Onboard Trajectory Generation for Entry Vehicles via Adaptive Multivariate Pseudospectral Interpolation

Marco Sagliano[1]
Deutsches Zentrum für Luft- und Raumfahrt, Bremen, Germany, 28359

Erwin Mooij[2]
Delft University of Technology, Delft, The Netherlands, 2629

Stephan Theil[3]
Deutsches Zentrum für Luft- und Raumfahrt, Bremen, Germany, 28359

I. Introduction

Since the beginning of the Apollo space flight program, entry guidance has been widely treated by engineers and researchers. The first, successful approach, used for several programs (Apollo, Space Transportation System [1]), was based on the planning of an entry trajectory in terms of the drag-velocity plane. The rationale for this choice resides in the fact that the typical environmental constraints (dynamic pressure, heat flux and load factor), as well as the range-to-go, can be efficiently represented in this drag-velocity plane. The longitudinal guidance can then be derived in several ways. For instance, assuming the equilibrium-glide approximation [2], extracting the longitudinal states (altitude, speed, flight-path angle) from the drag acceleration and its derivatives [3], or implementing constraints-tracking guidance schemes [4]. Similar results can be obtained if the drag-velocity plane is replaced by the drag-energy plane [5–7]. In any case, approximations, disturbances and modeling errors make the use of a feedback controller necessary to track the scheduled nominal

---

[1] GNC Engineer, Navigation and Control Dept., Robert Hooke Str. 7, AIAA Member
[2] Assistant Professor, Astrodynamics and Space Missions Department, Faculty of Aerospace Engineering, Kluyverweg 1, AIAA Associate Fellow
[3] GNC Department Head, Guidance, Navigation and Control Dept., Robert Hooke Str. 7

drag profile. In addition, a bank-reversal logic is usually implemented to keep the heading error within prescribed limits, chosen to steer the vehicle towards the terminal area for energy management (TAEM). In parallel to these approaches, the use of techniques based on optimal control [8–11] has achieved significant improvements. The increased CPU capabilities, together with the development of dedicated algorithms, have led to the possibility to transcribe the problem into a discrete, finite-dimensions problem (i.e. a nonlinear programming problem), which can be efficiently solved with one of the available well-known NLP solvers [12, 13]. The drawback of this approach is that the computed solution is optimal within the limits of the accuracy of the models, and the closeness of the inflight conditions to the nominal ones used to compute it. Even in the presence of tracking controllers, large off-nominal conditions can significantly deteriorate the performance of the system. A possible approach to mitigate the effect of large entry dispersion can be the generation of several reference trajectories corresponding to different initial conditions. During the entry the closest one to the inflight conditions is selected and tracked. However, this approach, here called Neighbor-Trajectory Tracking (NTT) does not guarantee necessarily the best performance, and to verify it, it has been used as benchmark for the performance of the method proposed here.

Significant steps in the use of interpolation-based techniques have already been performed. Saraf et al. [14] used interpolation schemes applied to extremal drag-energy profiles for generating landing footprints for entry missions. Schierman et al. [15] implemented a two-dimensional trajectory database-based scheme to compute online a trajectory by using piecewise-linear functions, and solved the online trajectory-generation problem as a linear programming problem. Lockner et al. [16] developed a more extensive approach based on Tensor Product Splines [17], which perform excellent for the Lunar Landing problem. Arslantas et al. [18, 19] used a similar technique for reachability-set computations. Sagliano et al. [20] merged this approach with pseudospectral methods to provide a real-time capable method able to deal with (limited) off-nominal conditions. However, this approach does no longer provide good performance when the initial dispersion increases. This is due to the fact that the optimal-solutions' behavior may significantly differ over the considered range, and therefore the multivariate scheme does not approximate the behavior of the system well.

In the current work, the multivariate pseudospectral interpolation (MPI) technique is extended

to deal with larger initial conditions by dynamically selecting a subspace of the stored database, leading to the proposed adaptive multivariate pseudospectral interpolation (AMPI) approach. A larger database of optimal trajectories is generated, and a multivariate interpolation approach is applied to a subset of trajectories, properly selected by the onboard software. The result is a highly-adaptive onboard trajectory-generator, which satisfies the tough requirements of memory and power of the onboard computers even in the presence of larger entry-interface dispersions. The rationale for the choice of this method is double: first, it naturally comes out from observing the transcription and the properties associated with the use of pseudospectral methods. Second, the basic multivariate interpolation application in low-dimensions (one in the case treated by Saraf et al. [14], and up to three in the work developed by Lockner et al. [16]) performed very well, and the results encouraged to explore its application in case of larger number of dimensions. Moreover, the concept shown here is quite straightforward, while other approximators, like neural networks, require more complexity, both in terms of network architecture needed to capture the behavior of the system, and for what regards its training. With the AMPI instead, once that the database is computed, no further tuning is required, as the framework is directly provided by the use of pseudospectral methods.

A strong advantage of this method is also associated with the capability to deal with asymmetric entry scenarios (i.e., scenarios where high-crossrange capability is required). In this case the bank-reversal logic is limited, as there is no guarantee that the heading error will enter the heading corridor (a pre-requisite for bank reversals). Moreover, this scenario cannot easily be handled with enhanced drag-energy approaches, which require a smooth drag-energy profile [23]. Moreover, in case important modifications to the scenario are required (e.g., a different landing site or different conditions at the entry interface) a complete, updated guidance scheme can be easily obtained, by generating a new database without any modification of the flight software.

The verification of the method is performed by coupling the method with a feedback controller, and comparing the results with those obtained by the tracking of the closest reference trajectory belonging to the database. Both systems use the same feedback controller. The paper is organized as follows. In Sec. II the reference scenario based on the SHEFEX-3 mission is given, while the framework for the generation of the optimal trajectories is described in Sec. III. In Sec. IV the adaptive

multivariate pseudospectral interpolation method, and the algorithms for its implementation are explained in detail. In Sec. V a comparison between the solutions provided by the AMPI algorithm and the corresponding optimal trajectories is shown, together with a CPU time analysis to assess the real-time capability of the method. In Sec. VI the closed-loop simulation campaign results are discussed, and a comparison w.r.t. the neighbor-trajectory tracking approach is performed. Finally, Sec. VII concludes this paper with some final remarks.

II. Reference Mission Description - SHEFEX-3

The reference scenario is one of the proposed mission profiles for SHEFEX-3 [27], depicted in Fig. 1. SHEFEX (SHarp Edge Flying EXperiment) is a DLR-led series of missions for scientific experiments and development of European technologies for atmospheric reentry. The considered launch site is Andøya Rocket Range, on the western coast of Norway, while the terminal area is located in Greenland. The entry interface is characterized by a steeper flight-path angle and a lower Mach number w.r.t. other entry missions, like those of the Space Shuttle or the X-33 [3, 5, 9]. The mission, from the point of view of the guidance system, terminates at the TAEM interface, associated with a Mach number equal to $\cong 2$. This requires good accuracy in terms of final altitude and final velocity. The nominal entry and terminal conditions are shown in Table 1.

Figure 1: SHEFEX-3 entry vehicle.

Table 1: Nominal entry and terminal conditions for SHEFEX-3 guided flight.

State	Initial Value	Terminal Value
h (km)	100.1	23.5 ± 2.5
θ (deg)	-11.60	-45.75
ϕ (deg)	71.89	66.40
V (m/s)	4712.3	595 ± 25
γ (deg)	-10.31	free
ψ (deg)	-85.92	free

III. Optimal Trajectory Generation

Once the interfaces are defined, it is possible to formulate the related optimal-control problem (OCP). The requirements of the mission include a minimization of the dispersions around the terminal point, for the prescribed ranges of altitude and velocity. The solution to this problem will provide us the reference trajectory and the reference controls, which satisfy all our requirements. Constraints, such as dynamic pressure, heat flux and vertical load factor, are also taken into account, since they are limited by the vehicle's structure.

In this specific case the cost function will be formulated to reduce the final dispersion, therefore, it is computed as the difference between the current and the desired final states, specifically the longitude and the latitude. The final altitude and velocity are included in the cost function as well albeit with smaller weights, to include the condition M = 2 in the database computation. For what regards the mathematical models used, an oblate, rotating Earth is considered, where the gravity acceleration is modeled as central field with only the J_2 term included. The atmosphere is modeled with the U.S. Standard Atmosphere 1976 (US76) model [26]. while the aerodynamics model has been provided by the DLR Institute of Aerodynamics and Flow Technology. For the states full ranges are taken. For the controls, the angle of attack is scheduled as a function of time, and more specifically, it is modeled as two constant values connected by a linear transition at a fixed time. The bank angle and the bank-angle rate limits are explicitly introduced in the transcription process, while bank-angle accelerations have been verified *a-posteriori*. Reference values for the controls are shown in Table 2.

Table 2: Flight control system constraints.

Controls	Values / Ranges
Upper angle of attack α_U (deg)	42
Lower angle of attack α_L (deg)	17.5
Begin of α maneuver $t_{\alpha,U}$ (s)	58
End of α maneuver $t_{\alpha,L}$ (s)	88
Bank angle σ (deg)	[-60, 60]
Angle of attack rate $\dot{\alpha}$ (deg/s)	[-5, 5]
Bank angle rate $\dot{\sigma}$ (deg/s)	[-5, 5]
Angle of attack acceleration $\ddot{\alpha}$ (deg/s^2)	[-4, 4]
Bank angle acceleration $\ddot{\sigma}$ (deg/s^2)	[-4, 4]

A. Cost Function

The objective of the optimal-control problem is to minimize the quadratic cost function J:

$$J = w_\theta \left[\theta(t_F) - \theta_{ref}\right]^2 + w_\phi \left[\phi(t_F) - \phi_{ref}\right]^2 + w_h \left[h(t_F) - h_{ref}\right]^2 + w_V \left[V(t_F) - V_{ref}\right]^2 \qquad (1)$$

The terminal conditions in terms of altitude and velocity have been included as soft constraints (through Eq. (1)) in the transcription to relax the trajectory-database computation. The weights w_i are equal to 1 for longitude θ and latitude ϕ, and equal to 10^{-4} for altitude h and velocity V.

B. Dynamics

During entry, the vehicle's motion is described by the following set of differential equations [23]:

$$\begin{aligned}
\dot{h} &= V \sin\gamma \\
\dot{\theta} &= \frac{V \cos\gamma \sin\psi}{r \cos\phi} \\
\dot{\phi} &= \frac{V \cos\gamma \cos\psi}{r} \\
\dot{V} &= -D - g \sin\gamma + \omega^2 r \cos\phi \left(\sin\gamma \cos\phi - \cos\gamma \sin\phi \cos\psi\right) \\
\dot{\gamma} &= \frac{L \cos\sigma}{V} + \left(\frac{V}{r} - \frac{g}{V}\right) \cos\gamma + \\
&\quad + 2\omega \cos\phi \sin\psi + \frac{\omega^2 r}{V} \cos\phi \left(\cos\gamma \cos\phi + \sin\gamma \sin\phi \cos\psi\right) \\
\dot{\psi} &= \frac{L \sin\sigma}{V \cos\gamma} + \frac{V}{r} \cos\gamma \sin\psi \tan\phi - 2\omega \left(\tan\gamma \cos\phi \cos\psi - \sin\phi\right) + \\
&\quad + \frac{\omega^2 r}{V \cos\gamma} \sin\phi \cos\phi \sin\psi \\
\dot{\sigma} &= u_\sigma
\end{aligned} \qquad (2)$$

r is the radial position, γ and ψ are the flight-path angle and the heading angle, the latter being equal to zero when the vehicle flies towards the local north. D and L are the drag and lift accelerations, while g is the gravity acceleration, and m is the mass of the vehicle, equal to 500 kg. Finally, ω is the Earth's rotation rate, equal to $7.2921 \cdot 10^{-5}$ rad/s. From Eq. (2) one can see that the state has been augmented by adding the bank angle. This permits to limit the bank-angle rate, which is the effective control input to the system.

C. Constraints

Three constraints are included in the transcription, that is, the dynamic pressure, $\overline{q}$, the stagnation heat flux, $\dot{Q}$ (computed by using the cold-wall model for a laminar boundary layer), and the vertical load factor, n_z. These three constraints can be computed according to

$$\overline{q} = \tfrac{1}{2}\rho V^2, \;\; \dot{Q} = k_q\sqrt{\rho}V^3, \;\; n_z = \tfrac{|L\cos\alpha + D\sin\alpha|}{g_0} \tag{3}$$

where ρ is the atmospheric density, expressed in $\mathrm{kg/m^3}$, k_q is a constant depending on the material and the geometry of the thermal protection system, for SHEFEX-3 equal to $1.2444 \cdot 10^{-3}$ $\mathrm{kg^{1/2}/m^3}$, and g_0 is the gravity acceleration at sea level, ($g_0 = 9.782$ $\mathrm{m/s^2}$). The structural limits of the vehicle and the active thermal-protection system dictate a limit for the above constraints. These limits are equal to $\overline{q}_U = 5 \cdot 10^4$ $\mathrm{N/m^2}$, $\dot{Q}_U = 6.5$ $\mathrm{MW/m^2}$, and $n_{z,U} = 10$ g, respectively. With these definitions, the optimal-control problem to be solved is complete.

IV. Adaptive Multivariate Pseudospectral Interpolation

The trajectory computation via AMPI is composed of five phases. The first two operations are performed offline, while the last three are online operations. The AMPI scheme is depicted in Fig. 2.

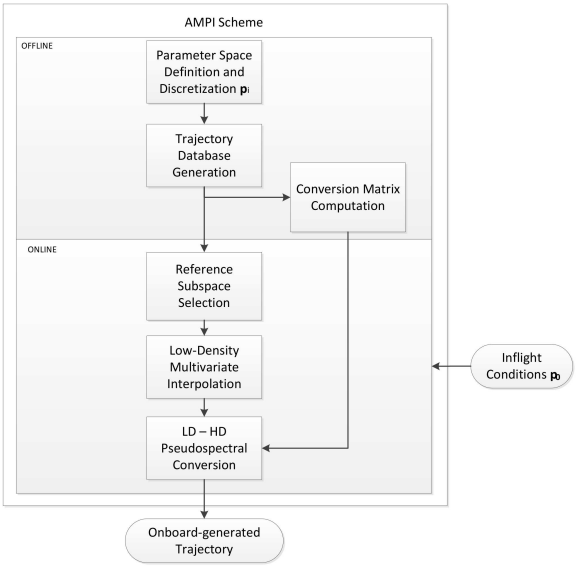


Figure 2: Scheme of Adaptive Multivariate Pseudospectral Interpolation.

The offline part involves the proper discretization of the parameters, which can be off-nominal, and determined during the flight (e.g., the states at the entry interface, provided by the navigation subsystem), and the computation of the corresponding trajectory database. It is then possible to apply the second part of the AMPI, which will run online. A specific range for each of the parameters needs to be determined and sampled, resulting in a series of discrete parameters, $\mathbf{p_i}$. Using these parameters, a corresponding series of parametric optimal-control problems is solved. This will result in a trajectory database to be stored online. A further output of the trajectory-database generation is the LD-HD conversion matrix, used, as the name suggests, to convert the low-density (LD) trajectory (less stringent in terms of on-board memory requirements) into a more meaningful high-density (HD) solution, with a process of loss-less conversion, as we will see. During the mission, the inflight parameters, $\mathbf{p_0}$, different from the nominal ones, will be analyzed to select the reference subspace from the entire trajectory database. The selected subspace will provide the basis to perform a multivariate interpolation process to compute the low-density representation of the trajectory. Finally, the previously computed LD-HD conversion matrix is used to transform the LD into the HD solution, that is, the onboard trajectory, and the reference controls. All these aspects will be explained in detail in the next subsections.

A. Definition and discretization of the parameter space

The first step is the proper definition of the parameter space. In this context we will consider as parameters the six (three components for the position, and three components for the velocity) entry-interface conditions, provided by the navigation solution. However, without loss of generality, we can refer to d parameters, and define the parameter space as composed of d vectors $\mathbf{p}_1, \ldots, \mathbf{p}_d$.

$$\mathcal{P} = \{\mathbf{p}_i\}_{i_1=1,\ldots,i_d=1}^{n_1,\ldots,n_d} \tag{4}$$

The result will be a set of parameters, which cover the entire envelope enclosed in the d-dimensional space $\mathcal{P}$. Indeed, for a complex mission, such as atmospheric entry, several inflight conditions can differ from the nominal ones, and this aspect directly affects the database size, too. According to the information provided by the responsible DLR team for the Launch and Ascent trajectory, the

following six-dimensional parameter space for the entry interface have been defined.

$$\begin{aligned} \mathbf{p}_1 &= \delta h = [-250, 0, +250] \text{ m}, \quad \mathbf{p}_4 = \delta V = [-70, 0, +70] \text{ m/s} \\ \mathbf{p}_2 &= \delta\theta = [-0.5, 0, +0.5] \text{ deg}, \quad \mathbf{p}_5 = \delta\gamma = [-0.5, 0, +0.5] \text{ deg} \\ \mathbf{p}_3 &= \delta\phi = [-0.5, 0, +0.5] \text{ deg}, \quad \mathbf{p}_6 = \delta\psi = [-0.5, 0, +0.5] \text{ deg} \end{aligned} \tag{5}$$

From Eq. (5) one can observe that three values for each of the six initial conditions are considered. Therefore, 3^6 different combinations of initial conditions are included in the trajectory database.

B. Generation of Trajectory-Database

With the definitions given in the previous section, it is now possible to modify and solve the parametric optimal-control problem defined in Sec. II. Specifically, we can use the values obtained from the discretization of the parameter space, resulting in a modification of the initial state:

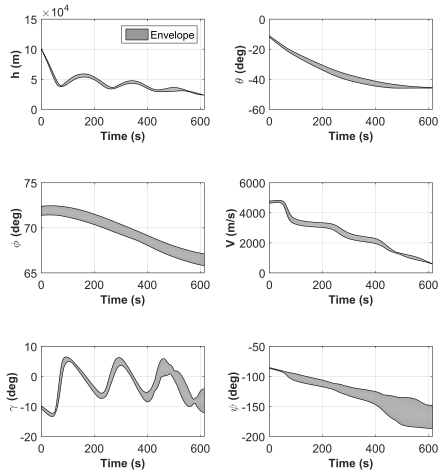
$$\mathbf{x}(t_0) = \mathbf{x}^*(t_0) + \begin{bmatrix} \delta h_i & \delta\theta_j & \delta\phi_k & \delta V_l & \delta\gamma_m & \delta\psi_n \end{bmatrix}^T, \quad i, j, k, l, m, n = [1, 2, 3] \tag{6}$$

where $\mathbf{x}^*(t_0)$ is the nominal entry interface, described in Table 1. In total, 3^6 trajectories have been computed by solving the parametric OCP defined in Eqs. (1)-(3), together with Eq. (6).

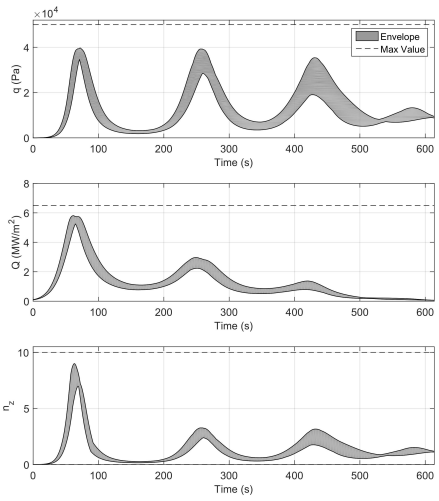
The states and the controls evolution for the entire database are depicted in Figs. 3(a) and 3(c), while the database trajectories are shown in Fig. 3(d). The constraints (satisfying the requirements) are illustrated in Fig. 3(b). In Fig. 3(d) one can see that all the trajectories terminate in the proximity of the TAEM. The circles show the parametrized dispersions for the latitude and longitude. In 3-D also the altitude parametrization would be seen, while the other three uncertainties cannot be visualized, but are taken into account, as one can see from the analysis of the single states. Small variations in the latitude and longitude were observed, but always within the limits defined by the requirements. From Figs. 3(a) and 3(c) one can see that all states and controls are smooth. Specifically, the vehicle follows an oscillating entry, clearly visible in the altitude and the flight-path angle profiles, as a result of the combination of the flight-path angle at the entry interface and the

limitations on the bank angle.

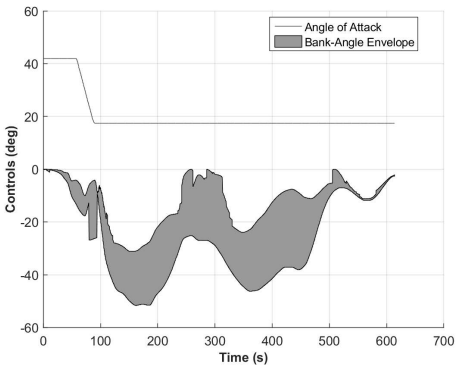
Initially the entry is dominated by the gravity, as the atmospheric density is too small to be of significance. Therefore, during the first 70 s the altitude decreases almost linearly to less than 40 km. The velocity slightly increases during these first seconds of the entry, because of the combination of the small value of D in comparison with g. Only when the drag and the lift increase, the flight-path angle tends to become smaller in magnitude; the velocity vector changes direction, and decreases in magnitude. From 70 s to 480 s the oscillating behavior is clearly visible from the evolution of the flight-path angle. After this phase the velocity is not large enough to generate a lift acceleration able to counteract the gravity, and the flight path tends to become steeper. The quasi-skip entry allows for achieving the nominal range of about 1400 km.



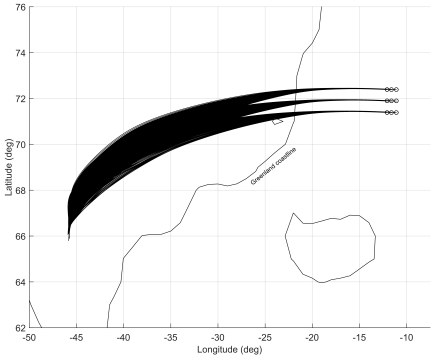
(a) States envelope.



(b) Constraints envelope.



(c) Controls envelope.



(d) Groundtracks.

Figure 3: Trajectory database - (a) states, (b) constraints, (c) controls, and (d) groundtracks.

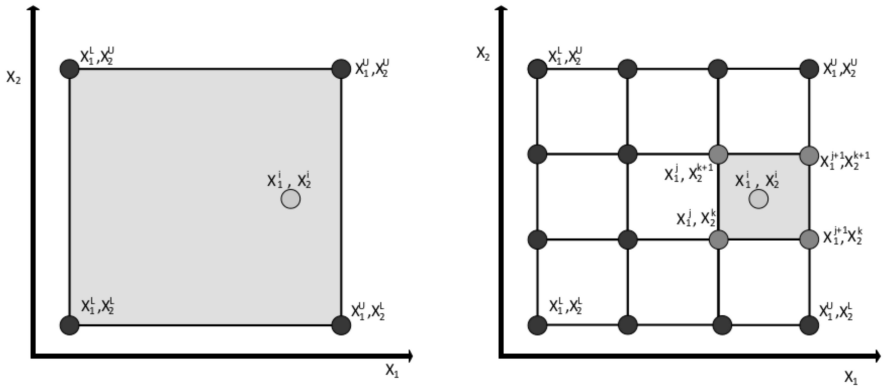
It is important to stress that pseudospectral methods belong to the local optimization schemes. Therefore, what they generate is a feasible trajectory, which *locally* satisfies the necessary conditions [28]. However, this is not an issue here, as we are interested to generate solutions, which simply satisfy our requirements, and these are verified during the trajectory-database generation. For this reason, in case more than one optimal solution exists for a given parametric optimal-control problem to be solved, the optimizer will find only one of them (the one, which results to be a *local* optimal solution) during the trajectory-database generation. Therefore, no ambiguities are possible, as they are excluded once that the database is computed. Another important point is the selection of the database points. Since the equations dominating the problem are nonlinear, it is difficult to theoretically estimate how the database points should be selected. The choice of the points can be done in practice as a trade-off between the accuracy we want to have (increasing with the number of nodes), and the data size and the time required to compute the trajectories (decreasing with the number of nodes). For the current case the database required a memory of about 12 Mb.

C. Selection of reference subspace

The d-dimensional space $\mathcal{P}$ previously defined covers the range of the possible off-nominal initial conditions taken into account. In the basic MPI approach [20], for each parameter an upper and a lower value were considered, and a corresponding optimal trajectory was generated. In that case, the database was made of 2^6 extremal trajectories, which are taken and combined to provide an approximated solution corresponding to the inflight initial conditions.

In the AMPI, for each of the initial six conditions we are considering three levels instead of two, as described by Eq. (5). Therefore the needed trajectories become 3^6 instead of 2^6. From this larger database, 2^6 neighbor-trajectories, enclosing the off-nominal initial conditions, are selected and combined to compute the corresponding trajectory. An intuitive 2-D example is depicted in Figs. 4(a) and 4(b).

In the MPI approach (Fig. 4(a)) an approximation of a function $\mathbf{F}(X_1^i, X_2^i)$ is built by using the information stored in the points $\mathbf{F}(X_1^L, X_2^L)$, $\mathbf{F}(X_1^U, X_2^L)$, $\mathbf{F}(X_1^L, X_2^U)$, and $\mathbf{F}(X_1^U, X_2^U)$. The closer the boundary points are, the more accurate the approximation $\mathbf{F}(X_1^i, X_2^i)$ is.



(a) (a) MPI approach: the trajectory is computed by using the four extremal trajectories of the database.

(b) (b) AMPI approach: the trajectory is computed by using the four closest trajectories around the off-nominal conditions.

Figure 4: Examples of domains for the application of the (a) MPI and (b) AMPI techniques.

For large dispersions this approach may not be accurate enough. In the AMPI approach the parameter space is organized into a finer grid; the function of $\mathbf{F}(X_1^i, X_2^i)$ is approximated by using the points $\mathbf{F}(X_1^j, X_2^k)$, $\mathbf{F}(X_1^{j+1}, X_2^k)$, $\mathbf{F}(X_1^j, X_2^{k+1})$, $\mathbf{F}(X_1^{j+1}, X_2^{k+1})$. The indices $[\tilde{h}^i, \tilde{h}^i + 1]$, $i = 1, \ldots, d$, representing the reference subspace, is determined by using Algorithm 1. This subspace, and not the entire space is then used for the computation of the adaptive trajectory.

Algorithm 1 - Selection of reference subspace. **Data:** Given: parameter-space elements $\mathbf{p}^i$, and initial conditions x_i, $i = 1,\ldots,d$

$\mathbf{U}_1 = [1\ 1\ 1\ 1\ 1\ 1]^T$;
for *i = 1 : d* **do**
  $\mathbf{eP}^i = \text{sign}(\mathbf{p^i} - \mathbf{U}_1 x_i)$
  for *j = 1 :* $n_i - 1$ **do**
    $\mathbf{\Delta p}_j^i = \mathbf{eP}_j^i$-$\mathbf{eP}_{j-1}^i$, $j = 2\ldots,n_i$;
    if $\sum_{j=1}^{n_i} \mathbf{eP}_j^i$==-$n_i$ **then**
      $[\tilde{h}^i, \tilde{h}^i + 1] = [n_i - 1, n_i]$;
    end
    else if $\sum_{j=1}^{n_i} \mathbf{eP}_j^i$==$n_i$ **then**
      $[\tilde{h}^i, \tilde{h}^i + 1]$ = [1,2];
    end
    else
      $idx = \mathbf{find}(\mathbf{\Delta p}_j^i \neq 0)$;
      $[\tilde{h}^i, \tilde{h}^i + 1] = [idx, idx + 1]$;
    end
  end
end
Result: $2d$ indices $[\tilde{h}^i, \tilde{h}^i + 1]$, $i = 1\ldots,d$

Once the subspace, represented by the closest 2^d trajectories to the off-nominal initial conditions, is determined, it is possible to apply the MPI as explained in the next sections.

D. Low-density multivariate interpolation

A solution of the OCP previously defined depends not only on the specific choice of a parameter vector $\mathbf{p} \in \mathcal{P}$, but also on time. Since we are using a transcription based on the flipped Radau pseudospectral (FRP) method [20–22], we propose to perform the multivariate interpolation at each of the collocation nodes defined by the roots of the flipped Radau polynomials [29]. In each discrete node τ_k, the interpolated variable $f_{int}(\tau_k, x_i)$ can be computed via Tensor-product spline [17].

$$f_{int}(\tau_k, \mathbf{x}_i) = s(\mathbf{x}_i) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_d=1}^{m_d} c_{i_1,\ldots,i_d} B_{i_1,k_1}(x_1) \ldots B_{i_d,k_d}(x_d). \tag{7}$$

Since each B-spline $B_{i_1,k_1},\ldots,B_{i_d,k_d}$ depends only on a single variable, the d-variate interpolation problem can be divided into d univariate problems, which can be solved via the numerical stable and efficient algorithm of De Boor [30]. In the frame of this work, the order k_i of the splines is equal to 2, while the knot vectors are defined as $\mathbf{t}_i = \left(t_j^i\right)_{j=1}^4 = \left(p_1^i, p_1^i, p_2^i, p_2^i\right), \;\; i = 1, \ldots, d$. They define a piecewise-linear interpolation in each direction on the given d-dimensional grid, while the coefficient matrix is given by the trajectory-database points, $\mathbf{C} = \mathbf{F}(\tau_k, \mathbf{p}_i)$, and no further effort is required to determine it. The interpolated solution is therefore evaluated by using Algorithm 2.

Algorithm 2 - Tensor-product spline interpolation.
Data: Given: knot vector $\mathbf{t}$, coefficients $\mathbf{C}$, spline $s \in \mathbb{S}_{2,\mathbf{t}_1} \bigotimes \cdots \bigotimes \mathbb{S}_{2,\mathbf{t}_d}$, evaluation point $\mathbf{x}_i \in \mathbf{P}_c$
$\mathbf{A}_0 = \mathbf{C}$;
for *i = 1 : d* **do**
  $\mathbf{A}_i$ = EvalUnivSpline($\mathbf{t}_i, \mathbf{A}_{i-1}, x_i$);
  $\mathbf{A}_i = \mathbf{A}_i'$;
  $f_{int}(\mathbf{p}) = \mathbf{A}_i$;
end
Result: interpolated values $f_{int}(\mathbf{x}_i) = f(\tau_k, \mathbf{x}_i)$

The EvalUnivSpline function in Algorithm 2 denotes the evaluation of a univariate spline with coefficients $\mathbf{C}$ at a point x_i while the operator $(\cdot)'$ performs a cyclic rotation, such that $\mathbf{A} \in \mathbb{R}^{n_1,n_2,\ldots,n_d} \Rightarrow \mathbf{A}' \in \mathbb{R}^{n_2,\ldots,n_d,n_1}$.

The interpolation scheme based on Algorithms 1 and 2 allows for generating onboard trajectories, which cover the entire range enclosed in the parameter space $\mathbf{p} \in \mathcal{P}$. While there is formal proof [16] that the interpolated states will always be defined within the boundaries given by the database itself, this may not be true for the constraints, which are nonlinear combinations of the states. However, this risk is avoided if a sufficient number of nodes is chosen. This, together with the avoidance of the Runge phenomenon [22, 31], prevents violation of any nonlinear constraint acting on the system. It is moreover important to emphasize that interpolated trajectories are not formally solutions of the underlying OCP. However, they represent a good, real-time capable, approximation of the optimal solutions without the computational burden needed to generate them.

E. LD-HD Pseudospectral conversion

The previous algorithm provides the interpolated values in a small number of nodes, having the so-called low-density discrete solution. The objective of this section is to convert the LD discrete solution into a HD discrete solution, able to represent the trajectory with no need to store large amount of data onboard. The following properties justify the choice of using the pseudospectral methods for the characterization of the discrete domain: i) Spectral convergence in the case of a smooth problem; ii) Straightforward implementation; iii) Sparse structure of the associated NLP problem; iv) Mapping between the covectors of the NLP discrete solution and the costates of the optimal continuous solution in virtue of the Pseudospectral Covector Mapping Theorem [32], and v) Removal of the Runge phenomenon [31].

The removal of the Runge phenomenon has an important implication: since all the polynomials generated using the FRP nodes do not have undesired oscillations, the interpolated solutions computed in these points will be smooth as well. Therefore, a database representing accurate trajectories can be reduced to storing the nodal values, which can be converted into a high-density discrete solution with no need to evaluate splines, as we will see in the next section. This approach significantly reduces the onboard memory requirements, as well as the onboard CPU burden.

Let us suppose to have computed the values representing the LD discrete solutions in the $N_{LD}+1$ FRP nodes (that is, the N_{LD} FRP nodes plus the node at -1. The solution is formed by the time

vector $\underline{\mathbf{t}}_{LD}$, the states $\underline{\mathbf{X}}_{LD}$, and by the controls $\underline{\mathbf{U}}_{LD}$, having dimensions equal to $n_s \times (N_{LD}+1)$ and $n_c \times (N_{LD}+1)$, respectively. The indices n_s and n_c are the number of states and controls associated with the problem under analysis. We want to transform the matrix $\underline{\mathbf{T}}_{LD}$ into a matrix $\bar{\mathbf{T}}_{HD}$, representing the HD states $\bar{\mathbf{X}}_{HD}$ and the HD controls $\bar{\mathbf{U}}_{HD}$. Accordingly, the matrix $\bar{\mathbf{T}}_{HD}$ has dimensions equal to $(n_s+n_c) \times (N_{HD}+1)$.

$$\underline{\mathbf{T}}_{LD} = \left\{ \begin{array}{c} \underline{\mathbf{X}}_{LD} \\ \underline{\mathbf{U}}_{LD} \end{array} \right\} = \left\{ \begin{array}{c} \underline{\mathbf{X}}_0, \underline{\mathbf{X}}_1, ..., \underline{\mathbf{X}}_{N_{LD}} \\ \underline{\mathbf{U}}_0, \underline{\mathbf{U}}_1, ..., \underline{\mathbf{U}}_{N_{LD}} \end{array} \right\}, \; \bar{\mathbf{T}}_{HD} = \left\{ \begin{array}{c} \bar{\mathbf{X}}_{HD} \\ \bar{\mathbf{U}}_{HD} \end{array} \right\} = \left\{ \begin{array}{c} \bar{\mathbf{X}}_0, \bar{\mathbf{X}}_1, ..., \bar{\mathbf{X}}_{N_{HD}} \\ \bar{\mathbf{U}}_0, \bar{\mathbf{U}}_1, ..., \bar{\mathbf{U}}_{N_{HD}} \end{array} \right\} \tag{8}$$

Moreover, the HD time vector $\bar{\mathbf{t}}_{HD}$ must be computed. If we apply the definition of Lagrange polynomial to a generic function $F(\tau)$, we can write

$$F(\tau) = \sum_{i=0}^{N_{LD}} F_i \prod_{\substack{k=0 \\ k \neq i}}^{N_{LD}} \frac{\tau - \underline{\tau}_k}{\underline{\tau}_i - \underline{\tau}_k}, \; \tau \in [-1, 1] \tag{9}$$

where F_i represents a generic LD variable. It can be replaced with the p^{th} row of $\underline{\mathbf{T}}_{LD}$ as they are sampled in the same way. Moreover, the continuous variable $\tau \in [-1, 1]$ can be sampled in the $N_{HD}+1$ high-density discrete nodes. The result will be the HD representation of our variables

$$\bar{\mathbf{T}}^p_{HD}(\bar{\tau}_m) = \sum_{i=0}^{N} \underline{\mathbf{T}}^p_{LD,i} \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{\bar{\tau}_m - \underline{\tau}_k}{\underline{\tau}_i - \underline{\tau}_k}, \; p = 1, \ldots, n_s + n_c, \; m = 0, \ldots, N_{HD} \tag{10}$$

Equation (10) can be extended to all the rows of the matrix $\bar{\mathbf{T}}_{HD}$, and rewritten in matrix form as

$$\bar{\mathbf{T}}_{HD} = \underline{\mathbf{T}}_{LD} \mathbf{P}_{FRP} \tag{11}$$

where the matrix $\mathbf{P}_{FRP}$ has dimensions $(N_{LD}+1) \times (N_{HD}+1)$, and is given by

$$\mathbf{P}_{FRP} = \begin{bmatrix} \prod_{k=1}^{N_{LD}} \frac{\bar{\tau}_0 - \underline{\tau}_k}{\underline{\tau}_0 - \underline{\tau}_k} & \cdots & \prod_{k=1}^{N_{LD}} \frac{\bar{\tau}_{N_{HD}} - \underline{\tau}_k}{\underline{\tau}_0 - \underline{\tau}_k} \\ \cdots & \cdots & \cdots \\ \prod_{k=0}^{N_{LD}-1} \frac{\bar{\tau}_0 - \underline{\tau}_k}{\underline{\tau}_{N_{LD}} - \underline{\tau}_k} & \cdots & \prod_{k=0}^{N_{LD}-1} \frac{\bar{\tau}_{N_{HD}} - \underline{\tau}_k}{\underline{\tau}_{N_{LD}} - \underline{\tau}_k} \end{bmatrix} \tag{12}$$

The elements $\bar{\tau}_m$ represent the high-density discrete pseudotime vector, defined between -1 and 1. Since both the LD nodes $\underline{\tau}_k$, and the HD nodes $\bar{\tau}_m$ are part of the process of the database generation (as they are part of the transcription), the matrix $\mathbf{P}_{FRP}$ can be computed offline and stored, with a significant saving in CPU time, and the trajectory synthesis is reduced to a multivariate interpolation process defined by Eq. (7) and to the matrix multiplication defined in Eq. (11). To complete the generation of the HD solution, we still need the HD discrete physical time vector associated with the interpolated solution. It can be computed as $\bar{t}_m = \frac{\underline{t}_f - \underline{t}_0}{2}\bar{\tau}_m + \frac{\underline{t}_f + \underline{t}_0}{2}$, $m = 0, \ldots, N_{HD}$, where the initial time t_0 is assumed to be known, while the final time $\underline{t}_f$ is computed by applying the multivariate interpolation approach described in Algorithms 1 and 2 to the final times stored in the trajectory database. The trajectory representing the guidance solution is completely generated with the application of the Algorithms 1 and 2, and Eq. (11).

V. Performance assessment

A. Accuracy of AMPI trajectories

In this section a measure of the accuracy of the trajectories generated via AMPI is given. The comparison between AMPI and OCP solutions is performed as follows: for each of the first 100 cases associated with the Monte-Carlo campaign of Sec. VI, the mean and the standard deviation of the maximum errors (except the final time, which has a unique value for each trajectory) are computed. They quantify the difference between the AMPI trajectories and the OCP solutions, and are given by

$$\mu_i^* = \frac{1}{n}\sum_{i=1}^{n} \max_k \left[\Delta \mathbf{x}_i(t_k)\right], \qquad \sigma_i^* = \left\{ \frac{1}{n}\sum_{i=1}^{n} \left[\max_k \left(\Delta \mathbf{x}_i(t_k)\right) - \mu_i^* \right]^2 \right\}^{1/2} \tag{13}$$

where $k \in [1, \ldots, N_{HD}+1]$ and $\mathbf{x} = [h\ \theta\ \phi\ V\ t_F]$. These results can therefore be seen as an upper boundary for the errors generated by the AMPI, and this analysis is not affected by the use of the tracking controller. Results are depicted in Fig. 5. Specifically the first plot shows the difference in altitude and velocity as relative error. The second plot shows the maximum relative error in terms of longitude and latitude. Moreover, the error in terms of final time is shown in the third plot of the same figure. Flight-path angle and heading angle have not been shown here, as they are not

constrained. All the errors are shown in percentage.

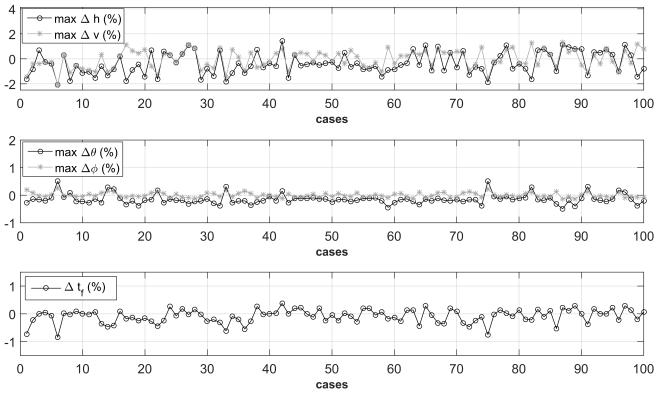


Figure 5: Accuracy assessment: AMPI vs OCP - relative errors.

From Fig. 5 it is possible to observe that the maximum error in terms of altitude and velocity are both equal to 2.1%, while it is equal to 0.5% and 0.3% in terms of longitude and latitude. The maximum values of the errors are associated with initial conditions close to the mean value of the intervals included in the database. Therefore, they are a good measure of the worst performance that the AMPI can generate. The maximum relative error in terms of final time t_f is equal to 0.8%, which corresponds to 5.2 s. This means that, despite the non-linearity of Eq. (2), a proper choice of the database size permits to capture the overall behavior of the system. Larger differences between the AMPI solutions and the true optimal trajectories can be observed in terms of flight-path angle, heading angle and bank angle (omitted in Fig. 5). This because these variables have no specified final values. This is true especially for the bank angle; while the flight-path angle and the heading angle are consequence of the flight evolution, the bank angle can be fully controlled by the vehicle, and shows in general the largest envelope among the three. This is visible in Figs. 3(a) and 3(c), where the corresponding database envelopes can be observed.

Also in absolute terms AMPI trajectories show a good correspondence w.r.t. the optimal trajectories. For instance, the mean error in terms of altitude is $\cong$ -232 m, and 1.27 m/s in terms of velocity. The same conclusions can be drawn for the mean error of longitude (0.052 deg), and latitude (0.001 deg). For what regards the final times t_f there is a mean difference of -0.5 s, with a standard deviation of 1.5 s, on a mean total time of 600 s per trajectory.

B. CPU Time analysis

For what regards the real-time capability of the proposed method, this is indirectly demonstrated by showing the CPU time required to compute the trajectory via AMPI. Also in this case the first 100 cases treated in the MC campaign described in Sec. V have been used. The results are compared with the CPU times obtained by solving the corresponding OCPs. Obviously, the absolute values cannot be used as a benchmark for onboard implementation. For this reason we refer to the AMPI as a real-time capable method, and here we focus on the relative difference on a normal machine. The tests have been repeated five times, to check the consistency of the results. However, for readability purposes, only the first three sets of data have been plotted. The CPU time analysis has been performed with MatlabR2015a, running on a laptop having a CPU i7-3687U and a clock frequency of 2.10 GHz. Results are depicted in Fig. 6.

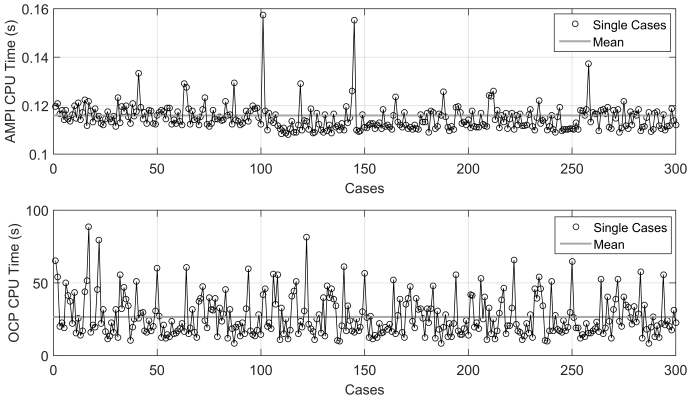


Figure 6: CPU Time analysis: AMPI vs OCP.

In the first subplot the results obtained with the AMPI are depicted. The black circles represent the CPU times associated with the single runs, while the gray line is the mean value, equal in this case to 116 ms, with a minimum value of 108 ms, and a maximum value of 253 ms (not shown in Fig. 6). The standard deviation is equal to 13 ms. In the second subplot the same information, associated with the corresponding OCPs, are plotted. The mean time required to solve the OCP is 26.5 s, with a standard deviation of 14.0 s. Minimum and maximum values observed in this case are equal to 8.3 and 88.5 s, respectively.

If we refer to the mean values, the AMPI algorithm is about 229 times faster than the algorithm

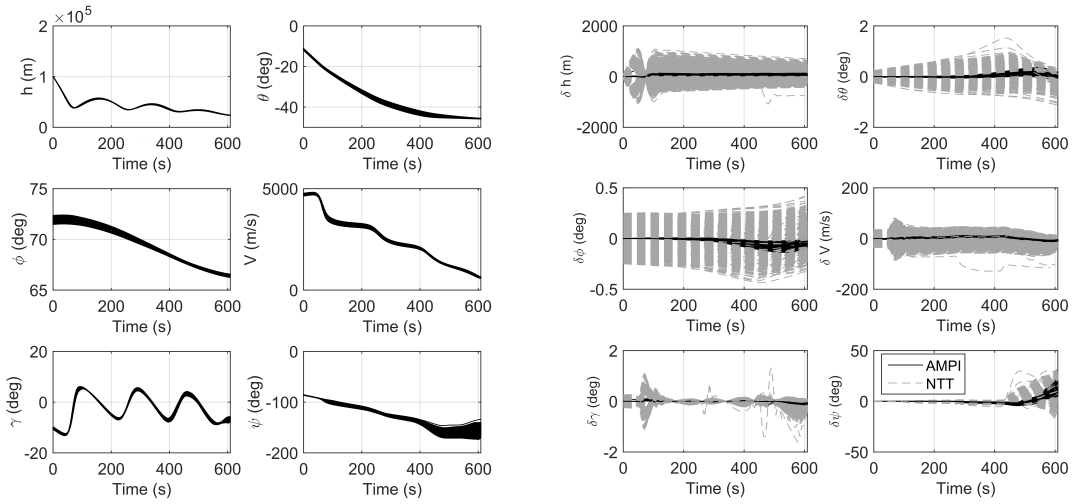
required to solve the corresponding OCP. Moreover, the values of the standard deviation give us indications about the real-time capability of the proposed method. Indeed, when the AMPI is used, the small standard deviation confirms reduced variations in the execution time required to generate a solution. In this case the number of operations required can be predetermined, and does not depend on the specific values that the initial conditions assume. Therefore, the obtained CPU times are very similar to each other. When the optimal control theory is used instead, larger variations in the execution time are expected, as it is not possible in general to predict the number of iterations required to solve a non-convex optimal problem as the one defined in Sec. III.

VI. Numerical Results

A. Simulation Campaign

For the validation of the method, a Monte Carlo campaign of 1000 cases has been simulated. For each case, random dispersions (3σ) consistent with the ranges of Eq. (5) have been used. Results are compared with a so-called neighbor-trajectory tracking (NTT) approach. In this case, given the initial conditions, the trajectory belonging to the database, having the closest initial conditions to the current ones, is selected as reference solution. A common tracking controller has been added to the guidance schemes. The states, the state errors, the controls, and the groundtracks are compared and plotted in Figs. 7(a)-8(c), together with the histograms plotted in Figs. 9(a)-9(c), which show the differences in terms of final altitude, velocity and range-to-go. Moreover, the footprints are showed in 9(d).

The multivariate approach generates meaningful trajectories, as can be seen in Fig. 7(a), and significantly reduce the error (plotted in Fig. 7(b)) that has to be handled by the feedback controller. This is well visible from Figs. 8(c) and 9(d), where the trajectories and the final footprints (represented as crossrange and range-to-go) are plotted. These figures show a large improvement of performance when the AMPI is used, w.r.t. the NTT approach. To quantify the results, three dispersion circles, having radii equal to 25, 50 and 75 km, have been defined. The dispersion area of the cases associated with the use of the AMPI technique is about 4.2% of the area obtained by using the same feedback controller to track the NTT solution. Another consequence is that the online



(a) States.

(b) States errors.

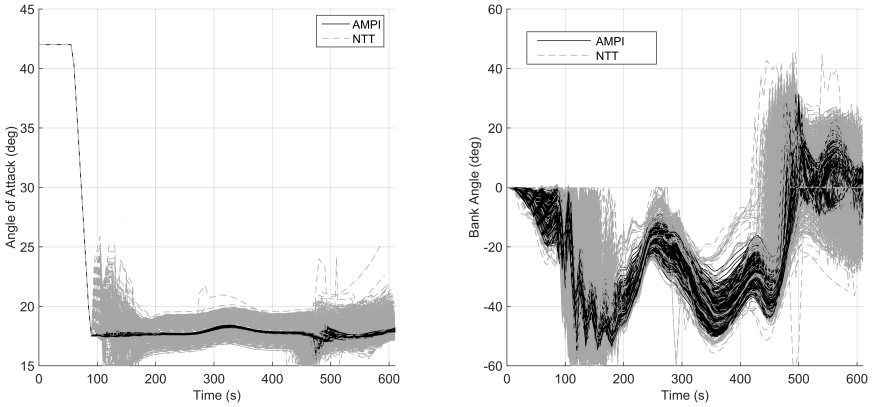
Figure 7: MC campaign (N = 1000): AMPI vs NNT approach - (a) state, (b) state errors.

adaptation of the trajectory simplifies the work of the feedback controller (Figs. 8(a) and 8(b)). Indeed, the proposed technique allows to have smoother angle-of-attack and bank-angle profiles, and in general a reduced control activity w.r.t. the nominal tracking, as the feedback control needed for the compensation of the errors is significantly reduced.

In terms of altitude a reduced dispersion associated with the use of the AMPI is shown in Fig. 9(a). In the worst cases, there is an error of $\cong$ 200 m w.r.t. the reference value when the AMPI is used, while this value can be up to 900 m for the NTT. In terms of final velocities, both the systems achieve good performance, (a maximum error of about 2 m/s in both cases). Completely different are the results in terms of range-to-go, as shown in the histogram of Fig. 9(c). Indeed, 740 cases fall within a distance of less than 6.5 km w.r.t. the nominal target point when the AMPI is used, despite the large initial dispersions, against 270 cases corresponding to the NTT. The cases associated with the AMPI method go up to 963 cases against 389 (NTT) if a radius of 10.7 km is considered.

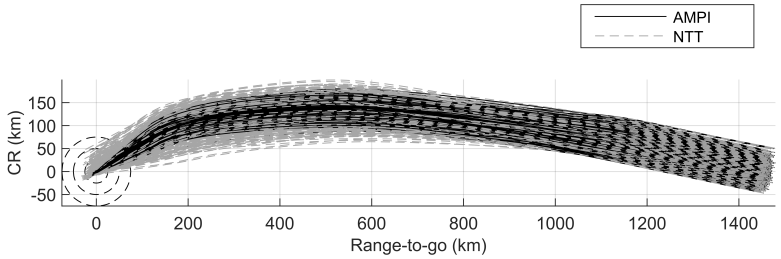
Table 3: Dispersion Analysis - 1000 MC runs.

Ellipse / Controller	$[25 \times 25]$	$[50 \times 50]$	$[75 \times 75]$	Outside
AMPI	1000	0	0	0
NTT	834	164	2	0



(a) Angle of attack.

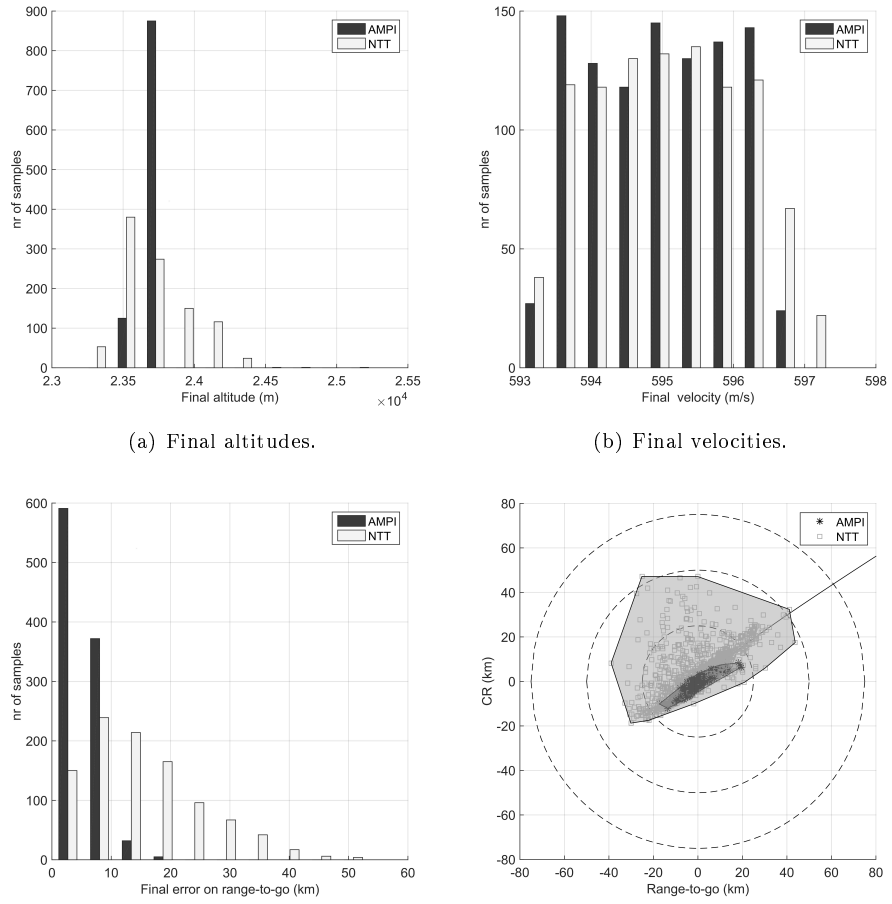
(b) Bank angle.



(c) Trajectories.

Figure 8: MC campaign (N = 1000): AMPI vs NNT approach - (a) angle of attack, (b) bank angle, and (c) groundtracks.

Table 3 lists the results of the Monte Carlo campaign in terms of distance w.r.t. the nominal final position. It is possible to see that 100% of the cases fall into the finest ellipse, i.e., within a radius of 25 km versus 83.4% of the cases associated to the use of the nominal tracking. Indeed, in the former case the dispersion area is $\cong$ 214 km^2 while in the case of nominal tracking the resulting area is $\cong$ 3,675 km^2. The two areas are depicted in Fig. 9(d). Finally, for both the methods, the constraints are always satisfied.



(a) Final altitudes.

(b) Final velocities.

(c) Final range-to-go.

(d) Footprints.

Figure 9: MC campaign (N = 1000): AMPI vs NNT approach - (a) angle of attack, (b) bank angle, and (c) footprints.

VII. Conclusions

In this work the multivariate pseudospectral interpolation approach has been coupled with an algorithm of subspace selection to be able to generate online nearly-optimal real time trajectories for entry scenarios in presence of wide dispersions at the entry interface. The Monte-Carlo campaign has demonstrated the feasibility of this approach, having as further advantage a significant improvement in the guidance performances, analyzed both in terms of longitudinal error, and in terms of footprint dispersion, reduced to $\cong 6\%$ of the one obtained with a more traditional NTT approach.

A further advantage coming from the use of this technique is its lossless reduction of the database size. High-density discrete trajectories can be obtained by storing a significantly smaller number of values, reducing the size to $\cong 3\%$ of a database stored in traditional way, and this aspect directly

relaxes the requirements for the memory of the onboard CPU. This aspect, together with the good performance obtained, suggest that the proposed method can be a good choice for scenarios having large dispersions at the entry interface.

References

[1] Harpold J., C. and Graves C. A., "Shuttle Entry Guidance", *Journal of the Astronautical Sciences*, Vol.27 No. 3, 1979

[2] Lu P., "Entry Guidance: a Unified Method", *Journal of Guidance, Control, and Dynamics*, Vol.37 pp. 713-728, 2014, doi:10.2514/1.62605

[3] Mease K. D., Chen D. T., Teufel P. and Schöneberger H., "Reduced-Order Entry Trajectory Planning for Acceleration Guidance", *Journal of Guidance, Control and Dynamics*, Vol.25 No. 2, pp. 257-266, 2002, doi:10.2514/2.4906

[4] Mooij E.: "Adaptive Heat-Flux Tracking for Re-entry Guidance," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA 2014-4144, San Diego, CA, 2014, doi:10.2014/6.2014-4142.

[5] Lu P. and Hanson J. M., "Entry Guidance for the X-33 Vehicle", *Journal of Spacecraft and Rockets*, Vol.35 No. 3, pp. 342-349, 1998, doi:10.2514/2.3332

[6] Tu K., Munir M.S. and Mease K. D., "Drag-Based Predictive Tracking Guidance for Mars Precision Landing", *Journal of Guidance, Control and Dynamics*, Vol.23 No. 4, pp. 620-628, 2000, doi:10.2514/2.4607

[7] Mooij E., "Robustness Re-entry Guidance and Control System Design and Analysis", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2007-6779, Hilton Head, SC, 2007, doi:10.2514/6.2007-6779

[8] Betts J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2^{nd} ed.*, SIAM, Philadelphia, 2010

[9] Bollino K. P.,"High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles", *Ph.D. Dissertation*, Mechanical and Astronautical Engineering Dept., Naval PostGraduate School., 2006

[10] Singh B. and Bhattacharya R., "Optimal Guidance of Hypersonic Vehicles Using B-Splines and Galerkin Projection", *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA 2008-7263, Honolulu, HA, 2008, doi:10.2514/6.2008-7263

[11] Marwaha M, Singh B. Valasek J., and Bhattacharya R., "Integrated Guidance and Fault Tolerant Adaptive Control for Mars Entry Vehicle", *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA 2009-5668, Chicago, IL, 2009, doi:10.2514/6.2009-5668

[12] Gill, P. E., Murray W. and Saunders M. A., "User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming"', Software User Manual, Department of Mathematics, University of California, San Diego, CA, 2008

[13] Wächter A. and Biegler L.T., "On the implementation of an interior-point filter linesearch algorithm for large-scale nonlinear programming", *Math. Program. 106(1)*, Springer-Verlag, New York, 2006.

[14] Saraf A., Levitt J.A., Mease K. D. and Ferch M., "Landing footprint computation for entry vehicles", *AIAA Guidance, Navigation and Control Conference and Exhibit*, Providence, RI, 2004, AIAA-2004-4774, doi:10.2514/6.2004-4774

[15] Schierman J.D., Ward D. G., Hull J. R., Gandhi N., Oppenheimer M. W., and Doman D. B., "Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight-Test Results", *Journal of Guidance, Control, and Dynamics*, Vol.27 pp. 975-988, 2004, doi:10.2514/1.10344

[16] Lockner E., Oehlschlägel T., Theil S., Knauer M., Tietjen J. and Büskens C., "Real-Time capable trajectory synthesis via multivariate interpolation methods for a moon landing manoeuvre", *CEAS Space Journal*, DOI 10.1007/s12567-014-0063-z, 2014

[17] Lyche T. and Morken K., "Spline Methods", Department of Informatics, University of Oslo. available at http://www.uio.no/studier/emner/matnat/ifi/INF-MAT5340/v11/undervisningsmateriale/book.pdf, 2011

[18] Arslantas Y. E., Oehlschlägel T., Sagliano M., Theil S. and Braxmaier C., "Approximation of Attainable Landing Area of a Moon Lander by Reachability Analysis"*17-th International Conference and Control (HSSC)*, Berlin, Germany, 2014

[19] Arslantas Y. E., Oehlschlägel T., Sagliano M., Theil S. and Braxmaier C., "Safe Landing Area Determination for a Moon Lander by Reachability Analysis", *International Astronautical Conference. IAC-14-C.1.7.2*, Toronto, Canada, 2014

[20] Sagliano M., Oehlschlägel T., Theil S. and Mooij E., "Real time adaptive feedforward guidance for entry vehicles", *3rd CEAS Eurognc conference*, Toulouse, 2015

[21] Sagliano M., "Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems", *Operations Research Letters*, Volume 42, Issue 3, 2014, doi: 10.1016/j.orl.2014.03.003

[22] Sagliano M. and Theil S., "Hybrid Jacobian Computation for Fast Optimal Trajectories Generation", *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA 2013-4554, Boston, MA, 2013, doi:10.2514/6.2013-4554

[23] Sagliano M.,"Development of a Novel Algorithm for High Performance Reentry Guidance", *Ph.D. Dissertation*, Fachbereich Produktionstechnik , University of Bremen, 2016, urn:nbn:de:gbv:46-00105082-13

[24] Huneker, L., Sagliano M. and Arslantas Y.E., "SPARTAN: An Improved Global Pseudospectral Algorithm for High-Fidelity Entry-Descent-Landing Guidance Analysis", *30th International Symposium on Space Technology and Science*, Kobe, Japan, 2015

[25] Department of Defense, "World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems", Technical Report, 3rd ed., January 2000.

[26] NOAA, "U.S. Standard Atmosphere 1976", Technical Report NASA-TM-X-74335, NOAA-S/T 76-1562 October 1976.

[27] Sagliano M., Samaan M., Theil S. and Mooij E., "SHEFEX-3 Optimal Feedback Entry Guidance", *AIAA SPACE 2014 Conference and Exposition*, AIAA 2014-4208, San Diego, CA, 2014, doi:10.2514/6.2014-4208

[28] Ross I. M., *A Primer on Pontryagin's Principle in Optimal Control: 2nd ed.*, Collegiate Publishers, ISBN 978-0-984-35711-6, 2015

[29] Sagliano. M., Mooij E. and Theil S., "Onboard Trajectory Generation for Entry Vehicles via Adaptive Multivariate Pseudospectral Interpolation", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2016-2115, San Diego, CA, 2016, doi:10.2514/6.2016-2115

[30] De Boor C., *A Practical Guide to Splines*, Springer, New York, 2001

[31] Garg D., "Advances in Global PseudoSpectral Methods for Optimal Control", *Ph.D. Dissertation*, Department of Mechanical and Aerospace Engineering., University of Florida, FL, 2011.

[32] Gong Q., Ross I. M., Kang W. and Fahroo F.,"Connections Between The Covector Mapping Theorem and Convergence of Pseudospectral Methods for Optimal Control", *Comput Optim Appl*, 2008, doi: 10.1007/s10589-007-9102-4