show that the solver moves further and then converges more accurately around the new local minimum.

Finally, the convergence behavior can be also seen in Figs. 21- 23, where the colormap moves from blue to red as the number of iterations goes from 1 to 9, which is the last iteration. All the states quickly converge without violating any

constraints, except the thrust rate in iteration 1, visible in Fig. 22. This violation is due to the fact that the corresponding transcription used to initialize the SPCP algorithm does not include the thrust rate constraint yet. The 3-D trajectories can be seen in Fig. 23. Note that once that the algorithm discards iteration 3, corresponding to the quasi-straight trajectory visible on the left side, the successive iterations show trajectories that do not significantly change anymore, but only get refined until the convergence threshold is satisfied.

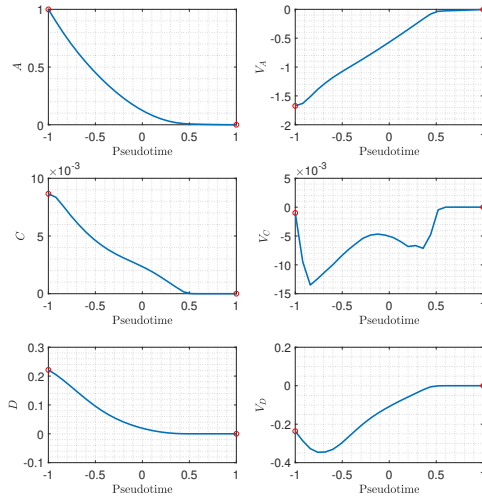
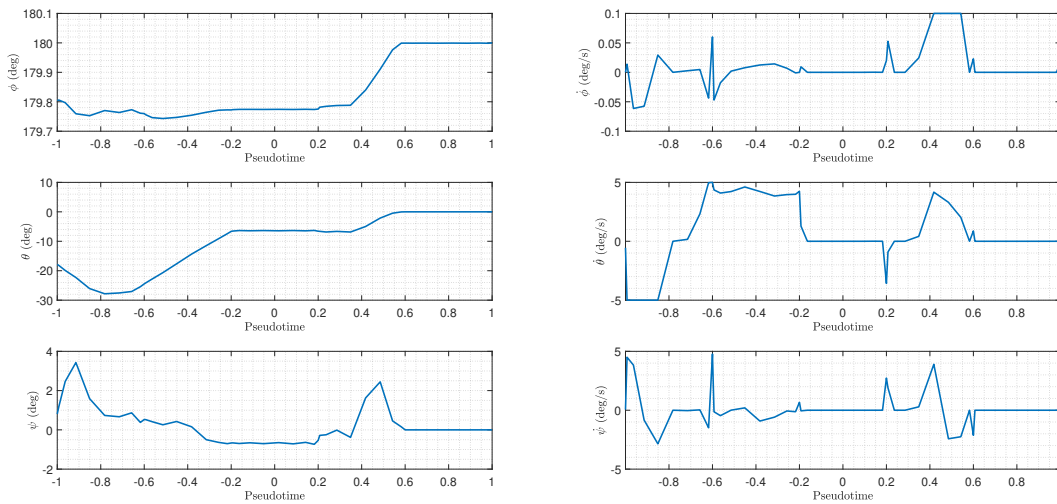


Fig. 13 Landing Guidance Solution - Translational States.



(a) Attitude States

(b) Attitude Rates

Fig. 14 Landing Guidance Solution: (a) Attitude states, and (b) Attitude rates.

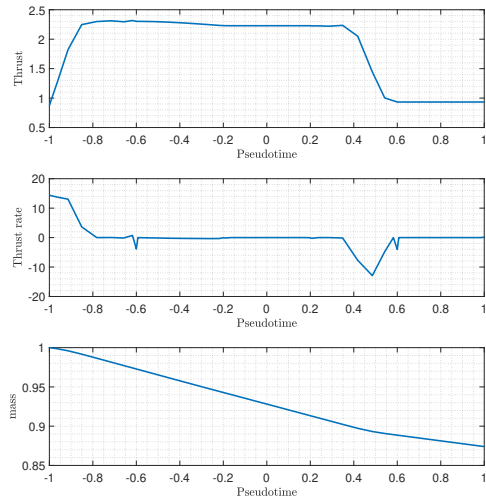


Fig. 15 Landing Guidance Solution: thrust, thrust rate, and mass.

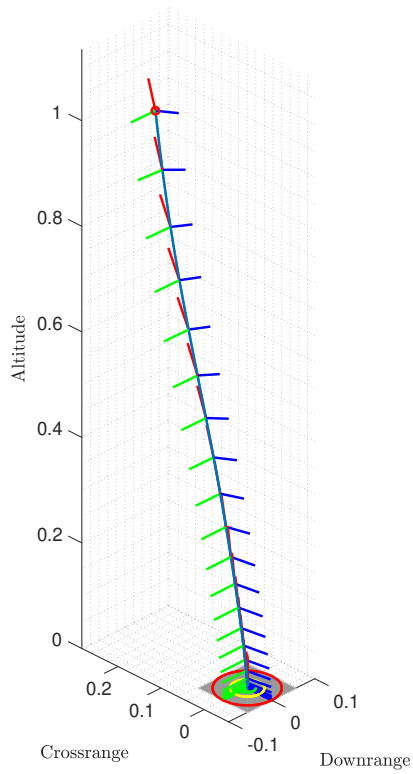
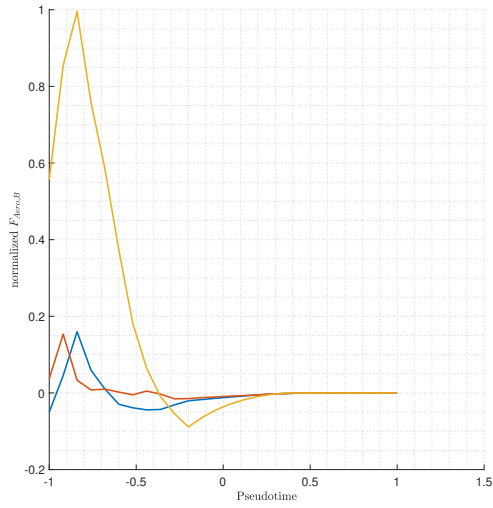
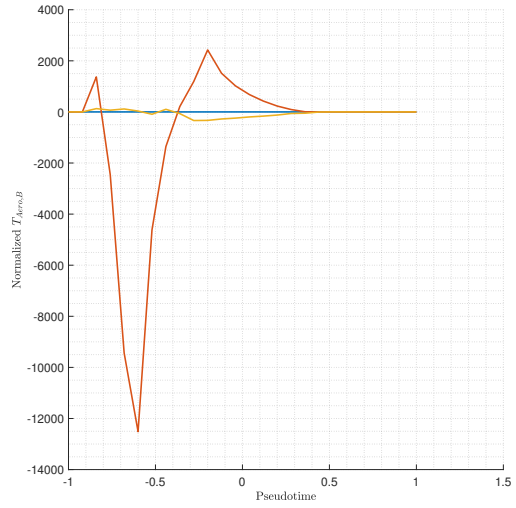


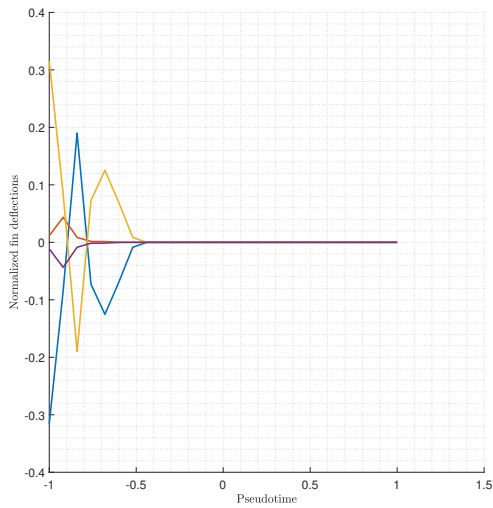
Fig. 16 Landing Guidance Solution - Trajectory x -body is in red, y -body in green, z -body in blue.



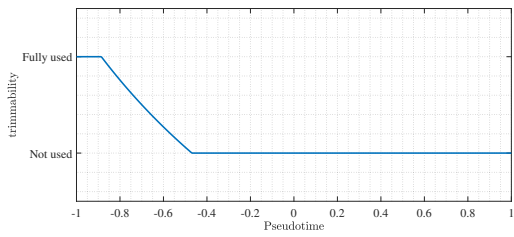
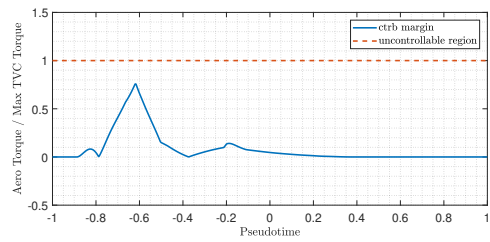
(a) Aerodynamic forces



(b) Aerodynamic torques



(c) Fin deflections



(d) Trimming model

Fig. 17 Landing Guidance Solution: (a) Aerodynamic forces, (b) Aerodynamic torques, (c) Fin deflections, and (d) Trimming model.

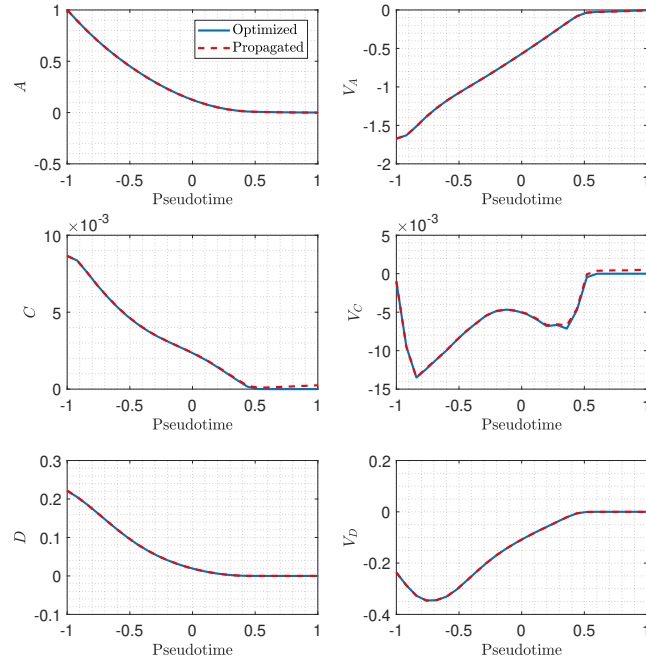


Fig. 18 Landing Guidance Solution - Comparison with Runge-Kutta Propagation.

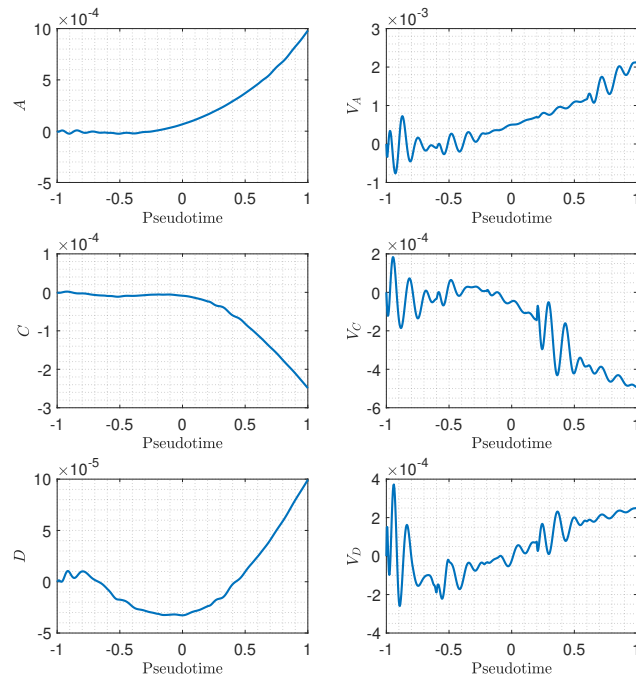
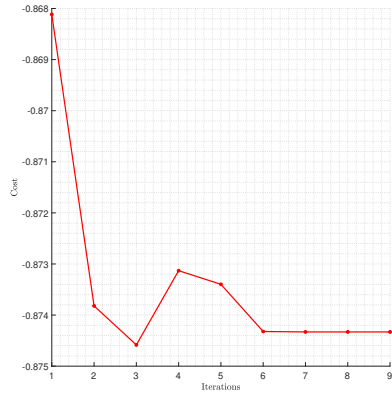
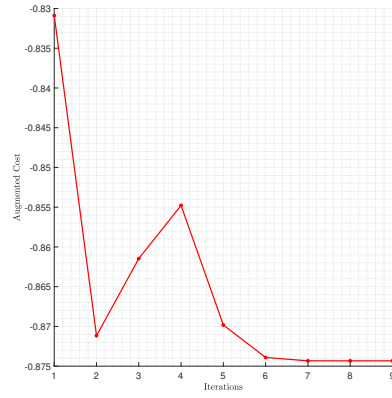


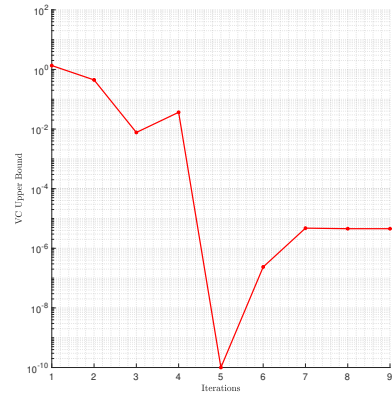
Fig. 19 Landing Guidance Solution - Error with respect Runge-Kutta Propagation.



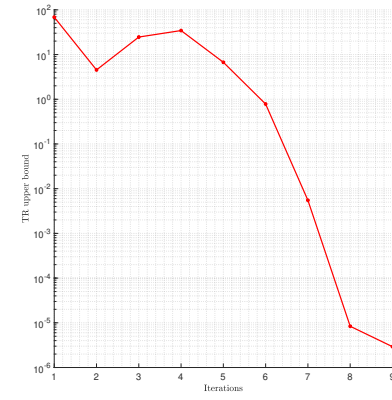
(a) Cost function



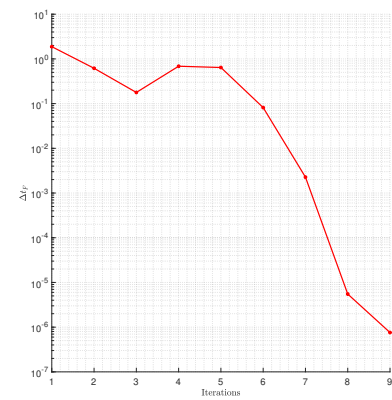
(b) Augmented cost function



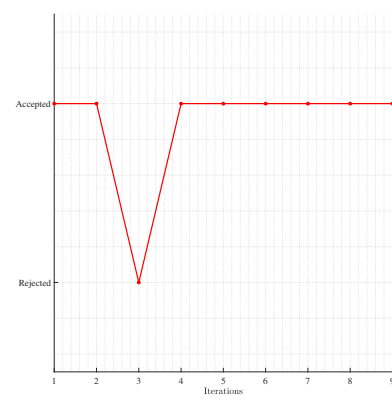
(c) Virtual control upper bound



(d) Trust region upper bound



(e) t_F variation



(f) Rejection / Acceptance

Fig. 20 Convergence behavior: (a) Cost function, (b) Augmented cost function, (c) virtual control upper bound, (d) trust region upper bound, (e) t_F variations, and (f) acceptance / rejection policy.

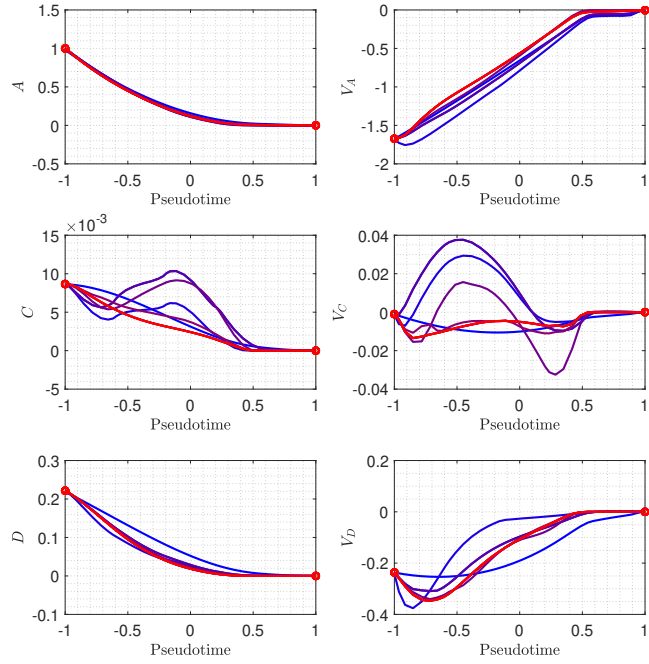


Fig. 21 Convergence behavior - Translational states.

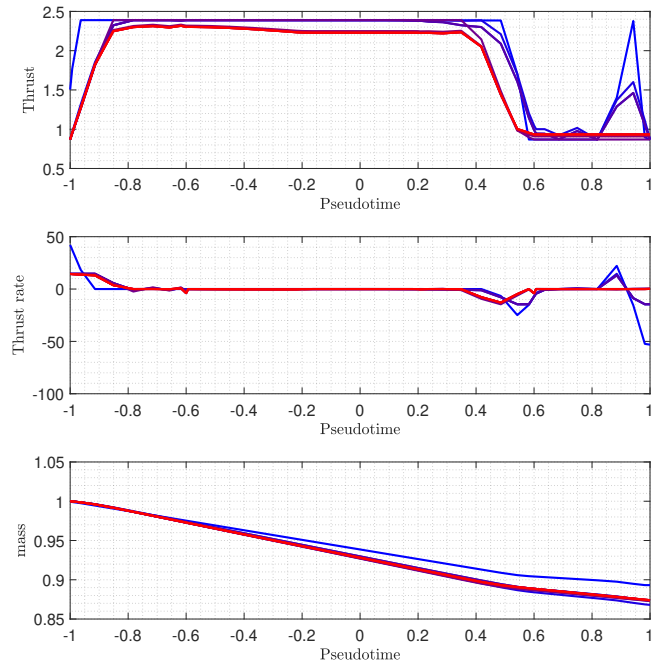


Fig. 22 Convergence behavior - thrust, thrust rate and mass.

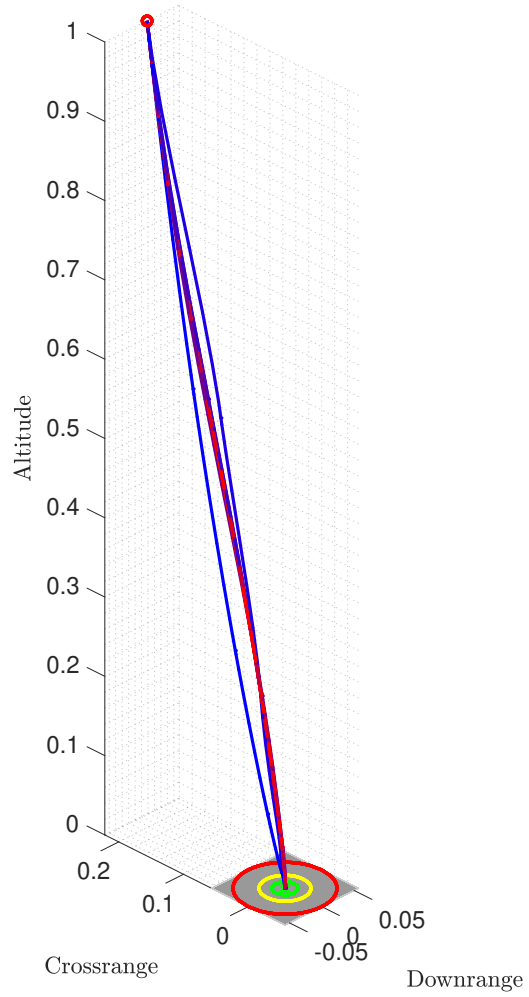


Fig. 23 Convergence behavior - trajectory.

VII. Conclusions

In this paper we proposed an approach able to deal with both the aerodynamic descent and the powered landing phases of a reusable rocket. In the former case the control means are represented by the attitude of the vehicle with respect to the airspeed, which induces the aerodynamic forces that effectively drive the motion of the rocket during the unpowered descent, while in the latter these contributions are effects to be coupled with the thrust force, which is the main control means during the landing phase.

The approach exploited the distinction between convex and non-convex terms to come up with a convex subproblem in which the need of numerical linearization is reduced as much as possible. The reformulation of the problem in terms of rates of Euler angles allowed to express the system in affine form, with the corresponding benefits in terms of convergence behavior. Moreover, a transcription based on hp pseudospectral methods allowed on one side to improve the accuracy of the obtained solution, and on the other side to naturally formulate the free-final time version of the problem.

Numerical results and convergence behavior confirmed that the proposed approach is a viable method to quickly solve both the aerodynamic and landing guidance problems while providing at the same time a very accurate solution, with errors in the meter-range for what regards position, and 0.2 m/s in terms of velocity. The method is therefore a candidate technology to cover the entire descent guidance problem in the frame of the development of guidance and control for reusable rockets.

Future work will include the validation of the methodology over a wide range of initial conditions and a more thorough assessment of the effects of the choice of the weights penalizing virtual controls and trust regions.

Acknowledgments

The first author thanks the Member of the Executive Board of DLR Prof. Hansjörg Dittus for providing special funding, Dr. Sano and Dr. Harigae for granting hosting at JAXA during the development phase of CALLISTO, and Dr. Ishimoto and Dr. Tsukamoto for their support throughout the development of this work.

References

- [1] space.com, “Wow! SpaceX Lands Orbital Rocket Successfully in Historic First,” 2015. URL <http://www.space.com/31420-spacex-rocket-landing-success.html>.
- [2] Musk, E., “Making Mankind a Multiplanetary Species,” 2016. URL https://www.youtube.com/watch?v=H7UyfqI_TE8.
- [3] space.com, “Blue Origin Makes Historic Reusable Rocket Landing in Epic Test Flight,” 2015. URL <http://www.space.com/31202-blue-origin-historic-private-rocket-landing.html>.
- [4] Bezos, J., “Going to Space to Benefit Earth,” 2019. URL <https://www.youtube.com/watch?v=GQ98hGUe6FM&>.
- [5] space.com, “NASA’s Moon-by-2024 Push Could Help Put Astronauts on Mars by 2033, Chief Says,” 201. URL <https://www.space.com/nasa-moon-2024-landing-mars-2033.html>.
- [6] Wu, W., Li, C., Zuo, W., Zhang, H., Liu, J., Wen, W., Su, Y., Ren, X., Yan, J., Yu, D., Dong, G., Wang, C., Sun, Z., Liu, E., Yang, J., and Ouyang, Z., “Lunar farside to be explored by Chang’e-4,” *Nature Geoscience*, Vol. 12, No. 4, 2019, pp. 222–223. URL <https://doi.org/10.1038/s41561-019-0341-7>.
- [7] Dumont, E., Ishimoto, S., Tatioussian, P., Klevanski, J., Reimann, B., Ecker, T., Witte, L., Riehmer, J., Sagliano, M., Giagkozoglou, S., Petkov, I., Rotärmel, W., Schwarz, R. G., Seelbinder, D., Markgraf, M., Sommer, J., Pfau, D., and Martens, H., “CALLISTO: a Demonstrator for Reusable Launcher Key Technologies,” 2019.
- [8] Sagliano, M., Tsukamoto, T., Maces-Hernandez, J. A., Seelbinder, D., Ishimoto, S., and Dumont, E., “Guidance and Control Strategy for the CALLISTO Flight Experiment,” *8th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, 2019.
- [9] Blackmore, L., “Autonomous Precision Landing of Space Rockets,” *National Academy of Engineering: “The Bridge on Frontiers of Engineering”*, Vol. 4, No. 46, 2016, pp. 15–20. URL <https://www.nae.edu/164334/Autonomous-Precision-Landing-of-Space-Rockets>.
- [10] Blackmore, L., Acikmese, B., and Scharf, D. P., “Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171. doi:10.2514/1.47202, URL <http://dx.doi.org/10.2514/1.47202>.

- [11] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004. URL https://www.ebook.de/de/product/3677442/stephen_stanford_university_california_boyd_lieven_university_of_california_los_angeles_vandenberghe_convex_optimization.html.
- [12] Ben Tal, A., and Nemirovski, A., *Modern Lectures on Convex Optimization*, Society for Industrial and Applied Mathematics, 2001. URL <https://www.amazon.com/Lectures-Modern-Convex-Optimization-Applications/dp/0898714915?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbiori05-20&linkCode=sm2&camp=2025&creative=165953&creativeASIN=0898714915>.
- [13] Liu, X., Shen, Z., and Lu, P., “Entry trajectory optimization by second-order cone programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2015, pp. 227–241. doi:10.2514/1.G001210.
- [14] Harpold, J. D., and Graves, C. A., “Shuttle Entry Guidance,” *Journal of Astronautical Sciences*, Vol.27 No.3, May-June 1979, pp.239-268, 1979.
- [15] Sagliano, M., and Mooij, E., *Optimal Drag-Energy Entry Guidance via Pseudospectral Convex Optimization*, 2018. doi: 10.2514/6.2018-1315, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2018-1315>.
- [16] Wang, Z., and Grant, M. J., “Hypersonic Trajectory Optimization by Sequential Semidefinite Programming,” AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-0248, URL <http://dx.doi.org/10.2514/6.2017-0248>.
- [17] Wang, Z., and Grant, M. J., “Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, 2017, pp. 2603–2615. doi:10.2514/1.g002150.
- [18] Wang, Z., and Lu, Y., “Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization,” *Journal of Spacecraft and Rockets*, 2020, pp. 1–14. doi:10.2514/1.a34640.
- [19] Acikmese, B., and Ploen, S. R., “Convex programming approach to powered descent guidance for mars landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. doi:10.2514/1.27553.
- [20] Szmuk, M., Eren, U., and Acikmese, B., “Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance,” AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-1500, URL <http://dx.doi.org/10.2514/6.2017-1500>.
- [21] Szmuk, M., Acikmese, B., and Berning, A. W., “Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints,” AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2016. doi:10.2514/6.2016-0378, URL <http://dx.doi.org/10.2514/6.2016-0378>.
- [22] Szmuk, M., and Acikmese, B., “Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time,” *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-0617.
- [23] Sagliano, M., “Pseudospectral Convex Optimization for Powered Descent and Landing,” *Journal of Guidance, Control and Dynamics*, 2018. doi:10.2514/1.G002818, URL <http://arc.aiaa.org/doi/abs/10.2514/1.G002818>.
- [24] Sagliano, M., “Generalized hp Pseudospectral-Convex Programming for Powered Descent and Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 0, No. 0, 0, pp. 1–9. doi:10.2514/1.G003731, URL <https://doi.org/10.2514/1.G003731>.
- [25] Liu, X., “Fuel-Optimal Rocket Landing with Aerodynamic Controls,” *Journal of Guidance, Control, and Dynamics*, 2018, pp. 1–13. doi:10.2514/1.g003537, URL <https://doi.org/10.2514/1.G003537>.
- [26] Yang, R., and Liu, X., “Fuel-optimal powered descent guidance with free final-time and path constraints,” *Acta Astronautica*, Vol. 172, 2020, pp. 70–81. doi:10.1016/j.actaastro.2020.03.025.
- [27] Wang, J., Li, H., and Chen, H., “An Iterative Convex Programming Method for Rocket Landing Trajectory Optimization,” *The Journal of the Astronautical Sciences*, 2020. doi:10.1007/s40295-020-00235-y.
- [28] Liu, X., Shen, Z., and Lu, P., “Entry Trajectory Optimization by Second-Order Cone Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2016, pp. 227–241. doi:10.2514/1.G001210, URL <https://doi.org/10.2514/1.G001210>.
- [29] Sagliano, M., Theil, S., Bergsma, M., D’Onofrio, V., Whittle, L., and G., V., “On the Radau Pseudospectral Method: theoretical and implementation advances,” *CEAS SPACE Journal*, 2017. doi:10.1007/s12567-017-0165-5.

- [30] Marwege, A., Riehmer, J., Klevanski, J., Guelhan, A., Ecker, T., Reimann, B., and Dumont, E., “First Wind Tunnel Data of CALLISTO - Reusable VTVL Launcher First Stage Demonstrator,” 2019.
- [31] Department of Defense, “World Geodetic System 1984,” Tech. Rep. NIMA TR8350.2 3rd Ed., 1984.
- [32] Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [33] Mao, Y., Szmuk, M., and Acikmese, B., “Successive convexification of non-convex optimal control problems and its convergence properties,” *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016. doi:10.1109/cdc.2016.7798816.
- [34] Bonalli, R., Cauligi, A., Bylard, A., and Pavone, M., “GuSTO: Guaranteed Sequential Trajectory optimization via Sequential Convex Programming,” *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019. doi:10.1109/icra.2019.8794205.
- [35] Sagliano, M., “Development of a Novel Algorithm for High Performance Reentry Guidance,” Ph.D. thesis, 2016. URL <http://elib.suub.uni-bremen.de/edocs/00105082-1.pdf>.
- [36] Garg, D., “Advances in Global Pseudospectral Methods for Optimal Control,” Ph.D. thesis, University of Florida, Gainesville, 2011.
- [37] Simplicio, P., Marcos, A., and Bennani, S., “Guidance of Reusable Launchers: Improving Descent and Landing Performance,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 10, 2019, pp. 2206–2219. doi:10.2514/1.g004155.