Semi-Analytical Adaptive Guidance Computation for Autonomous Planetary Landing

Paolo Lunghi[a,*], Pierluigi Di Lizia[a], Roberto Armellin[b], Michèle Lavagna[a]

[a] *Politecnico di Milano, Aerospace Science & Technology Dept., via La Masa 34, 20156 Milano, Italy*
[b] *The University of Auckland, Engineering Block 1 - Bldg 401, 20 Symonds St., Auckland Central, Auckland 1010, New Zealand*

Abstract

A novel algorithm for autonomous landing guidance computation is presented. The trajectory is expressed in polynomial form of minimum order to satisfy a set of 17 boundary constraints, depending on 2 parameters: time-of-flight and initial thrust magnitude. The consequent control acceleration is expressed in terms of differential algebraic (DA) variables, expanded around the point of the domain along the nominal trajectory followed at the retargeting epoch. The DA representation of the objective and constraints gives additional information about their sensitivity to variations of the optimization variables, which is exploited to find the desired fuel minimum solution (if existing) robustly and with a very light computational effort.

Keywords: Adaptive Guidance, Planetary Landing, Spacecraft Autonomy, Trajectory Optimization, Hazard Detection and Avoidance, Differential Algebra

1. Introduction

Onboard autonomy is a key feature for the next space systems generation. The capability to actively explore in-loco the operational environment, to select the most scientifically relevant locations to explore, and to adapt the system's

*Corresponding author
Email addresses: paolo.lunghi@polimi.it (Paolo Lunghi), pierluigi.dilizia@polimi.it (Pierluigi Di Lizia), roberto.armellin@auckland.ac.nz (Roberto Armellin), michelle.lavagna@polimi.it (Michèle Lavagna)

behavior to detected conditions, increases both the robustness and the flexibility of the vehicle operations, avoiding limitations due to interplanetary distances. Precision landing with Hazard Detection and Avoidance (HDA) capabilities represents a typical case in which high autonomy is required. Among the big challenges to deal with in such a scenario are the short duration of the terminal approach phase, the telecommunications delays, and the necessity to analyze in-loco landing sites and possible trajectories: they all clearly ask for a high level of on-board autonomy in the guidance navigation and control. A crucial issue in this context is the formulation of algorithms capable to be executed in real-time on board the spacecraft, given the limited computing capabilities of flight processors, coupled with strict requirements on update frequencies requested by the whole GNC system.

Complex missions in which landing plays a major role are being developed by all the major agencies: ESA is working together with ROSCOSMOS to establish a cooperative program of lunar exploration, of which the future Luna-Resource Lander (Luna-27) mission is a major element. Part of the European contribution for the Luna-27 mission is the PILOT (Precise and Intelligent Landing using Onboard Technologies) subsystem for enhancing landing capabilities through high landing precision and hazard avoidance. An analogous collaboration has been established in the ExoMars program for Mars exploration. The partially failed landing attempt of the lander *Schiapparelli* in 2016 [1, 2] clearly confirmed the criticality of Entry, Descent, and Landing (EDL) maneuvers for such space missions. A second lander, carrying a rover for surface exploration, is scheduled for launch [3]. It is still evident how the lacking of a full and robust HDA system and the need of maximizing safety affect the mission by imposing strict requirements on the selection of the landing site in terms of landing ellipse minimum dimension, and maximum rock size and abundance [4]. Several other missions involving complex landing and ascent maneuvers are under evaluation by ESA in cooperation with different agencies and in preparation for future arrival of astronauts, including a Lunar Polar Sample Return mission, a Mars Sample Return mission, and the HERACLES mission [5]. Since 2006, technologies

for autonomous landing are studied by NASA in the frame of the Autonomous Landing Hazard Avoidance Technology (ALHAT) program. ALHAT technologies, tested at the end of 2014 in free flight on the Morpheus lander demonstrator, are going to be integrated in future Moon and Mars missions, with a new Martian rover in the Mars 2020 mission [6]. In May 2019, NASA announced the ARTEMIS program, with the aim of sending a human crew back to the Moon by 2024, and to develop new technologies mandatory for a human Mars mission in the 2030s [7]. At the beginning of 2019, CNSA, with the Chang'e-4 mission, has performed its second lunar landing (and the first ever soft landing on the Moon's far side) with a lander/rover system, in the Von Kármán crater floor within the South Pole-Aitken basin [8], while ISRO is carrying out a lunar exploration program, including landing and surface exploration by means of rovers, with a series of missions named Chandrayaan[1].

Spacecraft autonomy in interplanetary missions is a wide and very active research topic: for additional information, and to have a wider overview of the operative scenario, especially for the Moon case, the interested reader can refer to [9, 10, 11, 12].

This paper focuses on the problem of fast and efficient on-board trajectory generation, which is required to achieve the necessary levels of precision and safety during landing. Fuel consumption minimization is assumed as main criterion in the design of a divert trajectory. Aside obvious overall mass containment considerations that affect every space mission, a fuel-optimal approach in hazard avoidance phases maximizes the attainable landing area and increases the possibility of subsequent trajectory corrections during the maneuver, maximizing the robustness and the safety of the landing phase.

The earliest approach to the problem, pushed by the limitation in computational power in the years of the Apollo missions, consisted in the development of a closed-form explicit solution in which the trajectory is expressed in a quartic

[1]ISRO Chandrayaan 2 mission updates, https://www.isro.gov.in/chandrayaan2-latest-updates, Last visit: 31st Dec. 2021

polynomial in time [13]. This kind of solution was later generalized and still considered in more recent years for the landing of the Mars Science Laboratory [14], and a popular variant of this explicit guidance, called *E-Guidance*, was proposed also to achieve HDA capabilities [15]. Despite this easiness of implementation, this category of algorithms is not able to handle any other constraints but boundary conditions, and it does not consider any kind of optimality in the solution [16]. A non-linear formulation was proposed in [17], capable to achieve large retargeting maneuvers, still without considering optimality. Optimal formulations were obtained by means of calculus of variations [18, 19], achieving fuel-optimal or time-optimal trajectories, but under significant assumptions, like planar o nearly planar maneuver, which can limit retargeting capabilities. It has been largely demonstrated that a fuel optimal guidance profile consists in a maximum-minimum-maximum thrust profile [20]: the discontinuous nature of this solution presents a well known challenge for optimization algorithms. Direct optimization proved effective to cope with such difficulties [21, 22]. Convex optimization was proposed as a method to ensure the desired convergence properties. This approach, compared to explicit polynomial schemes, proved effective with a dramatically larger region of initial feasible states, at the price of a relatively large number of optimization variables [23, 24, 25]. The approach was experimentally validated on a ground rocket system [26] and the code for its implementation has recently been optimized for execution on flight hardware [27]. Convexification can be coupled with pseudospectral discretization and a multi-phase approach to extend its flexibility [28]. Also the application of artificial intelligence techniques based on Artificial Neural Networks has recently been proposed: a deep recurrent architecture was found to be capable of imitating a fuel-efficient guidance system for pinpoint landing [29] with good performance in a simulated environment. The effectiveness of the system in a perturbed environment is still to be verified. Reinforcement learning proved also effective in the powered descent guidance problem, both as a standalone method and coupled with traditional guidance methods [30, 31].

A semi-analytical formulation has been developed by our research group

at Politecnico di Milano [32]. In this method, the trajectory is expressed in polynomial form of minimum order to satisfy the set of 12 state boundary constraints – associated with the initial and final position and velocity vectors – and 5 control constraints – due to the thrust vector at the epoch of trajectory generation request and to the desired vertical attitude at touchdown. The acceleration polynomial profile is, therefore, quadratic onto the vertical axis, and cubic onto the two horizontal directions. By imposing boundary conditions, separately for each axis, a fully determined trajectory profile is obtained as a function of only two optimization parameters: the time-of-flight t_f and the thrust magnitude T_0 at the retargeting epoch. Maintaining a restricted set of free parameters enables the adoption of very simple optimization algorithms, such as compass search, ensuring at the same time the fulfillment of the additional constraints and short computation time. Nevertheless, this approach is not free of weaknesses. In particular, the combination of low gradient values together with simple direct optimization methods generally lead to a suboptimal solution. In some cases, this can yield undesired sudden changes in the solution found, with even small variations of the ordered retargeting.

This paper extends and improves the above semi-analytical approach by introducing the use of Differential Algebra (DA) techniques. In DA, the application of usual algebraic operators is broaden from real numbers to functions, modeled as their arbitrary-order Taylor expansions around a selected point [33, 34, 35]. The three components of the acceleration are expressed as DA variables, expanded around the nominal trajectory followed by the lander at the retargeting epoch: in this way the DA formulation leads to an exact, and not only approximate, representation of the acceleration profile, due to its polynomial nature. From the acceleration history, the mass profile is easily obtained through integration, leading to a DA representation of the objective and the constraints as functions of the two optimization variables. Such a representation brings about useful information on the represented quantities. It provides their values at the expansion point, as well as their sensitivity to the optimization variables. This additional information is exploited in the optimization process to enhance the

original method in terms of computational efficiency and quality of the identified solution.

2. Review of Planetary Landing Problem Formulation

The formulation of the optimization problem recalls the one adopted in the previous study [32]. The approach is summarized in the followings. Some modifications are applied, especially in the constraints formulation, to reduce approximations, to get a smoother constrained objective function (which eases the optimization), and to obtain a more rigorous scaling of the problem.

A planetary landing is characterized by fast dynamics. HDA maneuvers are expected to happen in the last few kilometers of altitude, the expected time of flight is in the order of 1 min, and the mass is supposed to significantly change during the maneuver. A constant gravity field with a flat ground are assumed, for the traveled distance is small compared to the planet's radius. Low relative velocity ($<100\,\mathrm{m\,s^{-1}}$) allows to neglect aerodynamic forces, if an atmosphere is present, especially in case of low density such as the case of Mars [23].

The dynamics of the spacecraft translation are expressed in a ground reference system (see Fig. 1) as:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \dfrac{\mathbf{T}}{m} + \mathbf{g} \\ \dot{m} = -\dfrac{T}{I_{\mathrm{sp}} g_0} \end{cases} \tag{1}$$

where the state vector $[\mathbf{r}\ \mathbf{v}\ m]$ includes the position vector $\mathbf{r}$, the velocity $\mathbf{v}$ and the spacecraft mass m. $\mathbf{T}$ is the control thrust vector ($T = \|\mathbf{T}\|$), while $\mathbf{g}$ is the constant acceleration of gravity vector of the planet, I_{sp} the specific impulse of the main engine, and g_0 the standard gravity acceleration on Earth. Please note that in rest of the dissertation, the ground reference system is assumed to be centered at the target landing site, with x as the altitude, y as the *Downrange* direction and z as the *Crossrange* direction (as shown in Fig. 1). In any case

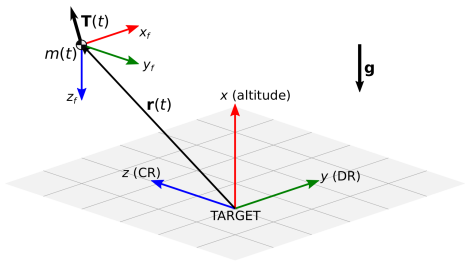


Figure 1: Ground $[x\ y\ z]$ and flight $[x_\mathrm{f}\ y_\mathrm{f}\ z_\mathrm{f}]$ reference systems.

these assumptions are made just to ease the implementation of the proposed algorithms, and do not compromise the generality of the problem formulation.

The mass dynamics are linked to the rest of the system by the thrust magnitude T. The thrust-to-mass ratio $\mathbf{P}$ is defined:

$$\mathbf{P} = \mathbf{T}/m = \dot{\mathbf{v}} - \mathbf{g} \tag{2}$$

The mass mass dynamics can be then rewritten as

$$\dot{m} = -\frac{P}{I_\mathrm{sp} g_0} m \tag{3}$$

where $P = \|\mathbf{P}\|$, which is a first order linear ordinary differential equation whose solution can be computed once the thrust-to-mass ratio profile in time is known.

No assumption is made about the spacecraft navigation system, except that the initial states $[\mathbf{r}_0\ \mathbf{v}_0\ m_0]$ at the initial time t_0 are supposed to be known. For the sake of simplicity, t_0 is set to 0 in the reminder of the paper. The maneuver is assumed to end at the time instant t_f: at this point, the final state vector is constrained to assume the target desired values for what concerns position and velocity, while the final mass remains free: $[\mathbf{r}_\mathrm{f}\ \mathbf{v}_\mathrm{f}\ m(t_\mathrm{f})]$.

2.1. Trajectory Formulation

It is assumed that the main thruster is tightly connected to the spacecraft structure. The thrust vector, when expressed in the body-fixed reference system (BRS) depicted in Fig. 2, has only one non-zero component along the $\mathbf{x}_\mathrm{b}$ direction:

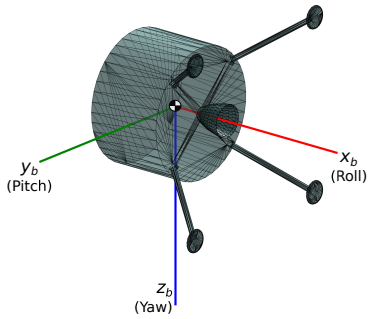


Figure 2: Body-fixed reference system.

$\mathbf{T}_{(\mathrm{BRS})} = [-T\ 0\ 0]^T$. The direction of the thrust vector is then controlled by the attitude of the lander, while its change in magnitude is obtained by the throttling of the main thruster.

Attitude is defined as the rotation of the BRS with respect an auxiliary reference called flight reference system (FRS), denoted by the unit vectors $[\mathbf{x}_\mathrm{f}\ \mathbf{y}_\mathrm{f}\ \mathbf{z}_\mathrm{f}]^T$ in Fig. 1, centered in the center of mass of the spacecraft, the $\mathbf{x}_\mathrm{f}$ axis pointing toward downrange, the $\mathbf{z}_\mathrm{f}$ axis pointing downwards. The rotation is expressed in Euler angles, where θ (pitch angle) is the first rotation around $\mathbf{y}_\mathrm{b}$, ψ (yaw angle) is the second rotation about $\mathbf{z}_\mathrm{b}$, and ϕ (roll angle) is the third rotation around $\mathbf{x}_\mathrm{b}$. This sequence of Euler angles (231) is adopted to avoids singularities in the computation. The system of Eqs. (1) can be re-written, including the attitude parameterization, as:

$$\begin{gathered} \dot{x} = v_\mathrm{x} \qquad \dot{y} = v_\mathrm{y} \qquad \dot{z} = v_\mathrm{z} \\ \dot{v}_\mathrm{x} = -T\frac{\cos\psi\sin\theta}{m} + g_\mathrm{x} \qquad \dot{v}_\mathrm{y} = -T\frac{\cos\psi\cos\theta}{m} \qquad \dot{v}_\mathrm{z} = T\frac{\sin\psi}{m} \\ \dot{m} = -\frac{T}{I_\mathrm{sp} g_0} \end{gathered} \tag{4}$$

It can be noticed that the roll angle ϕ does not affect the system: it is a reasonable result, because a rotation of the spacecraft around the nozzle direction does not modify the thrust pointing direction. Then, the roll angle can be dropped from the guidance profile: its value is actually free and available to

fulfill other design requirements about the landing maneuver (i.e. to optimize the pointing of navigation sensors). In the guidance computation algorithm, its value is assumed to be null constant. Discontinuities are allowed in the thrust magnitude (is assumed that the thrust control has a high bandwidth w.r.t. the lander dynamics), but not in the spacecraft attitude: this imposes an additional boundary constraint on the initial acceleration

$$\dot{\mathbf{v}}(0) = -\frac{T_0}{m_0}\begin{bmatrix}\cos\psi_0 \sin\theta_0 \\ \cos\psi_0 \cos\theta_0 \\ -\sin\psi_0\end{bmatrix} + \mathbf{g} = \mathbf{f}(T_0) \tag{5}$$

whose only unknown is then the initial thrust magnitude T_0.

At the end of the maneuver, the attitude is required to be aligned with the local vertical at the landing site. In case of horizontal terrain, this adds two more boundary constraints:

$$\dot{v}_{\mathrm{y}}(t_{\mathrm{f}}) = \dot{v}_{\mathrm{z}}(t_{\mathrm{f}}) = 0 \tag{6}$$

leading to a total of 17 boundary constraints for position, velocity and acceleration components:

$$\begin{aligned} \mathbf{r}(0) &= \mathbf{r}_0 & \mathbf{r}(t_{\mathrm{f}}) &= \mathbf{r}_{\mathrm{f}} \\ \mathbf{v}(0) &= \mathbf{v}_0 & \mathbf{v}(t_{\mathrm{f}}) &= \mathbf{v}_{\mathrm{f}} \\ \dot{\mathbf{v}}(0) &= \mathbf{f}(T_0) & \dot{\mathbf{v}}(t_{\mathrm{f}}) &= [\mathit{free}, 0, 0]^T \end{aligned} \tag{7}$$

The 3 components of the acceleration are then expressed in a polynomial form of the minimum order needed to satisfy the boundary conditions:

$$\dot{\mathbf{v}}(t) = \begin{bmatrix}\dot{v}_{\mathrm{x}} \\ \dot{v}_{\mathrm{y}} \\ \dot{v}_{\mathrm{z}}\end{bmatrix} = \begin{bmatrix}\dot{v}_{0\mathrm{x}} + c_{1\mathrm{x}}t + c_{2\mathrm{x}}t^2 \\ \dot{v}_{0\mathrm{y}} + c_{1\mathrm{y}}t + c_{2\mathrm{y}}t^2 + c_{3\mathrm{y}}t^3 \\ \dot{v}_{0\mathrm{z}} + c_{1\mathrm{z}}t + c_{2\mathrm{z}}t^2 + c_{3\mathrm{z}}t^3\end{bmatrix} \tag{8}$$

By integrating the acceleration twice and applying the aforementioned boundary constraints, the trajectory is fully defined, unless for the time-of-flight t_{f} and

the initial thrust magnitude T_0, that constitute the optimization variables. The thrust-to-mass ratio can be obtained from the acceleration through the Eq. (2). The corresponding thrust profile is simply:

$$\mathbf{T} = m\mathbf{P} \tag{9}$$

in which the mass profile is computed by numerical integration of Eq. (3). From the thrust vector a complete guidance profile, in the form of Euler angles and thrust magnitude, can be obtained by simple conversions.

2.2. Trajectory Constraints

A feasible landing trajectory is subject to several constraints on both the optimization variables and the states assumed by the system along the whole maneuver (the so-called path constraints). Efficient formulation is required to achieve a precise and fast optimization. Here the constraints relative to the pinpoint landing problem are completely rephrased for an effective and robust evaluation during the guidance computation, having in mind their implementation in an optimization scheme based on Taylor map inversion (as described later in Sec. 3).

The initial thrust magnitude is bounded to the thrust actually available on-board:

$$0 < T_{\min} \leq T_0 \leq T_{\max} \tag{10}$$

while the time-of-flight must lie between its lower and upper limit:

$$0 < t_{\min} \leq t_{\mathrm{f}} \leq t_{\max}, \quad t_{\max} = m_{\mathrm{fuel}} \frac{I_{\mathrm{sp}} g_0}{T_{\min}}, \quad t_{\min} = \left(\frac{2 r_{0\mathrm{x}}}{T_{\max}/m_{\mathrm{dry}} - \|\mathbf{g}\|} \right)^{0.5} \tag{11}$$

The theoretical $t_{\max}$ is determined by the amount of fuel on board m_{fuel}, whereas $t_{\min}$ corresponds to the time required by the lander to reach the ground with maximum thrust pointing downward. Evidently $t_{\min}$ does not corresponds to a feasible soft landing maneuver, and it is adopted as a theoretical lower limit

to exclude singularities arising towards $t_\mathrm{f} = 0$. In the actual implementation, time-of-flight appears always at the denominator. Since floating point division is always slower than multiplication (and even more for DA variables, see Section 3), for code efficiency purposes the inverse time-of-flight $\tau_\mathrm{f} = 1/t_\mathrm{f}$ is adopted instead of t_f as optimization variable, without accuracy losses, with corresponding $\tau_\mathrm{min} = 1/t_\mathrm{max}$ and $\tau_\mathrm{max} = 1/t_\mathrm{min}$.

From here onward, the notation $\mathbf{x} = [\tau_\mathrm{f}, T_0]^T$ is adopted for the optimization variables vector. Box constraints, path constraints and any additional constraint not implicitly satisfied by the polynomial formulation are written in the form of $g(\mathbf{x}) \leq 0$; in order to achieve better convergence properties, they are also scaled to assume a value between -1 and 0 inside the feasible domain. This leads to the normalized inequalities:

$$\frac{T_0 - T_\mathrm{max}}{T_\mathrm{max} - T_\mathrm{min}} \leq 0 \tag{12}$$

$$\frac{T_\mathrm{min} - T_0}{T_\mathrm{max} - T_\mathrm{min}} \leq 0 \tag{13}$$

$$\frac{\tau_\mathrm{f} - \tau_\mathrm{max}}{\tau_\mathrm{max} - \tau_\mathrm{min}} \leq 0 \tag{14}$$

$$\frac{\tau_\mathrm{min} - \tau_\mathrm{f}}{\tau_\mathrm{max} - \tau_\mathrm{min}} \leq 0 \tag{15}$$

Boundaries on thrust are applied also as path constraint:

$$\frac{T(t) - T_\mathrm{max}}{T_\mathrm{max} - T_\mathrm{min}} \leq 0 \tag{16}$$

$$\frac{T_\mathrm{min} - T(t)}{T_\mathrm{max} - T_\mathrm{min}} \leq 0 \tag{17}$$

The angular acceleration of the spacecraft is limited by the actual control torques M_Cmax given by the attitude control system. Coupled terms in attitude dynamics make the extrapolation of the control torques from the thrust vector not straightforward. The objective is to characterize such a rotational rate constraint without coupling the problem to the rotational dynamics, to save computation time. Torques are approximated by the decoupled term due to

the angular acceleration, which is an accurate approximation as the rotational rate remains limited. Exploiting this approximation leads to the following (normalized) inequality:

$$\left(\frac{I_{\max}}{M_{\mathrm{Cmax}}}\right)^2 \|\dot{\boldsymbol{\omega}}(t)\|^2 - 1 \leq 0 \tag{18}$$

in which $\dot{\boldsymbol{\omega}}$ is the derivative of the rotational velocity vector, and $I_{\max}$ is the maximum moment of inertia at initial time t_0, adopted to avoid the on-board calculation of inertia properties, and to include a certain safety margin in the torques evaluation. The assumption of constant roll angle made in Sec. 2.1 implies that the rotational velocity lies in the plane perpendicular to the control acceleration vector (which is aligned to the thrust vector and then to the roll axis). It is then possible to obtain $\boldsymbol{\omega}$ from the relation:

$$\dot{\mathbf{a}} = \dot{a}\hat{\mathbf{a}} + \boldsymbol{\omega} \times \mathbf{a} = \frac{\dot{\mathbf{a}} \cdot \mathbf{a}}{\|\mathbf{a}\|} \frac{\mathbf{a}}{\|\mathbf{a}\|} + \boldsymbol{\omega} \times \mathbf{a} \tag{19}$$

where $\mathbf{a} = \dot{\mathbf{v}} - \mathbf{g}$ is the control acceleration vector (whose derivative $\dot{\mathbf{a}}$ is known exactly, being a polynomial in time), $\dot{a}$ is the derivative of the control acceleration modulus and $\hat{\mathbf{a}}$ is the control acceleration unit vector. The rotational rate vector can then be computed as:

$$\boldsymbol{\omega} = \frac{\left[\left(\frac{\dot{\mathbf{a}} \cdot \mathbf{a}}{\|\mathbf{a}\|^2}\right)\mathbf{a} - \dot{\mathbf{a}}\right] \times \mathbf{a}}{\|\mathbf{a}\|^2} = \frac{\mathbf{a} \times \dot{\mathbf{a}}}{\|\mathbf{a}\|^2} \tag{20}$$

The complete procedure to obtain (20) from (19) is detailed in Appendix A.

In a feasible landing path, the obvious constraint that the lander must not impact the ground can be improved considering a *Glide-Slope Constraint*, which imposes that the lander has to remain in a cone defined by the maximum slope angle $\delta_{\max}$, as showed in Fig. 3. This constraint adds a further margin of safety, even in presence of bulky terrain features near the landing site, limiting at the same time the angle of view on the target. In fact, the performance of vision-based navigation systems depend also on the inclination between the trajectory and the ground [36, 37]. The normalized form of this constraint is:

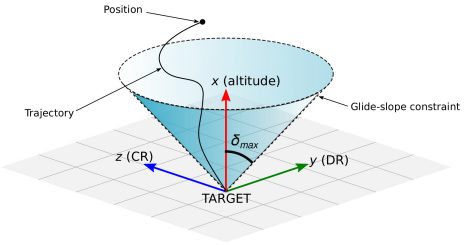


Figure 3: Glide-slope constraint.

$$\frac{r_{\mathrm{y}}^2(t) + r_{\mathrm{z}}^2(t)}{r_{\mathrm{x}}^2(t)\,\tan^2(\delta_{\max})} - 1 \leq 0 \tag{21}$$

Finally, the polynomial formulation does not explicitly consider any constraint on final mass, that must be included between the initial value and the spacecraft dry mass. Since the mass trend is strictly monotone (by problem construction) the evaluation of the maximum mass constraint is redundant: the only constraint with respect the minimum mass is verified. Anyway, the initial mass value is exploited to obtain the normalized relation:

$$\frac{m_{\mathrm{dry}} - m(t_{\mathrm{f}})}{m_0 - m_{\mathrm{dry}}} \leq 0 \tag{22}$$

2.3. Differentiation and integration

In the computation of the objective function and the related constraints, some numerical integration and differentiation are involved. In particular the evaluation of the mass dynamics of Eq. (3) needs an integration, while the derivative of the rotation rate vector $\boldsymbol{\omega}$ is required for the evaluation of the control torque constraint. Plus, all the path constraints need to be satisfied at every time instant during the landing. Pseudospectral techniques allow us to evaluate them discretely at Chebyshev-Gauss-Lobatto (CGL) points. CGL points are selected for they include an explicit node at the end of the domain, simplifying the evaluation of the system mass at the end of the maneuver. Plus, they make possible the adoption of precise and efficient quadrature formulas, as well as differentiation by means of the Chebyshev differentiation matrix [38].

2.4. Optimization Problem

The optimization problem for planetary landing takes the form:

Find T_0 *and* τ_f*, in the domain defined by the inequalities* (12)*,* (13)*,* (14)*, and* (15)*, that minimize the fuel consumption computed by the Eq.* (3)*, subject to constraints* (16)*,* (17)*,* (18)*,* (21) *and* (22)*.*

Denoting as $\mathbf{g}(\mathbf{x})$ the vector of the constraints defined by Eqns. (12–18), (21), and (22), the problem can be written in the compact form:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{g}(\mathbf{x}) \leq 0 \tag{23}$$

where $f(\mathbf{x}) = -m(t_\mathrm{f})$. The optimization could be solved with any nonlinear programming (NLP) solver: the choice of this solver has a huge impact over the final convergence properties and computational time. In this paper, a dedicated optimization algorithm based on DA is developed. In the following section, some notes about differential algebra are reported. Then, the algorithm proposed for landing trajectory optimization is detailed, and the results of some practical simulations are presented and discussed.

3. Optimization Based on Taylor Map Inversion

Differential Algebra techniques were devised to attempt solving analytical problems through an algebraic approach [33]. DA techniques rely on the observation that it is possible to extract more information on a function rather than its mere values: quantities (functions or simple numbers) are represented by their Taylor expansion map around a specified point in the variables space. The straightforward implementation of differential algebra in a computer allows computation of the Taylor coefficients of a function up to a specified order k, along with the function evaluation, with a fixed amount of effort. Standard mathematical operators can be defined for DA variables as well as for real numbers. Similarly to the algorithms for floating point arithmetic, the algorithms for functions followed, including methods to perform composition of functions,

to invert them, to solve nonlinear systems explicitly, and to treat common elementary functions [33, 34]. In addition to these algebraic operations, the DA framework is endowed with differentiation and integration operators, therefore finalizing the definition of the DA structure. For additional details about DA and its usage in navigation and guidance see for example [39, 40, 41, 42].

In the reminder of the paper we will express the DA expansion of a quantity (number or function) $f(x)$ as $[f(x)] = \mathcal{P}_{f(x)}(\delta x)$, in which the square brackets remind that the output is a DA variable, $\mathcal{P}$ indicates a Taylor map or polynomial, the subscript the quantity represented by the Taylor expansion, and the δ reminds that the Taylor expansion is function of the variation with respect to the reference values. The DA computation adopted in this work was practically implemented in the software DACE[2] (Differential Algebra Computational Engine) [43].

This work leverages on the properties of the DA variables to perform a fast and precise optimization. The search for the optimum is divided in two subsequent phases. First, the algorithm searches for a feasible point (*feasibility* phase); once feasibility is achieved, the solution is refined to gain the optimum (*optimality* phase). The feasibility function $\Phi(\mathbf{x})$ is defined as:

$$\Phi(\mathbf{x}) = \sum_i w_i \big(g_i(\mathbf{x})\big)^2 \tag{24}$$

where the weight w_i corresponds to the Heaviside step function of the i-th constraint:

$$w_i = H\big(g_i(\mathbf{x})\big) \tag{25}$$

introduced to consider only active constraints. Given $\Phi(\mathbf{x})$, the feasibility phase goes through the following steps:

1. The constraints are evaluated at the current point $\mathbf{x}_j$, to identify active constraints and weights w_{ij}. The first guess $\mathbf{x}_0$ is here assumed fixed in a predefined point inside the optimization domain. In a real scenario, the

[2]https://github.com/dacelib/dace, Last visit: 31st Dec. 2021

actual spacecraft state at the retargeting epoch can be used as first guess to further reduce the iterations required for the feasibility phase. If the point is feasible, the algorithm passes to the optimality phase.

2. The feasibility function is expanded as a DA variable about the current point $\mathbf{x}_j$:

$$\Phi_j = \Phi(\mathbf{x}_j) \quad \longrightarrow \quad [\Phi_j] = \sum_i \mathcal{P}_{w_i\left(g_i(\mathbf{x}_j)\right)^2}(\delta\mathbf{x}_j) = \mathcal{P}_{\Phi_j}(\delta\mathbf{x}_j) \tag{26}$$

where $\mathcal{P}_{w_i\left(g_i(\mathbf{x}_j)\right)^2}(\delta\mathbf{x}_j)$ denotes the Taylor expansion of the i-th constraint about the reference point $\mathbf{x}_j$ with respect to the optimization variables. To save additional computation time, when a constraint is evaluated in multiple points (e.g. the path constraints at CGL points) only the most violated is included in the computation of $[\Phi_j]$.

3. The Taylor expansion of the gradient of Φ_j with respect to the optimization variables τ_f and T_0 is easily obtained by differentiating the polynomial $[\Phi_j]$ with respect to them. The two resulting polynomials, which form a 2-dimensional array of DA variables, map any variation of $\mathbf{x}$ into the corresponding value of the partial derivatives:

$$[\nabla\Phi_j] = \mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j) \tag{27}$$

4. The polynomial map of the gradient is inverted. To this aim, the constant part of $[\nabla\Phi_j]$ (corresponding to the value of the gradient at the expansion point $\mathbf{x}_j$) is subtracted: the resulting map describes the variation of the gradient as a function of the variation of $\mathbf{x}_j$:

$$[\delta\nabla\Phi_j] = \mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j) - \mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j)\big|_{\delta\mathbf{x}_j=0} = \mathcal{P}_{\delta\nabla\Phi_j}(\delta\mathbf{x}_j) \tag{28}$$

The map in Eq. (28) is then inverted. The inverse map provides the variation of $\mathbf{x}_j$ necessary to achieve any desired variation of the gradient [34]:

$$[\delta \mathbf{x}_j] = \mathcal{P}^{-1}_{\delta\nabla\Phi_j}(\delta\nabla\Phi_j) \tag{29}$$

The inversed map is evaluated at $-\mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j)\big|_{\delta\mathbf{x}_j=0}$ to obtain the correction step to move from $\mathbf{x}_j$ to the point of the domain that cancels the gradient. The result of the evaluation is a vector of two reals:

$$\Delta\mathbf{x}_{\text{step}} = \mathcal{P}^{-1}_{\delta\nabla\Phi_j}\left(-\mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j)\big|_{\delta\mathbf{x}_j=0}\right) \tag{30}$$

5. The expansion point is updated with the relation:

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha\,\Delta\mathbf{x}_{\text{step}} \tag{31}$$

where the correction parameter α depends on the length of the computed step:

$$\alpha = \frac{a}{\|\Delta\mathbf{x}\|} + b \tag{32}$$

where a, $b \in \mathbb{R}^+$. This relation grants the possibility of reducing the step size for unacceptably large $\delta\mathbf{x}_{\text{step}}$, while imposing at the same time a minimum step size. This correction is necessary to achieve a feasible point: in fact, the minimum of the feasibility function Φ lies on the boundary of the feasible domain. Iterating from the infeasible side of the equation without including α would force the optimization process to get arbitrarily close to the feasibility region without reaching it. Once the expansion point is updated, the algorithm starts a new iteration from step 1. If a predefined maximum number of iterations is reached without achieving feasibility, or the requested step length is zero, the process is interrupted and the requested retargeting is classified as infeasible.

In the optimality phase, the solution computed in the feasibility step is refined towards the optimum. The optimality function corresponds to the original objective function modified as:

$$f(\mathbf{x}) = -m(\tau_{\mathrm{f}}) + \frac{1}{t}\sum_i \log\big(-g_i(\mathbf{x})\big) \tag{33}$$

where the second term of the summation is a logarithmic barrier that forces the solution to remain in the feasible domain. The optimality phase works as follows:

1. The optimality function is expanded as a DA variable:

$$f_j = f(\mathbf{x}_j) \quad \longrightarrow \quad [f_j] = \mathcal{P}_f(\delta\mathbf{x}_j) \tag{34}$$

   The solution of Eq. (3) is expanded with an explicit Runge-Kutta 4th order scheme along the CGL points, as described in the previous section. At the first step, the scaling parameter of the logarithmic barrier t is initialized at a certain value $t_0 \in \mathbb{R}^+$. In this case, all the constraints $g_i(\mathbf{x})$ are expanded in the DA function. Again, to save computation time, if the same constraint is evaluated in multiple CGL points, only the most violated is included.

2. The gradient of the optimality map $[\nabla f_j]$ is computed and inverted with the same procedure described in the feasibility step, to obtain the correction vector $\Delta\mathbf{x}_{\mathrm{step}}$. The current point is updated:

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \Delta\mathbf{x}_{\mathrm{step}} \tag{35}$$

   as well as the logarithmic barrier scaling factor:

$$t_{j+1} = kt_j \tag{36}$$

   with $k > 1$.

3. When operating with real numbers, the logarithmic barrier tends to infinity while approaching the constraints, preventing any step outside the feasibility region. The DA representation of the logarithm is a polynomial that generally assumes finite values on the feasibility boundary (unless the expansion point lies exactly on the boundary). Thus, the computed $\Delta\mathbf{x}_{\mathrm{step}}$

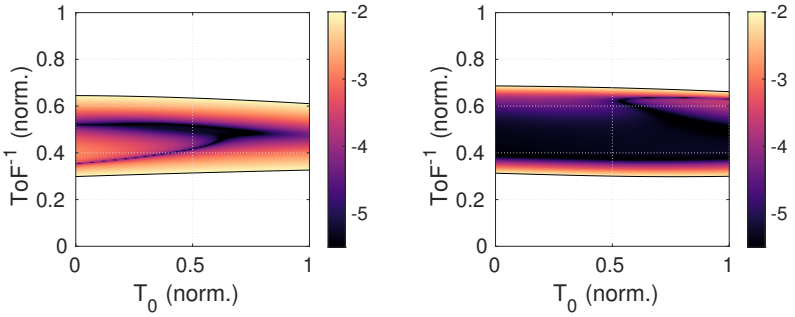
may turn out to point $\mathbf{x}_j$ towards an infeasible region. To avoid this, the constraints are evaluated at the new point: if the new point is feasible, the next iteration restarts from step 1; if not, a golden-section search is performed along the segment $\mathbf{x}_j - \mathbf{x}_{j+1}$ until feasibility is attained.

The optimality step ends when the step size goes below a predefined tolerance (and the solution found is flagged as optimal), or when a maximum number of iterations is reached. In the latter case, the algorithm achieves a suboptimal, though feasible, solution and a warning is issued.

3.1. Expansion Order Selection

To properly select the DA expansion order, trading between precision and computational performance, some preliminary tests on the DA representation of the objective function were carried out. The numerical solution obtained on a grid spanning the entire optimization domain was compared with the solution computed by evaluating the Taylor polynomials. Expansion orders from 2 to 15 were tested, with different expansion points and maneuver cases. Once a predefined relative accuracy threshold of 1% is selected, the effective area is defined as the part of the domain in which the error between the polynomial expansion and the numerical evaluation is within the threshold.

It is observed that increasing the expansion order leads to a more accurate representation of the objective function around the expansion point (and that proves the correct implementation of the DA computation), but without a significant increment of the size of the effective area (see Figure 4, in which a comparison between order 2 and 15 is presented). This is due to the peculiar shape of the objective function, which shows subtle changes of slope and makes its global representation difficult to obtain [32]. In an optimization context, it is more efficient to quickly catch the global features of the objective function, avoiding local minima, and then to refine the solution locally toward the optimum: for all the simulations presented in this paper, the lowest possible order (i.e. order 2) is adopted.



(a) DA order 2 (b) DA order 15

Figure 4: Effective area of the DA representation with respect to the numerical solution obtained with real numbers computation. The color inside indicates the error (in logarithmic scale) of the DA representation with respect to the real number evaluation.

4. Numerical Results

Different numerical simulations of a realistic lunar landing scenario are carried out in this section to estimate the performance of the proposed algorithm. For the sake of comparison, the lander model and the nominal landing trajectory adopted hereafter are taken from a previous work [32]. The nominal lander characteristics are summarized in Table 1.

Table 1: Lander architecture assumptions.

Quantity	**Value**	**Units**
m_{dry}	790	kg
T_{min}	1000	N
I_{sp}	325	s
T_{max}	2320	N
I_{max}	1000	$\mathrm{kg\,m^2}$
M_{Cmax}	40	N m

4.1. Single optimization

In Figure 5 the optimization process for 4 retargeting cases is shown in detail. Each image represents the optimization domain, with constraints boundaries and

the objective function. The optimization steps are depicted in black. The initial conditions are always the same, summarized in Table 2. In each case, the final target position vector $\mathbf{r}_\mathrm{f}$ is changed. Figure 5a is representative of what it could be a standard large retargeting for HDA purposes; case 5b corresponds to a large 2 km crossrange diversion coupled to an extreme brake (−2 km downrange), while the same, but with a downrange extension (+2 km) is shown in Figure 5c. Finally, a large diversion in altitude is displayed in the case 5d.

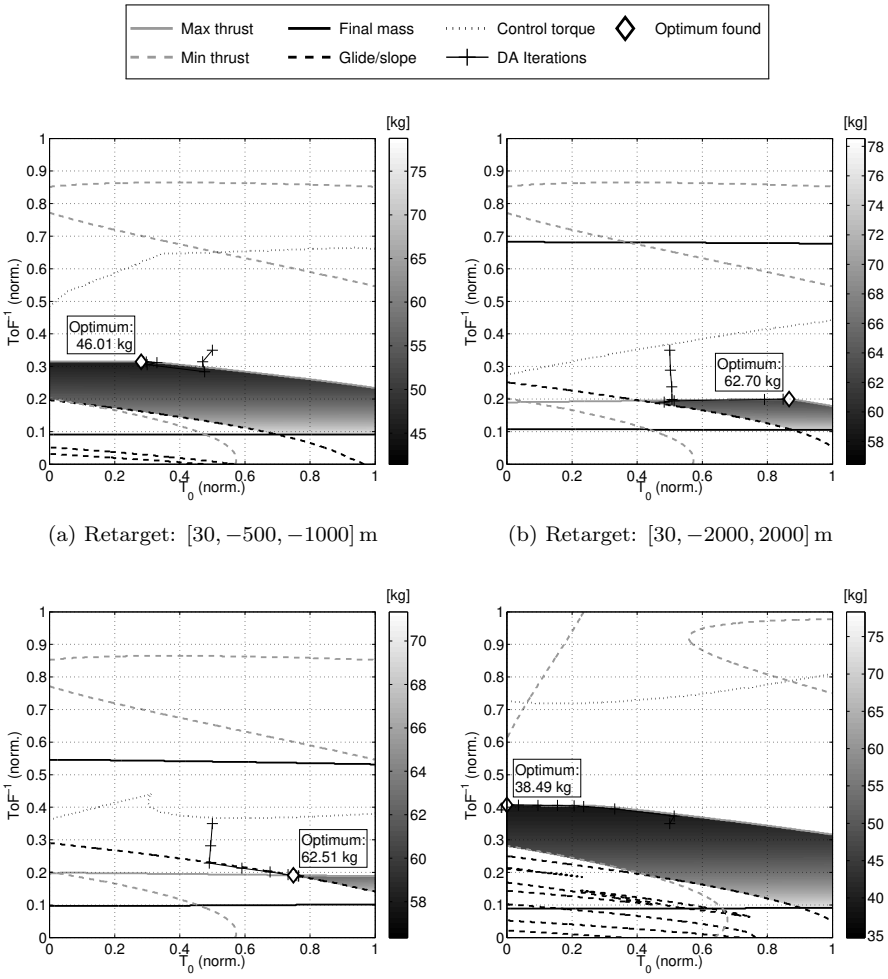
The DA optimization algorithm efficiently detects active constraints and reaches the global optimum solution, even in cases of small feasible area and multiple active constraints, such as the case of Figure 5c. Looking at the different iterations, it can be seen how the system tracks effectively the boundaries of the feasible domain. The optimal solution is, in most of the cases, on the edge of the feasibility, limited by the maximum available thrust. This result is not surprising, taking into account the bang-bang structure of the true optimal control problem the optimal polynomial trajectory tends to.

These results were compared with the solutions computed with a general-purpose nonlinear optimization software (SNOPT): in the worst case observed, the difference between the two solutions was less than 0.2 %.

4.2. Attainable area

A Monte Carlo (MC) simulation is exploited to assess the algorithm performance in terms of attainable landing area and fuel consumption. A series of 1×10^5 uniformly distributed random diversions in the range of ±4000 m along both downrange and crossrange axes is ordered from the initial conditions summarized in Table 2. The attainable landing area can be obtained by correlating the optimization results with the coordinates of the targets landing sites, as shown by Figure 6a, in which only the feasible points (satisfying all the constraints) are shown.

The system is able to compute a feasible landing path in an approximately circular landing area of radius larger than 2300 m centered at the nominal landing site (at the origin of the figure), performing better than required in



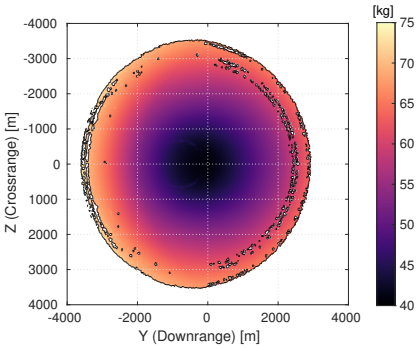
(a) Retarget: $[30, -500, -1000]$ m

(b) Retarget: $[30, -2000, 2000]$ m

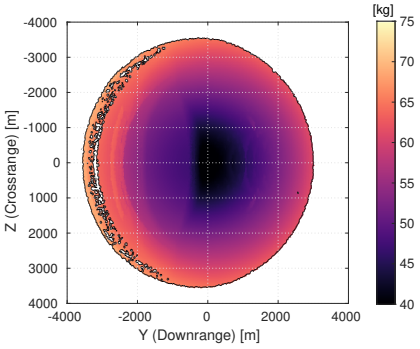
(c) Retarget: $[30, 2000, 2000]$ m

(d) Retarget: $[430, 350, -470]$ m

Figure 5: Single optimization cases. The algorithm effectively tracks the boundaries of the feasible area and efficiently detects the activation of different constraints at each iteration.



(a) DA Optimization



(b) Compass search Optimization

Figure 6: Attainable area and fuel consumption comparison. Colored areas denote feasible solutions, white zones are infeasible.

Table 2: Lunar landing nominal case ($\mathbf{e}$ denotes the vector of Euler angles).

Quantity	**Value**	**Units**
$\mathbf{r}_0$	$[2000, -1062.27, 0]$	m
$\mathbf{r}_\mathrm{f}$	$[30, 0, 0]$	m
$\mathbf{v}_0$	$[-35, 30, 0]$	$\mathrm{m\,s^{-1}}$
$\mathbf{v}_\mathrm{f}$	$[-1.5, 0, 0]$	$\mathrm{m\,s^{-1}}$
$\mathbf{e}_0$	$[-55, 0, 0]$	deg
$\mathbf{e}_\mathrm{f}$	$[-90, 0, 0]$	deg
m_0	865	kg

similar scenarios [44, 45]. In Figure 6b the results of the same simulation obtained with the modified compass search algorithm [32] are reported for comparison. The achievement of the true optimal solution by the new method is further confirmed by the shape of the fuel consumption function: the absence of the sudden variations visible in the compass search results indicates that the algorithm avoids stagnation in local minima.

4.3. Full landing Monte Carlo

To assess the robustness and the flexibility of the proposed approach, a MC analysis (with 1000 runs) of a full lunar landing with a retargeting maneuver is carried out. Taking again the case of Table 2 as reference scenario, deviations on initial position, velocity, attitude, and mass are introduced to assess the robustness of the proposed approach to uncertainties; a large dispersion on the horizontal components of the initial position is assumed to account for uncertainties about the requested retargeting. See Table 3 for a recap of the assumed deviations.

During the maneuver, the inertia drops linearly with the mass from the initial value toward the minimum $[845\ 675\ 675]\,\mathrm{kg\,m^2}$, correspondent to the dry mass $m_\mathrm{dry} = 790\,\mathrm{kg}$. While in the simulation of lander dynamics and attitude control the actual values for mass and MoIs are considered, in the guidance algorithm we

set $I_{\max} = 1000\,\mathrm{kg\,m^2}$ at all times. This value encompasses the whole envelope of assumed initial mass (6σ) and ensures the satisfaction of the attitude control constraint with a certain confidence (also tunable accordingly to mission-specific requirements).

Errors in state estimation and actuation disturbances are included to test the capability of the guidance algorithm to cope with realistic scenarios. To ease the comparison, the same case of [32] is adopted. Here, a brief summary of the introduced disturbances is given: please refer to [32] for a detailed description. During the diversion maneuver, the attitude is assumed to be estimated by a simple integration of a simulated inertial measurement unit (IMU), while the presence of an optical relative navigation system is taken into account by adding realistic errors to the estimation of the position and velocity. A 10 mm thrust misalignment (in a random direction for every MC sample) with respect the spacecraft CoM, and PWPF modulation of control torques by attitude thrusters are included as well to take into account disturbance torques. The guidance system updates the trajectory every 5 s, to cope with navigation and control errors.

Table 3: Lunar landing Monte Carlo initial dispersion.

Condition	**Nominal value**	$\mathbf{1\sigma}$	**UoM**
Initial mass m_0	865	± 10	kg
Initial MoIs $\mathbf{I}_0$	[931 762 762]	$\pm$[11.5 11.5 11.5]	$\mathrm{kg\,m^2}$
Initial position $\mathbf{r}_0$	$[2000\ -1060\ 0]^T$	$\pm[30\ 600\ 600]^T$	m
Initial speed $\mathbf{v}_0$	$[-35\ 30\ 0]^T$	$\pm[0.5\ 0.5\ 0.5]^T$	$\mathrm{m\,s^{-1}}$
Initial pitch angle θ_0	-55	± 5	deg
Initial yaw angle ψ_0	0	± 5	deg

Figure 7 shows the obtained trajectories, while Figures 8 and 9 show the resulting horizontal dispersion in terms of position and velocity, summarized, for the vertical component, in Figures 10 and 11. All the samples lead to a feasible and successful landing; the accuracy at touchdown is compatible with

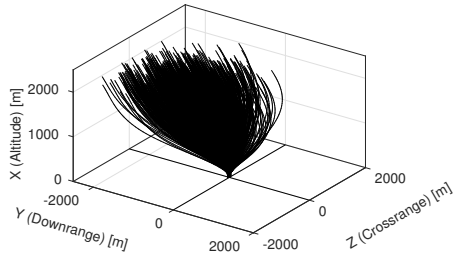


Figure 7: Full landing MC simulation: landing trajectories ensemble.

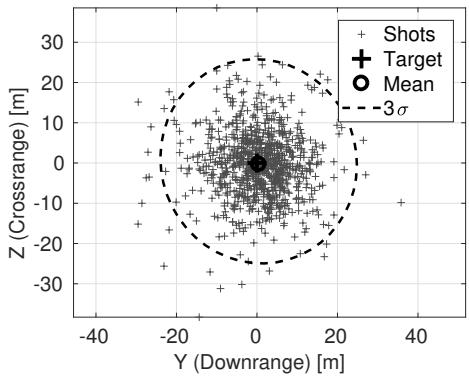


Figure 8: Full landing MC simulation: final horizontal dispersion (position).

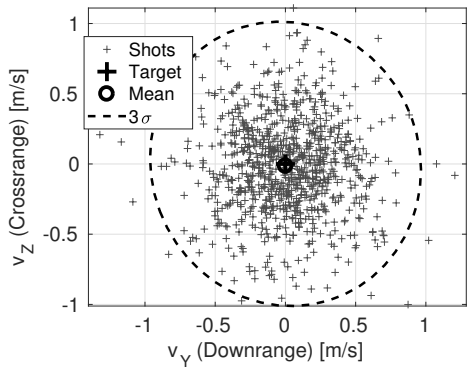


Figure 9: Full landing MC simulation: final horizontal dispersion (velocity).

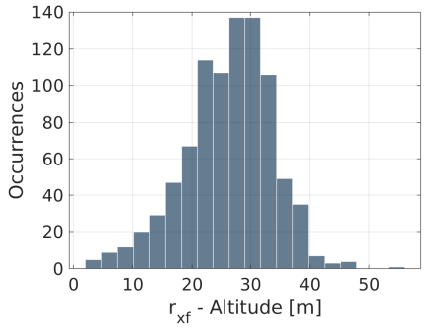


Figure 10: Full landing MC simulation: final altitude distribution.

HDA requirements, with an ordered diversion larger than what is expected in a real mission. Uncertainties at the end of the maneuver have the same order of magnitude of the assumed navigation error, comparable in size to a lander footprint [46]. This result underlines the primary role of the relative navigation system and confirms the outcomes of [32]. The proposed guidance algorithm is capable to drive effectively the system to the desired final states: drift and actuation errors are efficaciously counteracted by regular trajectory updates. Final uncertainty at touchdown is essentially due to navigation errors only, which affect the values of the initial and final states given as input to the guidance. This result conforms to existing literature, which identifies navigation inaccuracies as the main source of landing error [47]. This is not unexpected, as the observed performance demonstrates that the guidance works as desired, and it is basically as precise as the data which is fed with. The final attitude distribution is summarized in Figures 12 and 13: the final spacecraft attitude shows a very narrow dispersion within 1° around the vertical, the same order of magnitude of the attitude control error, enabling an actual and safe hazard avoidance maneuver.

The inclusion of constraint (18) ensures that, for a feasible maneuver, the attitude control system is capable to steer the lander toward the desired target. A further verification is performed on the angular rate during the maneuver, as too high values could be detrimental especially for certain attitude determination

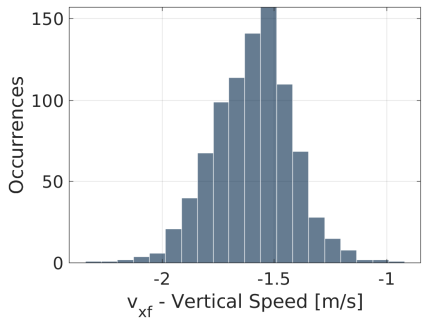


Figure 11: Full landing MC simulation: final vertical velocity distribution.

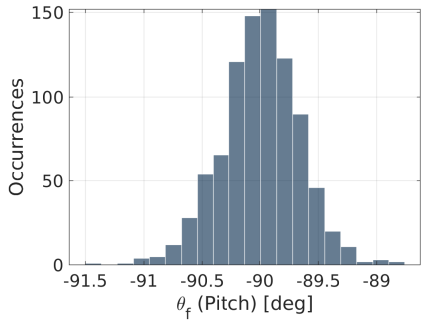


Figure 12: Full landing MC simulation: final pitch angle distribution.

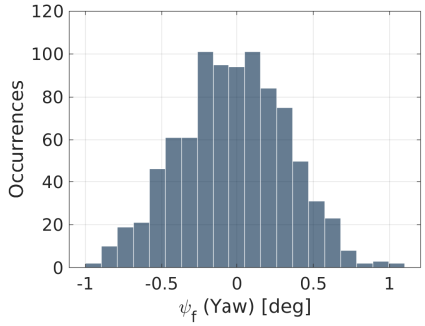


Figure 13: Full landing MC simulation: final yaw angle distribution.

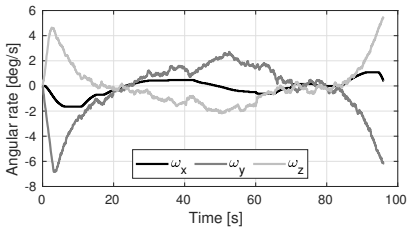


Figure 14: Angular rate trend, example for a $[+2000, +2000]$ m diversion.

and navigation sensors. The maximum rotational speed increases for larger requested diversion. Figure 14 reports the trend for a $[+2000, +2000]$ m retargeting, including all the disturbances assumed in the Monte Carlo simulation, which can be considered representative of a worst case scenario. As expected, the highest values are achieved at the beginning and at the end of the maneuver, when the diversion is commanded and when the lander recovers the vertical attitude. Nevertheless, the actual values are limited: in the whole Monte Carlo simulation, the absolute value of the rotational rate never exceeds $8\,^\circ\,\mathrm{s}^{-1}$, thus enabling a safe landing.

4.4. Computational efficiency

The simulation of Figure 6a was also exploited to obtain an estimation of the computation time. All the simulations were tested on a Intel® Core™ i7-2630QM CPU at 2 GHz of frequency.

Figure 15 illustrates the results obtained (indicated as DA Guidance, DAG, in the chart) compared with the modified compass search (MCS) and the problem formulation adopted in [32]. The histogram at the top reports the time dispersion for feasible cases. For DAG, the mean computation time is 25.23 ms with a standard deviation (STD) of 7.16 ms. This implies a 3σ computation time below 46.71 ms. If an infeasible retarget is ordered (Figure 15, bottom), the system performs a fixed number of iterations until the maximum number of iterations (which is set to 30 for the presented simulations) is reached, or until a stationary point (with null step length) is achieved. In most cases, the algorithm stops due to the maximum iteration limit and the computation time tends to be constant,

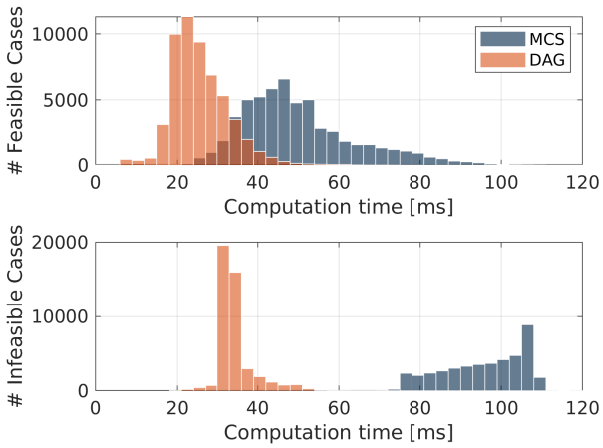


Figure 15: Monte Carlo simulation: computation time for feasible (top) and unfeasible (bottom) cases. Comparison between differential algebra guidance (DAG) and the previous modified compass search (MCS) formulation.

with a mean equal to 33.94 ms. The quite low dispersion (STD = 4.20 ms) is mainly due to the cases in which a stationary point in the feasibility function is reached. In both feasible and infeasible cases, the algorithm is very fast, with a stable computation time.

Looking at the comparison with the previous formulation, DAG resulted significantly faster in both feasible and infeasible cases, with a gain of ∼50 % in the former and ∼65 % in the latter. Moreover, the dispersion in the computation time is noticeably lower for the DAG with respect to the MCS. This latter property is connected with the number of iterations achieved, and it is particularly significant, for it denotes a better numerical stability of the the method, and a more predictable performance once ported on flight hardware.

Even if the actual implementation of DAG on a suitable on-board computer is beyond the scope of this work, preliminary considerations can be made about its possible real-time performance. In fact, flight processors run at significantly lower frequencies (1 order of magnitude) with respect to standard workstations. Moreover, differences in instructions sets and memory access speed make erroneous even simply scaling the runtime by the clock speed as a

method to estimate flight performance: an actual scaling factor of two orders of magnitude between desktop and flight hardware is reported in literature [27]. This factor would entail a rough estimate of 2.5 s for the mean runtime of DAG in flight, compatible with the 5 s update frequency considered in the landing maneuver simulation. Furthermore, it should be considered that the algorithm implementation adopted in this work was not optimized for real-time computing, and DACE is designed as a DA library for general purpose computation, which involves a certain amount of overhead. Consistent opportunities for runtime optimization are expected to be available for in-flight implementation.

Numerically, the operation of map inversion involves the inversion of the linear part of the DA map, followed by a fixed point iteration process to recover the high order coefficients of the Taylor expansion. On real-time hardware, the process of matrix inversion can be time and memory consuming, requiring the temporary storing of intermediate products of the computation. For the case under exam, which deals with only 2 DA variables, the operation requires the inversion of a 2×2 matrix, which is expected to have a negligible impact on the overall computation. Moreover, in previous studies, a real-time tailored implementation based on DACE proved effective in running a high-order numerical extended Kalman filter (involving the inversion of two 6×6 matrices for each time step) at up to 3 Hz on low-performance, flight representative hardware [48].

Taking into account all the possible improvements in code optimization, the proposed solution appears to be promising for on-board, real-time computation.

5. Conclusion

A semi-analytical method for on-board generation of near optimal trajectories for planetary landing has been improved with an ad hoc optimization algorithm based on differential algebra, together with a new formulation of trajectory constraints that makes their computation more efficient and the optimization more robust. The representation of the objective and constraints functions as DA objects opens up the possibility to exploit additional information, such as

their sensitivity to the optimization variables, during the optimization process to quickly estimate the position of the nearest stationary point. Consequently, the optimal solution is found in a low number of iterations. The optimizer has been extensively tested and has showed the capability of finding optimal solutions with an attainable area larger than typically required in real missions. Moreover, the proposed approach turned out to be very fast, as it relies on simple algebraic operations between coefficients of DA variables. The robustness of the proposed method was assessed by a Monte Carlo closed loop simulation including dispersion due to control and navigation errors representative of a real system. The potentially high frequency of the trajectory update makes the algorithm an ideal candidate for a real-time, closed-loop guidance.

Appendix A. Rotational rate vector computation

In this appendix, the procedure adopted to obtain the rotational rate vector, expressed in Eq. (20), from Eq. (19) is reported. The problem of Eq. (19) is to find a vector $\boldsymbol{\omega}$ such that:

$$\mathbf{a} \times \boldsymbol{\omega} = \mathbf{c} \tag{A.1}$$

where $\mathbf{c} = \frac{\dot{\mathbf{a}} \cdot \mathbf{a}}{\|\mathbf{a}\|^2}\mathbf{a} - \dot{\mathbf{a}}$. The general solution to the vector cross product equation is

$$\boldsymbol{\omega} = -\frac{\mathbf{a} \times \mathbf{c}}{\|\mathbf{a}\|^2} + k\mathbf{a} \tag{A.2}$$

However, the assumption of null rotation rate around the roll axis, which is parallel to the control acceleration $\mathbf{a}$, entails $\boldsymbol{\omega} \perp \mathbf{a}$ and $k = 0$. Thus, Eq. (A.2) can become

$$\boldsymbol{\omega} = -\frac{\mathbf{a} \times \mathbf{c}}{\|\mathbf{a}\|^2} = \frac{\left[\left(\frac{\dot{\mathbf{a}} \cdot \mathbf{a}}{\|\mathbf{a}\|^2}\right)\mathbf{a} - \dot{\mathbf{a}}\right] \times \mathbf{a}}{\|\mathbf{a}\|^2} = \frac{\mathbf{a} \times \dot{\mathbf{a}}}{\|\mathbf{a}\|^2} \tag{A.3}$$

References

[1] D. Bonetti, G. D. Zaiacomo, G. Blanco, I. P. Fuentes, S. Portigliotti, O. Bayle, L. Lorenzoni, ExoMars 2016: Schiaparelli coasting, entry and descent post flight mission analysis, Acta Astronautica 149 (2018) 93–105. doi:10.1016/j.actaastro.2018.05.029.

[2] F. Ferri, Ö. Karatekin, S. R. Lewis, F. Forget, A. Aboudan, G. Colombatti, C. Bettanini, S. Debei, B. Van Hove, V. Dehant, A.-M. Harri, M. Leese, T. Mäkinen, E. Millour, I. Muller-Wodarg, G. G. Ori, A. Pacifici, S. Paris, M. Patel, M. Schoenenberger, J. Herath, T. Siili, A. Spiga, T. Tokano, M. Towner, P. Withers, S. Asmar, D. Plettemeier, ExoMars atmospheric mars entry and landing investigations and analysis (AMELIA), Space Science Reviews 215 (1) (2019) 8. doi:10.1007/s11214-019-0578-x.

[3] B. Musetti, A. Allasio, M. Capuano, B. Vinai, F. Spoto, P. Baglioni, T. Blancquaert, A. Haldemann, A. Ivanov, O. Sedykh, ExoMars rover & surface platform mission: Preparing the system integration and verification phase, in: 69th International Astronautical Congress (IAC), Bremen, Germany, 2018.

[4] European Space Agency, ExoMars 2018 landing site selection user's manual, Tech. Rep. EXM-SCI-LSS-ESA/IKI-003, European Space Agency (Dec. 2013).

[5] R. Buchwald, F. Ebert, O. Angerer, A modular ascender concept for sample return missions, in: 69th International Astronautical Congress (IAC), Bremen, Germany, 2018.

[6] E. A. Robertson, Synopsis of Precision Landing and Hazard Avoidance (PL&HA) capabilities for space exploration, AIAA Guidance, Navigation, and Control Conference, Grapevine, TX, 2017, AIAA Paper 2017-1897. doi:10.2514/6.2017-1897.

[7] National Aeronautics and Space Administration, Forward to the Moon: NASA's strategic plan for lunar exploration, NASA press release (Jun. 2019).

[8] L. Qiao, Z. Ling, X. Fu, B. Li, Geological characterization of the Chang'e-4 landing area on the lunar farside, Icarus 333 (2019) 37–51. doi:10.1016/j.icarus.2019.05.029.

[9] D. Folta, N. Bosanac, A. Cox, K. Howell, The lunar icecube mission design: Construction of feasible transfer trajectories with a constrained departure, Vol. 158 of 26th AAS/AIAA Space Flight Mechanics Meeting, 2016, Napa, CA, 2016, p. 1369–1387.

[10] T. F. Dawn, J. Gutkowski, A. Batcha, J. Williams, S. Pedrotty, Trajectory design considerations for exploration mission 1, 2018 Space Flight Mechanics Meeting, Kissimmee, FL, 2018. doi:10.2514/6.2018-0968.

[11] S. Carletta, M. Pontani, P. Teofilatto, Design of low-energy capture trajectories in the elliptic restricted four-body problem, Proceedings of the International Astronautical Congress, IAC, Washington, DC, 2019.

[12] K. Robinson, S. Spearing, D. Hitt, NASA's space launch system: Opportunities for small satellitesto deep space destination, 32nd Annual AIAA/USU Conference on Small Satellites, Logan, UT, 2018.

[13] A. R. Klumpp, Apollo Lunar Descent Guidance, Automatica 10 (2) (1974) 133–146. doi:10.1016/0005-1098(74)90019-3.

[14] E. Wong, J. Masciarelli, G. Singh, Autonomous Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars, in: AIAA Atmospheric Flight Mechanics Conference, Monterey, CA, 2002, AIAA Paper 2002-4619. doi:10.2514/6.2002-4619.

[15] B. Parreira, P. Rogata, E. Di Sotto, A. Caramagno, J. M. Rebordao, P. Motrena, S. Mancuso, Consolidated Performance Assessment of Hazard Avoidance Techniques for Vision Based Landing, in: AIAA Guidance,

Navigation, and Control Conference, Hilton Head, SC, 2007, AIAA Paper 2007-6854. doi:10.2514/6.2007-6854.

[16] I. Gerth, E. Mooij, Guidance for Autonomous Precision Landing on Atmosphereless Bodies, in: AIAA Guidance, Navigation, and Control Conference, National Harbor, MD, 2014, AIAA Paper 2014-0088. doi:10.2514/6.2014-0088.

[17] C. T. Chomel, R. H. Bishop, Analytical lunar descent guidance algorithm, Journal of Guidance, Control, and Dynamics 32 (3) (2009) 915–926. doi:10.2514/1.37700.

[18] A. Lee, Fuel-efficient descent and landing guidance logic for a safe lunar touchdown, AIAA Guidance, Navigation, and Control Conference, aIAA Paper 2011-6499. doi:10.2514/6.2011-6499.

[19] D. Hull, Optimal guidance for quasi-planar lunar descent with throttling, Vol. 140 of 21st AAS/AIAA Space Flight Mechanics Meeting, New Orleans, LA, 2011, p. 983–993.

[20] U. Topcu, J. Casoliva, K. D. Mease, Fuel Efficient Powered Descent Guidance for Mars Landing, in: AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, 2005, AIAA Paper 2005-6286. doi:10.2514/6.2005-6286.

[21] J. T. Betts, Survey of Numerical Methods for Trajectory Optimization, Journal of Guidance, Control, and Dynamics 21 (2) (1998) 193–207. doi:10.2514/2.4231.

[22] F. Fahroo, I. M. Ross, Direct trajectory optimization by a Chebyshev pseudospectral method, Journal of Guidance, Control, and Dynamics 25 (1) (2002) 160–166. doi:10.2514/2.4862.

[23] B. Açikmeşe, S. R. Ploen, Convex Programming Approach to Powered Descent Guidance for Mars Landing, Journal of Guidance, Control, and Dynamics 30 (5) (2007) 1353–1366. doi:10.2514/1.27553.

[24] L. Blackmore, B. Açikmeşe, D. P. Scharf, Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization, Journal of Guidance, Control, and Dynamics 33 (4) (2010) 1161–1171. doi:10.2514/1.47202.

[25] L. Blackmore, B. Açıkmeşe, J. M. Carson, Lossless convexification of control constraints for a class of nonlinear optimal control problems, Systems & Control Letters 61 (8) (2012) 863–870. doi:10.1016/j.sysconle.2012.04.010.

[26] D. P. Scharf, M. V. Regerhr, G. M. Vaughan, J. Benito, H. Ansari, J. Casoliva, S. Mohan, D. Dueri, B. Açıkmeşe, D. Masten, S. Nietfeld, ADAPT demonstrations of onboard large-divert guidance with a VTVL rocket, 2014 IEEE Aerospace Conference, Big Sky, MT, 2014. doi:10.1109/AERO.2014.6836462.

[27] D. Dueri, B. Açıkmeşe, D. P. Scharf, M. W. Harris, Customized real-time interior-point methods for onboard powered-descent guidance, Journal of Guidance, Control, and Dynamics 40 (2) (2017) 197–212. doi:10.2514/1.G001480.

[28] Y. Song, X. Miao, S. Gong, Adaptive powered descent guidance based on multi-phase pseudospectral convex optimization, Acta Astronautica 180 (2021) 386–397. doi:10.1016/j.actaastro.2020.12.019.

[29] R. Furfaro, I. Bloise, M. Orlandelli, P. Di Lizia, F. Topputo, R. Linares, A recurrent deep architecture for quasi-optimal feedback guidance in planetary landing, in: IAA SciTech Forum on Space Flight Mechanics and Space Structures and Materials, Moscow, Russia, 2018, pp. 1–24.

[30] B. Gaudet, R. Linares, R. Furfaro, Adaptive guidance and integrated navigation with reinforcement meta-learning, Acta Astronautica 169 (2020) 180–190. doi:10.1016/j.actaastro.2020.01.007.

[31] R. Furfaro, A. Scorsoglio, R. Linares, M. Massari, Adaptive generalized zem-zev feedback guidance for planetary landing via a deep re-

inforcement learning approach, Acta Astronautica 171 (2020) 156–171. doi:10.1016/j.actaastro.2020.02.051.

[32] P. Lunghi, M. Lavagna, R. Armellin, A Semi-Analytical Guidance Algorithm for Autonomous Landing, Advances in Space Research 55 (11) (2015) 2719–2738. doi:10.1016/j.asr.2015.02.022.

[33] M. Berz, Modern Map Methods in Particle Beam Physics, Academic Press, London, 1999.

[34] M. Berz, Differential Algebraic Techniques. Entry in Handbook of Accelerator Physics and Engineering, World Scientific, New York, 1999.

[35] M. Valli, Nonlinear Estimation and Filtering for Space Applications, PhD in Aerospace Engineering, Politecnico di Milano (2013).

[36] G. Flandin, B. Polle, N. Despré, J. Lheritier, N. Perrimon, P. Blanc-Paques, Maturing Vision Based Navigation Solutions to Space Exploration, in: AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario Canada, 2010, AIAA Paper 2010-7601. doi:10.2514/6.2010-7601.

[37] J. Riedel, A. Vaughan, R. A. Werner, W. Tseng-Chan, S. Nolet, D. Myers, N. Mastrodemos, A. Lee, C. Grasso, T. Ely, D. Bayard, Optical Navigation Plan and Strategy for the Lunar Lander Altair; OpNav for Lunar and other Crewed and Robotic Exploration Applications, in: AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario Canada, 2010, AIAA Paper 2010-7719. doi:10.2514/6.2010-7719.

[38] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, Spectral Methods in Fluid Dynamics, Springer, New York, 1988.

[39] M. Valli, R. Armellin, P. Di Lizia, M. Lavagna, Nonlinear mapping of uncertainties in celestial mechanics, Journal of Guidance, Control, and Dynamics 36 (1) (2013) 48–63. doi:10.2514/1.58068.

[40] M. Valli, R. Armellin, P. Di Lizia, M. R. Lavagna, P. D. Lizia, Nonlinear Filtering Methods for Spacecraft Navigation Based on Differential Algebra, Acta Astronautica 94 (1) (2014) 363–374. doi:10.1016/j.actaastro.2013.03.009.

[41] P. Di Lizia, R. Armellin, F. Bernelli-Zazzera, M. Berz, High order optimal control of space trajectories with uncertain boundary conditions, Acta Astronautica 93 (2014) 217–229. doi:10.1016/j.actaastro.2013.07.007.

[42] P. Di Lizia, R. Armellin, F. Bernelli-Zazzera, M. Berz, High Order Optimal Control of Space Trajectories with Uncertain Boundary Conditions, Acta Astronautica 93 (2014) 217–229. doi:10.1016/j.actaastro.2013.07.007.

[43] M. Rasotto, A. Morselli, A. Wittig, M. Massari, P. Di Lizia, R. Armellin, C. Valles, G. Ortega, Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics, in: 6th International Conference on Astrodynamics Tools and Techniques (ICATT), Darmstadt, Germany, 2016.

[44] J. Delaune, D. De Rosa, S. Hobbs, Guidance and Control system design for Lunar Descent and Landing, in: AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario Canada, 2010, AIAA Paper 2010-8028. doi:10.2514/6.2010-8028.

[45] M. C. Johnson, A Parameterized Approach to the Design of Lunar Lander Attitude Controllers, in: AIAA Guidance, Navigation, and Control Conference, Keystone, CO, 2006, AIAA Paper 2006-6564. doi:10.2514/6.2006-6564.

[46] R. Fisackerly, A. Pradier, B. Gardini, B. Houdou, C. Philippe, D. De Rosa, J. Carpenter, The ESA Lunar Lander Mission, in: AIAA SPACE 2011 Conference and Exposition, Long Beach, CA, 2011, AIAA Paper 2011-7217. doi:10.2514/6.2011-7217.

[47] T. A. Ely, M. Heyne, J. E. Riedel, Altair navigation during translunar cruise, lunar orbit, descent, and landing, Journal of Spacecraft and Rockets 49 (2) (2012) 295–317. doi:10.2514/1.52233.

[48] F. Cavenago, P. Di Lizia, M. Massari, A. Wittig, On-board spacecraft relative pose estimation with high-order extended kalman filter, Acta Astronautica 158 (2019) 55–67. doi:10.1016/j.actaastro.2018.11.020.