CrossMark
click for updates

Onboard Trajectory Generation for Entry Vehicles via Adaptive Multivariate Pseudospectral Interpolation

Marco Sagliano [*]

Deutsches Zentrum für Luft- und Raumfahrt, Robert Hooke Straße 7, Bremen, Germany, 28359

Erwin Mooij [†]

Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, The Netherlands

Stephan Theil [‡]

Deutsches Zentrum für Luft- und Raumfahrt, Robert Hooke Straße 7, Bremen, Germany, 28359

One of the most powerful analysis tools to deal with entry-guidance problems is the possibility to formulate them as optimal control problems. Environmental constraints, actuator limits, and strict requirements on the final conditions can be efficiently transcribed, resulting in a discrete, finite-dimension nonlinear programming (NLP) problem. However, NLP problems require a computational power, which often exceeds the vehicle's onboard capabilities. Moreover, it is important to ensure that the solution can be adapted to the actual flight conditions, which can differ from the nominal scenario. This paper proposes an approach based on an efficient use of multivariate pseudospectral interpolation scheme to generate real-time capable entry guidance solutions. The proposed onboard trajectory generation algorithm is able to deal with wide dispersions at the entry interface, and can improve the lateral performance in cases where the classic bank-reversal logic is not sufficient. The interpolation is applied to subspaces of a database of pre-computed trajectories, which can be efficiently stored onboard. The method is here proposed for initial-conditions variations, but can be applied to every mission parameter, which allows to find a corresponding optimal solution. Results have been generated for SHEFEX-3, an entry demonstrator vehicle, which was planned by the German Aerospace Center. Monte-Carlo simulations show how this approach is applicable, and yields significant improvements both in longitudinal and lateral guidance performance, with an improvement of the dispersion area of about 96%.

Nomenclature

Roman

$\mathbf{C}$	Matrix of coefficients
C_D	Drag coefficient
C_L	Lift coefficient
D	Drag acceleration (m/s^2)
g	Gravity acceleration (m/s^2)
h	Altitude (m)
$\tilde{h}_i$	i^{th} index of selected reference subspace
J_2	zonal harmonic
k_q	Heat flux coefficient (kg$^{1/2}$/m^3)

---

[*]Research Engineer, Guidance, Navigation and Control Department, PhD Candidate at University of Bremen, AIAA Member

[†]Assistant Professor, Astrodynamics and Space Missions Department, Faculty of Aerospace Engineering, AIAA Associate Fellow

[‡]Department Head, Guidance, Navigation and Control Department

American Institute of Aeronautics and Astronautics

L	Lift acceleration (m/s^2)
L_n	n$^{\text{th}}$ Legendre polynomial
m	Mass (kg)
M	Mach number
n_c	Number of controls
n_G	number of elements of the selected subspace
n_s	Number of states
n_z	Vertical load factor
N	Number of sampled points
N_{HD}	High-density number of nodes
N_{LD}	Low-density number of nodes
$\mathbf{p}$	parameter
$\mathcal{P}$	discretized parameter space
$\mathcal{P}_c$	parameter space
P_i	i$^{\text{th}}$ interpolant
$\mathbf{P}_{FRP}$	Low-density / High-density conversion matrix
$\bar{q}$	Dynamic pressure (N/m^2)
$\dot{Q}$	Heat flux (W/m^2)
r	Radial position (m)
R_e	Earth's equatorial radius (m)
$\mathbb{R}_i$	i^{th} set of real numbers
s	Generic spline
S	Area (m^2)
t	Time (s)
$\tilde{\mathbf{t}}_{HD}$	High-density discrete time vector (s)
$\mathbf{t}_{LD}$	Low-density discrete time vector (s)
$\mathbf{t}_i$	i$^{\text{th}}$ knot vector
T	Temperature (K)
T_h	Temperature gradient (K/m)
$\tilde{T}_{HD}$	High-density discrete solution
T_{LD}	Low-density discrete solution
u_σ	Bank angular velocity (rad/s)
$\mathbf{U}$	Vector with elements equal to 1
$\tilde{U}_{HD}$	High-density discrete controls
U_{LD}	Low-density discrete controls
V	Velocity modulus (m/s)
X	Domain point
$\tilde{X}_{HD}$	High-density discrete states
X_{LD}	Low-density discrete states

Greek

α	Angle of attack (rad)
γ	Flight-path angle (rad)
$\mu_\oplus$	Earth's gravitational parameter (m^3/s^2)
ϕ	Latitude (rad)
ψ	Velocity-azimuth angle (rad)
ρ	Atmospheric density (kg/m^3)
τ	Pseudospectral domain
θ	Longitude (rad)
w_ϕ	Latitude weight (1/rad)
w_θ	Longitude weight (1/rad)
w_h	Altitude weight (1/m)
w_V	Velocity weight (s/m)

Operators

$\dot{()}$	first time derivative $(()/s)$
$\ddot{()}$	second time derivative $(()/s^2)$
$()_{cur}$	Current value
$()_{f}$	Final value
$()_{ref}$	Reference value
$()_{U}$	Maximum value

Abbreviations

AMPI	Adaptive Multivariate Pseudospectral Interpolation
CPU	Central Processing Unit
DLR	Deutsches Zentrum für Luft- und Raumfahrt
FRP	Flipped Radau Pseudospectral
HD	High Density
LD	Low Density
NLP	Nonlinear Programming
OCP	Optimal Control Problem
SHEFEX	SHarp Edge Flying EXperiment
MPI	Multivariate Pseudospectral Interpolation
PS	Pseudospectral
TAEM	Terminal Area for Energy Management
US76	U.S. Standard Atmosphere 1976
WGS84	World Geodetic System 1984

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[1–3]), 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,[4] extracting the longitudinal states (altitude, speed, flight-path angle) from the drag acceleration and its derivatives,[5] or implementing constraints-tracking guidance schemes.[6,7] Similar results can be obtained if the drag-velocity plane is replaced by the drag-energy plane.[8–10] In any case, approximations, disturbances and modeling errors make the use of a feedback controller necessary to track the scheduled nominal 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[11–13] 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.[14–16] 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, significant off-nominal conditions can deteriorate the performance of the system, or in the worst case, threaten the mission success.

In this paper an approach to generate onboard a trajectory corresponding to the off-nominal initial conditions, and the controls to track it, is proposed. This is done on an efficient use of a database of pre-computed trajectories. Significant steps in this direction have already been performed. Saraf[17] uses interpolation schemes applied to extremal drag-energy profiles for generating landing footprints for entry missions. Lockner[18] developed a more extensive approach based on Tensor Product Splines,[19] which perform excellent

American Institute of Aeronautics and Astronautics

for the Lunar Landing problem. Arslantas[20, 21] used a similar technique for reachibility-set computations. Sagliano[22] merged this approach with pseudospectral methods to provide a real-time capable method able to deal with off-nominal, but limited initial conditions. In this last work a database of trajectories is stored onboard and combined to obtain an onboard-generated solution (i.e., the trajectory and the reference controls) able to deal with the six off-nominal initial 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 approximation 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, but the MPI approach is applied only to a part of it, which is properly selected by the onboard software according to the navigation solution provided at the entry interface. The result is a highly-adaptive onboard trajectory-generator, which satisfies the tough requirements of memory and power of the onboard computers even in presence of larger entry-interface dispersions.

A strong advantage of this method is also associated with the capability to deal with asymmetric entry scenarios, which is outside the capabilities of the classic bank-reversal logic. 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 computed, by performing a new computation of the database with no modifications of the flight software.

The verification of the method is performed by coupling the method with a feedback controller,[23] and comparing the results with those obtained by the tracking of the nominal solution with 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 reported 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 the simulation campaign results are reported and discussed. Section VI, finally, concludes this paper with final remarks and a brief outlook on future work.

II. Reference Mission Description - SHEFEX-3

The reference scenario is one of the proposed mission profiles for SHEFEX-3. 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. An alternative scenario with the terminal point in the Svalbard Archipelago has already been analyzed.[27] After the stages separation and the coast phase, the unpowered descent phase follows. A scheme of the SHEFEX-3 mission profile is depicted in Fig. 1.

Once an altitude of 100 km is reached, the nominal entry phase begins. 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.[5, 8, 12] The mission, from the point of view of the guidance system, terminates at the TAEM, for this mission associated with a Mach number equal to $\cong 2$. This requires good accuracy in terms of final altitude and final speed. The nominal entry and terminal conditions are shown in Table 1.

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

State	Initial Value	Terminal Value
h (km)	100.10	23.50 ± 2.5
θ (deg)	-11.60	-45.75
ϕ (deg)	71.89	66.40
V (m/s)	4712.26	595.00 ± 25
γ (deg)	-10.31	free
ψ (deg)	-85.92	free

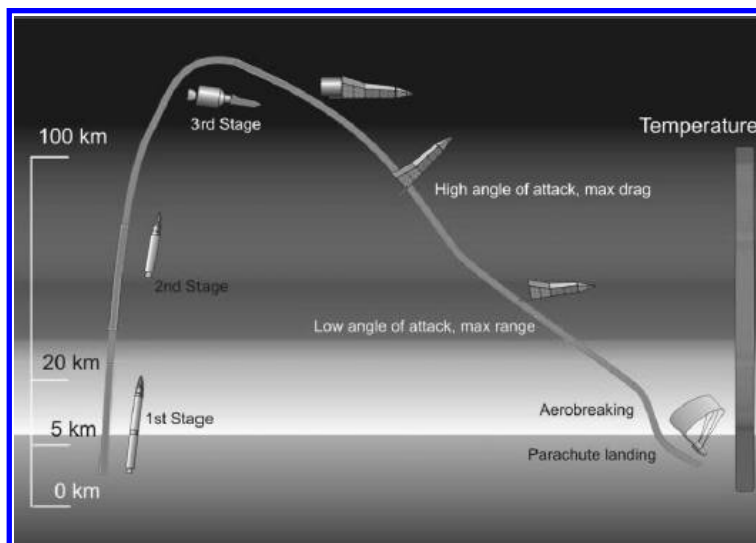American Institute of Aeronautics and Astronautics

Figure 1: SHEFEX-3 Mission Profile.

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, at 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, the gravity acceleration is derived from the World Geodetic System 1984 (WGS84) model[25] as central field with only the J_2 term. The atmosphere is modeled with the U.S. Standard Atmosphere 1976 (US76) model.[26]

The aerodynamics model has been derived by the DLR Institute of Aerodynamics and Flow Technology. It takes friction into account, and has been validated by using Navier-Stokes equations. To make the analysis more realistic, the equations of motion take the Earth's rotation into account as well. 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. This profile generates large heat flux during the first phase of the entry, and allows to test the thermal protection system deveped for SHEFEX. 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*. It is worth mentioning that in case bank-angle accelerations exceed the limits, they can be introduced in the transcription as well, but for the current mission scenario this step was not necessary. Reference values for the controls are shown in Table 2.

A.   Cost Function

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

$$J = w_\theta(\theta_f - \theta_{ref})^2 + w_\phi(\phi_f - \phi_{ref})^2 + w_h(h_f - h_{ref})^2 + w_V(V_f - V_{ref})^2 \tag{1}$$

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 are assumed to be equal to 1 for the longitude and the latitude, and equal to 10^{-4} for the altitude and the velocity.

American Institute of Aeronautics and Astronautics

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 manoeuver $t_{\alpha,U}$ (s)	58
End of α manoeuver $t_{\alpha,L}$ (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]

B. Dynamics

During entry, the vehicle's motion is described by the following set of differential equations[28]

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

where h and r are the altitude and the radial position, respectively, θ and ϕ are the longitude and the latitude, V is the velocity modulus, γ and ψ are the flight-path angle and the velocity azimuth angle, the latter 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. As one can see from the last relationship shown in Eqs. (2), the state has been augmented by adding the bank angle. This allows to limit the bank-angle rate, which is the effective control input to the system.

C. State Boundaries

As for the controls, the states are bounded too. Since the problem is not affected by the boundaries, full ranges are taken, i.e.,

$$
\begin{Bmatrix}
0 \text{ km} \\
-180° \\
-90° \\
10 \text{ m/s} \\
-90° \\
-180°
\end{Bmatrix}
\leq
\begin{Bmatrix}
h \\
\theta \\
\phi \\
V \\
\gamma \\
\psi
\end{Bmatrix}
\leq
\begin{Bmatrix}
120 \text{ km} \\
180° \\
90° \\
7000 \text{ m/s} \\
90° \\
180
\end{Bmatrix}
\tag{3}
$$

American Institute of Aeronautics and Astronautics

D.   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 included in the analysis can be computed according to

$$\overline{q} = \frac{1}{2}\rho V^2$$
$$\dot{Q} = k_q \sqrt{\rho} V^3 \tag{4}$$
$$n_z = \frac{|L\cos\alpha + D\sin\alpha|}{g_0}$$

where ρ is the atmospheric density, expressed in 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 $1.2444 \cdot 10^{-3}$ kg$^{1/2}$/m^3, and g_0 is the gravity acceleration at sea level, ($g_0 = 9.782$ m/s^2). The structural limits of the vehicle and the active thermal protection system imply a boundary for the constraints previously mentioned. These boundaries are equal to 10^4 N/m^2, 6.5 MW/m^2, and 10 g, respectively. With these definitions, the optimal-control problem to be solved is completely characterized.

IV.   Adaptive Multivariate Pseudospectral Interpolation

In this section the AMPI algorithm is explained. The trajectory computation via AMPI is composed of five phases. The first two operations are performed offline, while the last three are online operations, which can be performed according to the limits of the flight hardware. The working scheme is depicted in Fig. 2. We can distinguish two subparts in the scheme: an offline part, and an online part. 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 $\mathbf{p_i}$. Accordingly to these parameters, a corresponding series of parametric optimal-control problems is solved. This will result in the 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 entry-interface conditions, provided by the navigation solution. This is not the only choice, since every parameter that can be estimated on-board and that allows to compute a corresponding optimal solution can be potentially treated with this method. In this case, we consider six different off-nominal initial conditions (three components for the position, and three components for the velocity), but to keep the method general, let us refer to d parameters as $p^1, \ldots, p^d$. Each of the parameters is defined on a compact subset of the real numbers $\mathbb{R}_i \subset \mathbb{R}, i = 1, \ldots, d$, such that the Cartesian product

$$\mathbb{R}^d \supset \mathcal{P}_c = \prod_{i=1}^{d} \mathbb{R}_i = \mathbb{R}_1 \times \cdots \times \mathbb{R}_d, \tag{5}$$

defines the parameter space $\mathcal{P}_c$. In this context a solution of the OCP can be interpreted not only as a function of time, but also as function of a d-dimensional parameter vector. In the following each one-

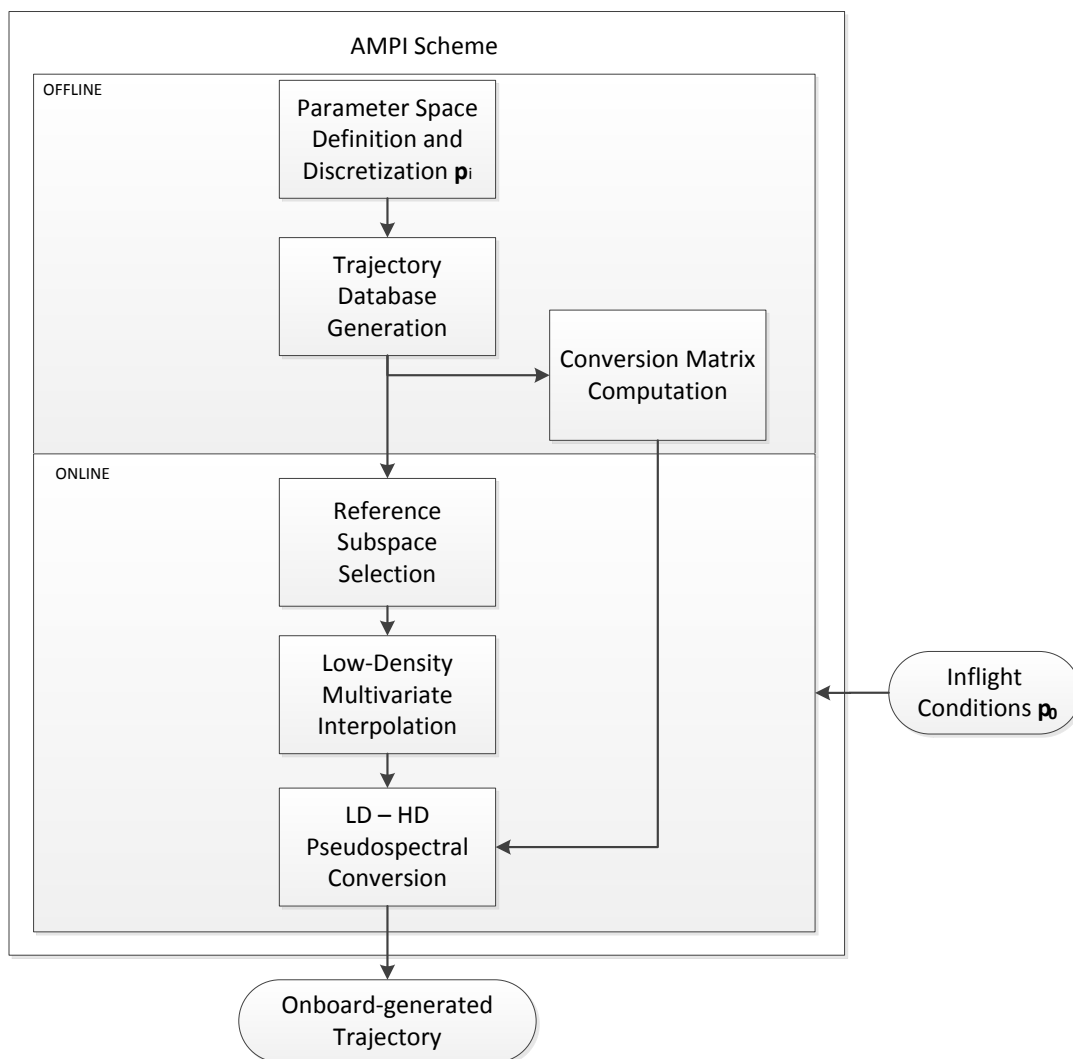American Institute of Aeronautics and Astronautics

Figure 2: Scheme of Adaptive Multivariate Pseudospectral Interpolation.

dimensional domain of a single reference parameter is discretized using a finite number of discrete points. Therefore, let

$$\mathbf{p}^i = \left\{ p_1^i, \ldots, p_{n_i}^i \right\} \tag{6}$$

define a strictly monotonically increasing set for each $i = 1, \ldots, d$. The Cartesian product of the sets given by Eq. (6) defines a d-dimensional $n_1 \times \cdots \times n_d$-rectangular grid

$$\mathcal{P} = \prod_{i=1}^{d} \mathbf{p}^i = \mathbf{p}^1 \times \cdots \times \mathbf{p}^d \tag{7}$$

which can be seen as a discretization of the parameter space $\mathcal{P}_c$ defined by Eq. (5). The set $\mathcal{P}$ consists of $n_G = \prod_{i=1}^{d} n_i$ elements and can equivalently be represented as a combination of all the grid points $\mathbf{p}_i$, where $\mathbf{p}_i = \left(p_{i_1}^1, \ldots, p_{i_d}^d \right) \in \mathbb{R}^d$, such that

$$\mathcal{P}_c \cong \mathcal{P} = \{ \mathbf{p_i} \}_{i_1=1,\ldots,i_d=1}^{n_1,\ldots,n_d}. \tag{8}$$

American Institute of Aeronautics and Astronautics

The parameter space is therefore completely defined. For each parameter, the related OCP can be redefined and solved. The result will be a set of parameters, which cover the entire d-dimensional space $\mathcal{P}_c$. Indeed, for a complex mission, such as the atmospheric entry, several inflight conditions can differ from the nominal ones, and this aspect directly affects the database size, too.

How can we set the parameters space for the atmospheric entry guidance? We can characterize the uncertainty on the initial states from a purely geometrical point of view. Indeed, a 1-D region of interest X can be represented as a straight line connecting two nodes representing the extreme values that this particular variable can assume, see Fig. 3(a). The extension of this region to two dimensions X, Y is geometrically represented by a rectangle (or in an easier way, by a square if the variables are properly normalized), where the vertices are the 2^2 possible combinations of extreme values that the variables X and Y can assume (Fig. 3(b)).

In three dimensions X, Y, Z, we will have a cube, which vertices represent the 2^3 possible combinations of parameters (Fig. 3(c)). Since the initial state of the vehicle at the entry interface is represented by the three components of position and the three components of speed, we will have a six-dimensional region of interest, which can be different from their corresponding nominal values. We can describe this multidimensional uncertainty as a *hexeract* (Fig. 3(d)), which is a member of the hypercube family, characterized by having a dimension equal to six.
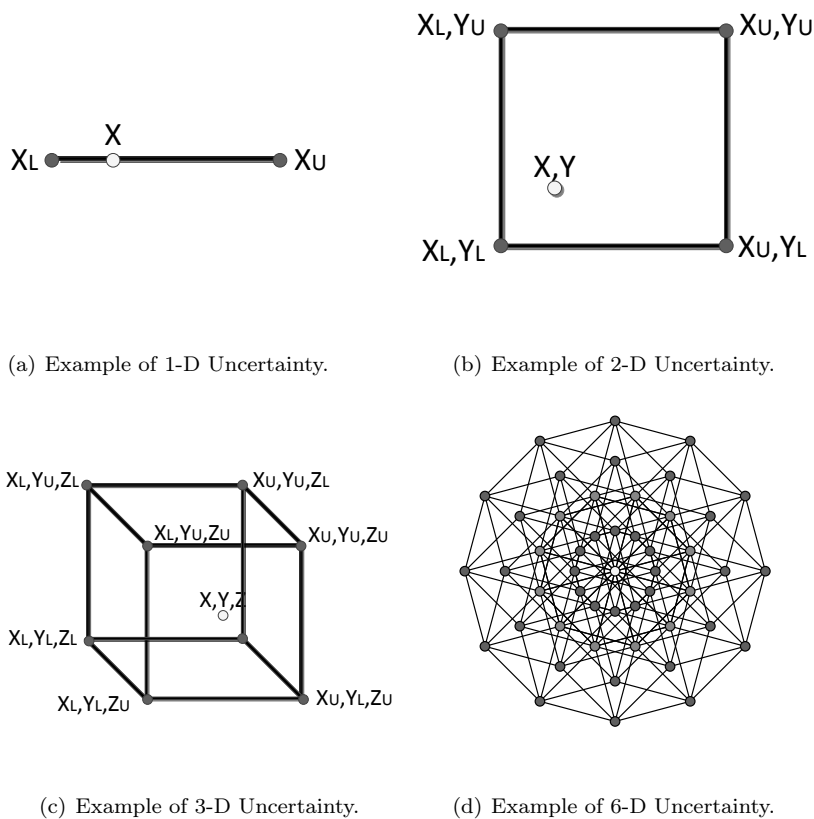

(a) Example of 1-D Uncertainty.

(b) Example of 2-D Uncertainty.

(c) Example of 3-D Uncertainty.

(d) Example of 6-D Uncertainty.

Figure 3: Uncertainties in 1-D, 2-D, 3-D, and 6-D.

A practical example of d-dimensional parameter space, with $d = 6$, is shown in Eq. (9). The hypervertices of the hexeract represent the extreme initial conditions that the vehicle can experience at the entry interface, while its hypervolume encloses all the possible initial states. The nominal conditions for SHEFEX-3 will represent then the hypercenter of our hexeract. According to the information provided by the responsible DLR team for the Launch and Ascent trajectory, the following parameter space for the entry interface has

been defined.

$$
\begin{aligned}
\mathbf{p}_1 &= \delta h = [-250, 0, +250] \text{ m} \\
\mathbf{p}_2 &= \delta\theta = [-0.5, 0, +0.5] \text{ deg} \\
\mathbf{p}_3 &= \delta\phi = [-0.5, 0, +0.5] \text{ deg} \\
\mathbf{p}_4 &= \delta V = [-70, 0, +70] \text{ m/s} \\
\mathbf{p}_5 &= \delta\gamma = [-0.5, 0, +0.5] \text{ deg} \\
\mathbf{p}_6 &= \delta\psi = [-0.5, 0, +0.5] \text{ deg}
\end{aligned}
\tag{9}
$$

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 now use the values obtained from the discretization of the parameter space. In this context, this means that we need to modify the initial state as

$$
\mathbf{x}(t_0) = \mathbf{x}^*(t_0) +
\begin{bmatrix}
\delta h_i \\
\delta\theta_j \\
\delta\phi_k \\
\delta V_l \\
\delta\gamma_m \\
\delta\psi_n
\end{bmatrix},
\qquad i, j, k, l, m, n = [1, 2, 3]
\tag{10}
$$

where $\mathbf{x}^*(t_0)$ is the nominal entry interface, reported in Table 1. In total, 3^6 trajectories have been computed. Each of the computed 3^6 trajectories has been formulated according to the OCP defined in Eqs. (1)-(4), together with Eq. (10).

The states and the controls evolution for the entire database are depicted in Figs. 4 and 5, while the envelope of the trajectories is shown in Fig. 6. Finally, the constraints are illustrated in Fig. 7.

In Fig. 6 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. 4 and 5 one can see that all the states and the controls are smooth, as expected. Specifically, the vehicle follows an oscillating entry, as it is visible from the altitude and the flight-path angles in Fig. 4, 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 counteract it. Therefore, during the first 70 s the vehicle decreases its altitude to less than 40 km. The velocity slightly increases during the 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 angle tends to become steeper. The quasi-skip entry allows to achieve the nominal range of about 1400 km.

Neither the control (Fig. 5), nor the constraints (Fig. 7) exceed the limits. Moreover, it is possible to observe that the terminal area can be reached with a good accuracy in terms of final latitude and longitude (Fig. 6). With the trajectory-database computed, it is now possible to introduce the algorithms for the onboard generation of the trajectory. Indeed, among these 3^6 trajectories, 2^6 trajectories, which define the hexeract enclosing the off-nominal initial conditions, will be used for the trajectory computation. How this is done will be clarified in the next subsection.

C. Selection of reference subspace

The d-dimensional space, $\mathcal{P}_c$ previously defined represents all the possible parameters that can differ from their nominal values, and can be estimated (i.e., with the navigation subsystem). In the basic MPI approach,[22] for each parameter an upper and a lower value are considered, and a corresponding optimal
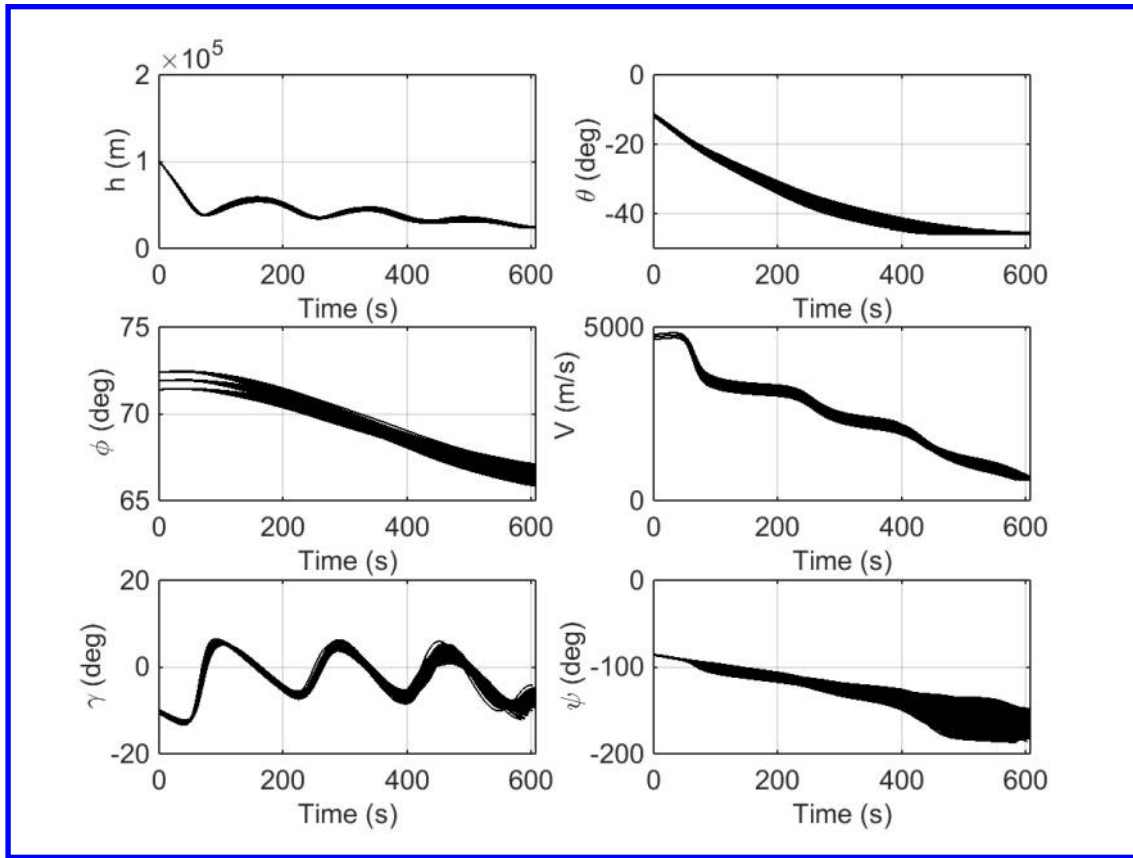
Figure 4: Trajectory database - states.

trajectory is generated. In that case, the database is made of 2^6 extremal trajectories, which are taken and combined to provide an approximated solution corresponding to the inflight initial conditions via MPI.

In the AMPI, these 2^6 trajectories are selected from a larger database. For each of the initial six conditions we are considering three boundary values instead of two values. 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. The selected trajectories will be closer to the initial conditions, leading to a better accuracy of the interpolated trajectory. We can observe how this approach works by reducing it to a 2-D example.

Suppose we have the function (in the most generic case this is a vector function) of two variables $\mathbf{F}(X_1, X_2)$, which is known in the points (X_1^L, X_2^L), (X_1^U, X_2^L), (X_1^L, X_2^U), (X_1^U, X_2^U).

We are interested to find an approximation of $\mathbf{F}(X_1^i, X_2^i)$ for every couple X_1^i, X_2^i such that $X_1^i \in [X_1^L, X_1^U]$, $X_2^i \in [X_2^L, X_2^U]$. The four extremal pairs defined above represent our trajectory database. The domain appears as plotted in Fig. 8(a).

It is possible to build an approximation of $\mathbf{F}(X_1^i, X_2^i)$ with a multivariate interpolation process (in this case bivariate). This multivariate interpolation is described in Algorithm 2. In this context let us consider it as a black box operator called **MPI**. It is possible to build an approximation of $\mathbf{F}(X_1^i, X_2^i)$ with a multivariate interpolation process (in this case bivariate). This multivariate interpolation is described in Algorithm 2. In this context let us consider it as a black box operator called **MPI**.

$$\mathbf{F}(X_1^i, X_2^i) \cong \mathbf{MPI}\left[\mathbf{F}(X_1^L, X_2^L), \mathbf{F}(X_1^U, X_2^L), \mathbf{F}(X_1^L, X_2^U), \mathbf{F}(X_1^U, X_2^U)\right] \qquad (11)$$

The solution is then a function of the information stored in these four points, as shown in Fig. 8(a).

The closer the boundary points are, the more accurate the approximation $\mathbf{F}(X_1^i, X_2^i)$ is. For large dispersions this approach may not be accurate enough. In the AMPI the parameter space is organized into a finer grid; then, for each case, the subspace enclosing the point we are interested in is detected, as
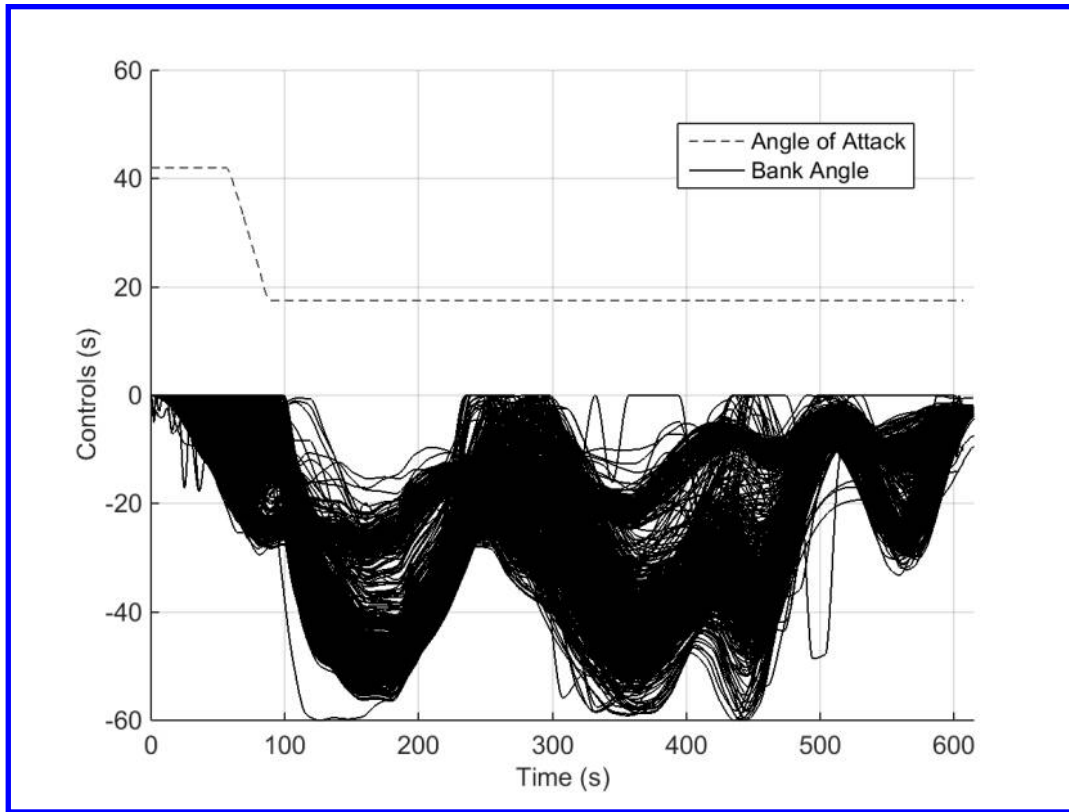
American Institute of Aeronautics and Astronautics

Figure 5: Trajectory database - controls.

shown in Fig. 8(b). The approximation will then be built as

$$\mathbf{F}(X_1^i, X_2^i) \cong \mathbf{MPI} \left[\mathbf{F}(X_1^j, X_2^k), \mathbf{F}(X_1^{j+1}, X_2^k), \mathbf{F}(X_1^j, X_2^{k+1}), \mathbf{F}(X_1^{j+1}, X_2^{k+1}) \right] \tag{12}$$

The objective of this phase is therefore the detection of the indices representing the subspace to use for the computation of the onboard trajectory. The selection of the reference subspace can be done with Algorithm 1, which can be applied to the parameter space p^i, and to the initial conditions x_i, representing the six-dimensional off-nominal initial conditions. The result will be the six couples of the indices, which detect the subspace to be used for the computation of the adaptive trajectory.

With the application of Algorithm 1, the subspace of the database can be easily determined, and it is possible to compute the guidance solution via MPI. The drawback of the classic interpolation schemes in d dimensions is the rapid increase of memory requirements, since the database suffers the so-called *curse of dimensionality*. This drawback is balanced by the use of the MPI scheme, which allows to strongly reduce the amount of data needed without losing accuracy. How this is done will be explained in the next subsections. In the parametric six-dimensional space, each subspace selected for the synthesis of the guidance solution is made of 2^6 neighboring trajectories, combined to provide an approximation of the functions $\mathbf{F}$, which in our case include the states, as well as the bank angle and the final time. This interpolated solution will be the low-density representation of the trajectory and the controls, which can be adapted to any initial condition compatible with the extension of the database. 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

As stated in the previous subsection, a solution of the OCP previously defined depends not only on the specific choice of a parameter vector $\mathbf{p} \in \mathcal{P}_c$, but also on time. Within this work the computation of a solution of an
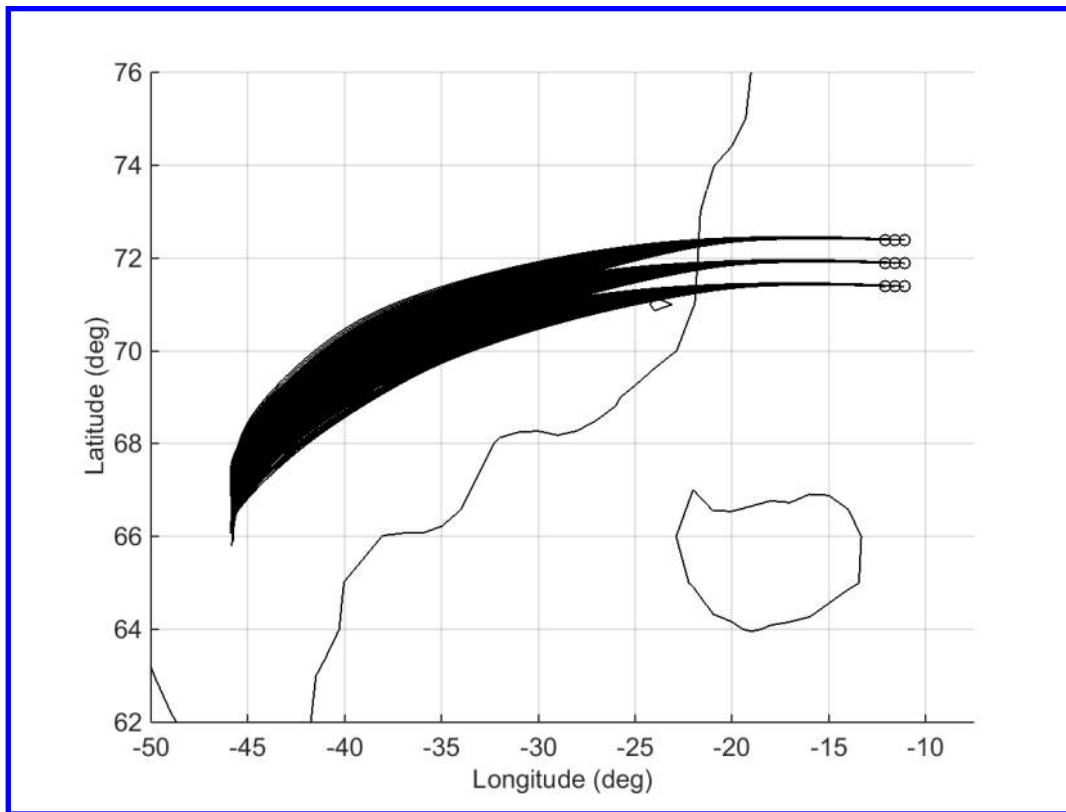
Figure 6: Trajectory database - groundtracks.

Table 3: Selection of reference subspace.

Data: Given: parameter-space elements $\mathbf{p}^i$, and initial conditions x_i, $i = 1,\ldots,6$
for i = 1 : d
    $\mathbf{eP}^i = \text{sign}(\mathbf{p^i} - \mathbf{U}x_i)$
    for j = 1 : $n_i - 1$
        $\mathbf{\Delta p}^i_j = \mathbf{eP}^i_j$-$\mathbf{eP}^i_{j-1}$, $j = 2\ldots,n_i$
        if $\sum_{j=1}^{n_i} \mathbf{eP}^i_j$==-$n_i$
            $[\tilde{h}^i, \tilde{h}^i + 1] = [n_i - 1, n_i]$;
        elseif $\sum_{j=1}^{n_i} \mathbf{eP}^i_j$==$n_i$
            $[\tilde{h}^i, \tilde{h}^i + 1] = [1,2]$;
        else
            $idx = \mathbf{find}(\mathbf{\Delta p}^i_j \neq 0)$
            $[\tilde{h}^i, \tilde{h}^i + 1] = [idx, idx + 1]$;
        end
    end
Result: 2^d indeces $[\tilde{h}^i, \tilde{h}^i + 1]$, $i = 1\ldots,d$

OCP is based on the solution of a corresponding NLP. The interpolation approach described in this section shows how to compute a given point of an interpolated trajectory by using the corresponding values of the trajectory-database. Since we are using a transcription based on the flipped Radau pseudospectral (FRP)

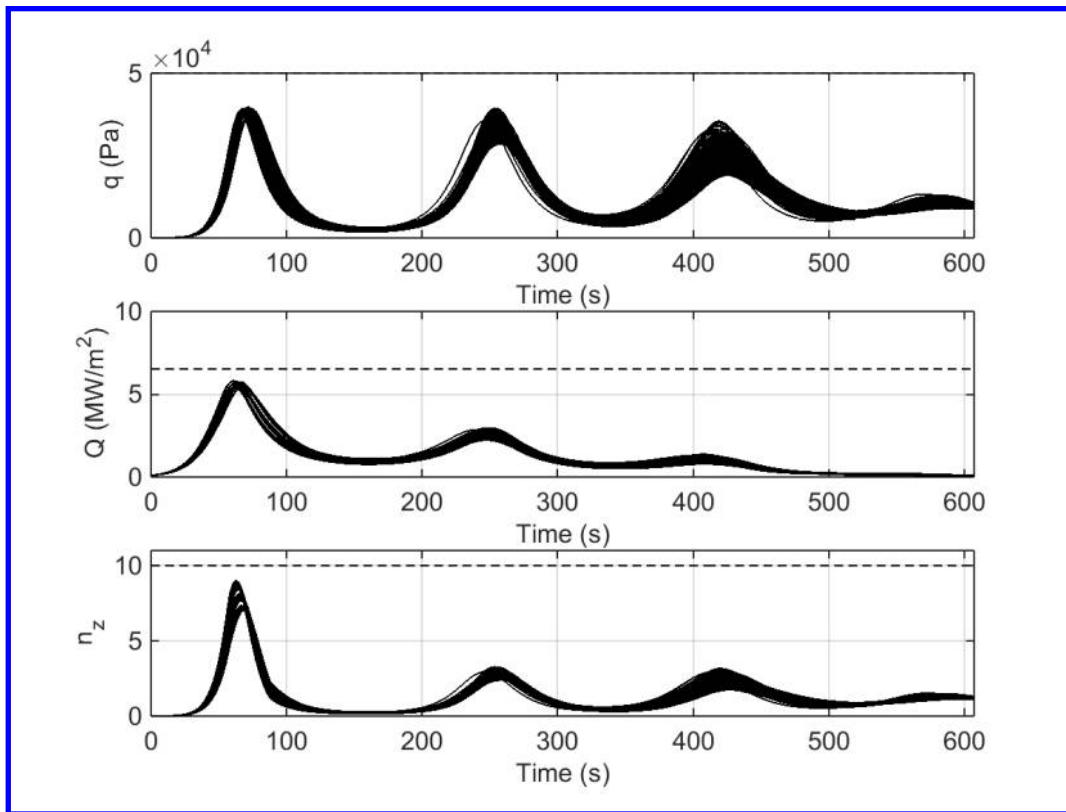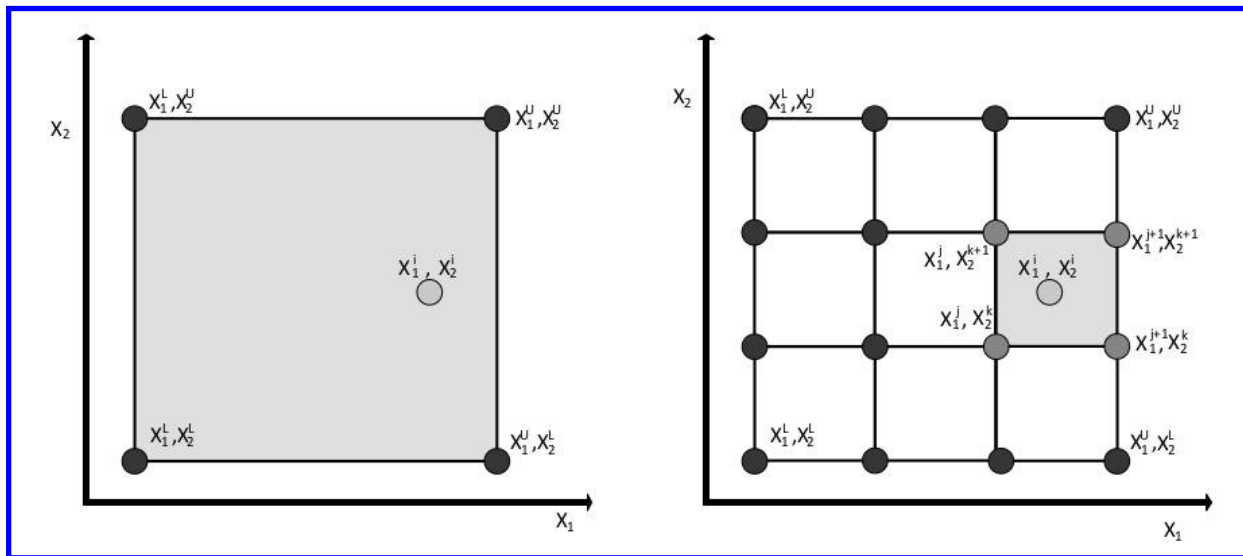American Institute of Aeronautics and Astronautics

Figure 7: Trajectory database - constraints.

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

(b) AMPI approach: given the larger database, four neighbor trajectories around the off-nominal conditions are selected, and used to compute the trajectory.

Figure 8: Examples of domains for the application of the MPI and AMPI techniques.

method,[22, 27] we propose to perform this interpolation at each of the collocation nodes defined by the roots

of the flipped Radau polynomials. Let us consider the definition of Legendre polynomials of order N:[29]

$$L_N(\tau) = \frac{1}{2^N N!} \frac{d^N}{d\tau^N} \left[(\tau^2 - 1)^N \right] \tag{13}$$

The variable τ is defined over the domain $[-1, 1]$. In the FRP method, the collocation points are selected as the roots of the polynomial R_N, computed as a linear combination of the Legendre polynomials of order N and $N-1$ with coefficients 1 and -1, respectively.

$$R_N(\tau) = L_N(\tau) - L_{N-1}(\tau) \tag{14}$$

The roots of the polynomial $R_N(\tau)$ give us the FRP nodes where the MPI is applied. This set of collocation nodes can be seen as the domain of the low-density discrete solution. In the most general case, we have 2^d trajectories for a $d-$dimensional parameter $\mathbf{p}$. This means that at each timestep, defined by the FRP roots, 2^d values for each of the states and controls will be used for the multivariate interpolation process.

Let us consider a generic variable f (i.e., a state or a control) of the optimal-trajectories set, associated with the 2^d values of $\mathbf{p}$ determined by the reference subspace algorithm. To make the notation simple, let us suppose that the reference subspace identifies the indices 1 and 2 for each of the d components of $\mathbf{p}$. For a d-dimensional off-nominal value of x_i, $i = 1, \cdots, d$, we have

$$x_i \in [p_1^i, \ p_2^i], \ i = 1, \cdots, d \tag{15}$$

At each timestep τ_k, and for each of the variables included in the trajectories we have 2^d values of $f(\tau_k, \mathbf{p}_i)$, belonging to the database, which need to be interpolated to provide an intermediate value $f_{int}(\tau_k, \mathbf{x}_i)$ consistent with the d off-nominal initial conditions $\mathbf{x}_i$. We can indicate the 2^d values of $f(\tau_k, \mathbf{p}_i)$ as $\mathbf{F}(\tau_k, \mathbf{p}_i)$. The grid points $\mathbf{p}_i$ are called supporting points while the $\mathbf{F}(\tau_k, \mathbf{p}_i)$ are called supporting values. To perform the multivariate interpolation we will use a simplified version of the tensor product spline.[19] A tensor product spline $s \in \mathbb{S}_{k_1, \mathbf{t}_1} \bigotimes \cdots \bigotimes S_{k_d, \mathbf{t}_d}$ on a grid $\mathbf{P}$ at a grid point $\mathbf{x} \in \mathbf{P}$ is in general defined by

$$s(\mathbf{x}) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_d=1}^{m_d} c_{i_1, \ldots, i_d} B_{i_1, k_1}(x_1) \ldots B_{i_d, k_d}(x_d). \tag{16}$$

In Eq. (16) $B_{i,k}$ denotes the ith B-spline of order k for a given non-decreasing knot vector $\mathbf{t} = (t_i)_{i=1}^{m+k}$. The coefficients

$$\mathbf{C} = (c_{i_1, \ldots, ß_d})_{i_1=1, \ldots, i_d=1}^{m_1, \ldots, m_d} \tag{17}$$

are computed, such that the resulting tensor product spline fulfills the interpolation condition, that is

$$s(\mathbf{p_i}) = f(\tau_k, \mathbf{p}_i) \, \forall \, i \in 1 \cdots d, \ k = 1, \cdots, N_{LD} \tag{18}$$

In our case, the interpolated variable $f_{int}(\tau_k, x_i)$ will be

$$f_{int}(\tau_k, \mathbf{x}_i) = s(\mathbf{x}_i) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_d=1}^{m_d} c_{i_1, \ldots, i_d} B_{i_1, k_1}(x_1) \ldots B_{i_d, k_d}(x_d). \tag{19}$$

Since each B-spline in Eq. (16) depends only on a single variable, the d-variate interpolation problem can be divided into d univariate problems. Each univariate spline interpolation is solved via the numerical stable and efficient algorithm of De Boor.[30]

Depending on the choice of k and a suitable knot vector, $\mathbf{t}$, the tensor product spline interpolation in Eq. (16) corresponds to an interpolation method such as piecewise-constant, piecewise-linear or piecewise-cubic Hermite interpolation. In general, the tensor-product spline interpolation allows the use of different interpolation methods for each of the d dimensions. It is now necessary to determine the knot vectors $\mathbf{t}_i$ and the coefficient matrices $\mathbf{C}$. Within this work the grid $\mathcal{P}$ consists of two supporting points for each direction, such that $n_i = 2, i = 1, \ldots, d$ and

$$\mathcal{P} = \left\{ p_1^1, p_2^1 \right\} \times \cdots \times \left\{ p_1^d, p_2^d \right\}. \tag{20}$$

American Institute of Aeronautics and Astronautics

The choice of the order of the splines $k_i = 2, i = 1, \ldots, d$ and the corresponding choice of the knot vectors

$$\mathbf{t}_i = \left(t_j^i\right)_{j=1}^4 = \left(p_1^i, p_1^i, p_2^i, p_2^i\right), \qquad i = 1, \ldots, d \tag{21}$$

defines a piecewise-linear interpolation in each direction on the given d-dimensional grid. For $\mathbf{C} = \mathbf{F}(\tau_k, \mathbf{p}_i)$ the piecewise linear interpolation method fulfills the interpolation condition given by Eq. (18) and there is no additional computational effort required to determine the coefficients $\mathbf{C}$. With these simplifications, the evaluation of the tensor-product spline interpolation at a given point of the parameter space $\mathbf{p} \in \mathcal{P}_c$ is described by the pseudo code according to Algorithm 2. The EvalUnivSpline($\mathbf{t}, \mathbf{c}, x_i$) function in Algorithm 2

Table 4: 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
    $\mathbf{A}_i = \text{EvalUnivSpline}(\mathbf{t}_i, \mathbf{A}_{i-1}, x_i)$;
    $\mathbf{A}_i = \mathbf{A}_i'$;
end
$s(\mathbf{p}) = \mathbf{A}_d$;
Result: interpolated values $s(\mathbf{x}_i) = f(\tau_k, \mathbf{x}_i)$

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 to generate onboard trajectories for the states and the control inputs of the assumed system for each element of the parameter space $\mathbf{p} \in \mathcal{P}_c$. It is important to emphasize that interpolated trajectories in general are not formally solutions of the equations of motion that are used within 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. With reference to Eqs. (13) and (14), for a given order, it is possible to extract the roots of the associated flipped Radau polynomial. This set of collocation nodes can then be used to approximate polynomial approximations of the original continuous functions. Given a function $F(\tau)$ sampled in $N + 1$ points, one can build this approximating function as

$$F(\tau) \cong \sum_{i=0}^N F_i P_i(\tau) \tag{22}$$

with $P_i(\tau)$ defined as

$$P_i(\tau) = \prod_{\substack{k=0 \\ k \neq i}}^N \frac{\tau - \tau_k}{\tau_i - \tau_k} \tag{23}$$

An example of Radau collocation points is reported in Fig. 9(b), where 11 nodes are considered, together with the uniform distribution having the same number of nodes. Once again, the considered interval is [-1, 1], which represents the natural domain of definition for the Legendre Polynomials. Note that the point $\tau = -1$ is still part of the discretization, even if it is not a collocation point.[31] At first glance, the polynomials can be built using every criterion for the choice of the nodes, for instance a uniform distribution. So why choosing the roots of the Legendre polynomials instead? The answer resides in the following property: the distribution of points derived from the roots of this particular family of functions is not affected by the *Runge*

American Institute of Aeronautics and Astronautics

phenomenon. *Runge phenomenon* is a problem of oscillation at the edges of an interval that occurs when interpolating with polynomials of high degree over a set of equispaced interpolation points. This oscillatory behavior does not permit a good approximation of the interpolant in the extremes of the interval considered. The phenomenon can be well understood if one looks at the example function

$$F(\tau) = \frac{1}{1 + 25\tau^2} \tag{24}$$

and samples the function with 11 and 25 uniformly distributed nodes. In this example, $F(\tau)$, defined in the interval $[-1, 1]$, represents the function we want to rebuild via interpolation using a discrete set of values $F(\tau_i)$, $i = 0, 1.., 10$ and $i = 0, 1.., 24$. As anticipated, with the chosen distribution of points, the interpolating polynomial will show a poor approximation, in the extremes of the interval considered, as we can see in Fig. 9 (a). The intuition would suggest then to simply increase the number of nodes; as a consequence, the
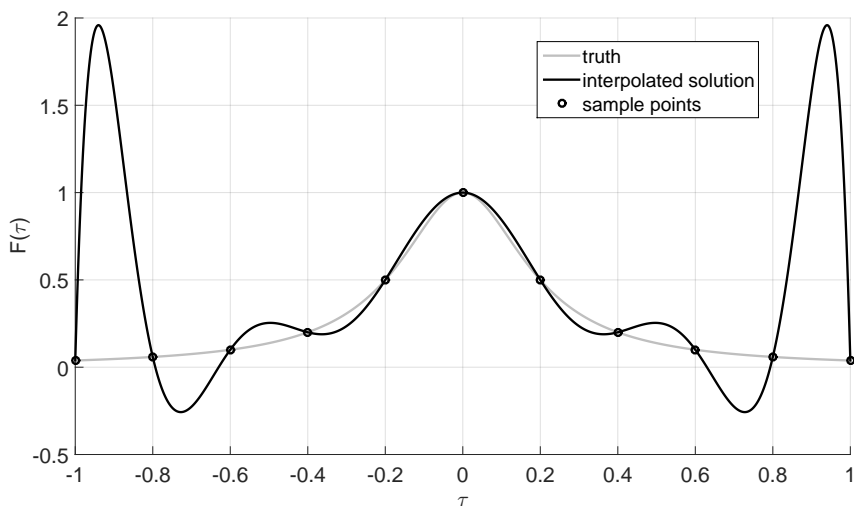
(a) Interpolation scheme based on uniform distribution (11 nodes).
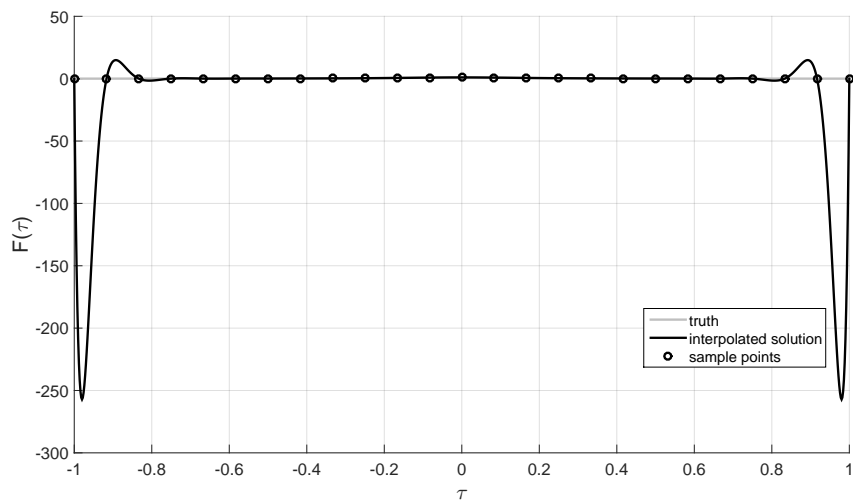
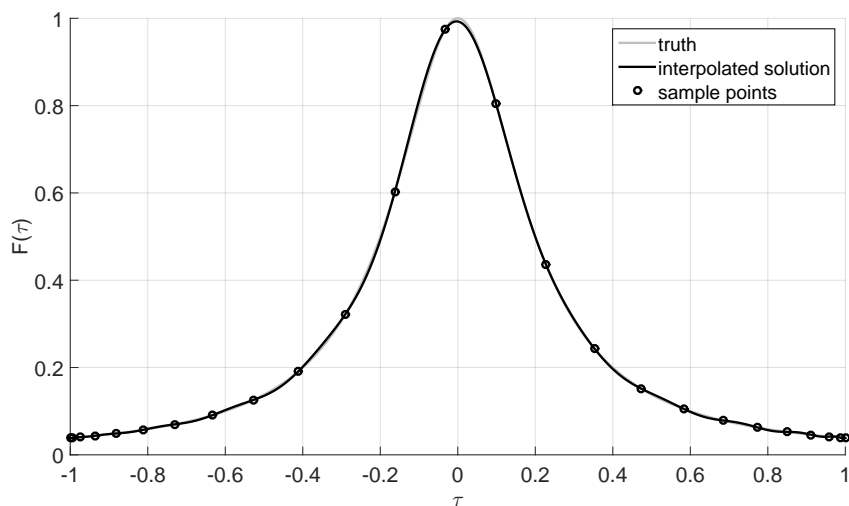(b) Interpolation scheme based on uniform distribution (25 nodes).

Figure 9: Runge Phenomenon: interpolation of the function $F(\tau) = \frac{1}{1+25\tau^2}$ using (a) 11 and (b) 25 uniformly distributed nodes.

amplitude of the Runge oscillations even becomes larger, and thus the approximation worse, as depicted in

American Institute of Aeronautics and Astronautics

Fig. 9 (b). Figures 10 (a) and 10 (b) show how this phenomenon is completely avoided using the roots of the Legendre polynomials instead. The reason for this behavior stems from the fact that the chosen linear

(a) Interpolation scheme based on Legendre-Radau polynomials roots (11 nodes).

(b) Interpolation scheme based on Legendre-Radau polynomials roots (25 nodes).

Figure 10: Runge Phenomenon: interpolation of the function $F(\tau) = \frac{1}{1+25\tau^2}$ using (a) 11 and (b) 25 nodes derived from Flipped Radau-Legendre Polynomials.

combinations of Legendre polynomials are orthogonal in the domain $\tau \in (-1, 1]$. We will refer to this domain as *pseudospectral time*. Since our problems are defined over a domain represented by the physical time, a mapping between the pseudospectral time $\tau \in [-1, 1]$ and physical time $t \in [t_0, t_f]$ is needed. It is easy to demonstrate that this mapping can be built using two affine functions:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \tag{25}$$

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0} \tag{26}$$

The following properties justify the choice of using pseudospectral methods for the characterization of the discrete domain.

American Institute of Aeronautics and Astronautics

- Spectral convergence in the case of a smooth problem.

- Straightforward implementation.

- Sparse structure of the associated NLP problem.

- Mapping between the costates of the NLP discrete solution and the costates of the optimal continuous solution in virtue of the Pseudospectral Covector Mapping Theorem.[32]

- Removal of the Runge phenomenon.[33, 34]

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 storing 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.

How can this be used for the conversion of the HD discrete solutions? 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 $\mathbf{t_{LD}}$, the states $\mathbf{X_{LD}}$, and by the controls $\mathbf{U}_{LD}$. The matrices $\mathbf{X}_{LD}$ and $\mathbf{U}_{LD}$ have dimensions $n_s \times (N_{LD} + 1)$ and $n_c \times (N_{LD} + 1)$, respectively, where n_s and n_c are the number of states and controls associated with the problem under analysis. We can group the states and the controls in a matrix $\mathbf{T}_{LD}$, having dimensions $(n_s + n_c) \times (N_{LD} + 1)$.

$$\mathbf{T}_{LD} = \left\{ \begin{array}{c} \mathbf{X}_{LD} \\ \mathbf{U}_{LD} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_{N_{LD}} \\ \mathbf{U}_0, \mathbf{U}_1, ..., \mathbf{U}_{N_{LD}} \end{array} \right\} \tag{27}$$

Our objective is to efficiently convert the matrix $\mathbf{T}_{LD}$ into a matrix $\tilde{\mathbf{T}}_{HD}$ representing the HD discrete solution,

$$\tilde{\mathbf{T}}_{HD} = \left\{ \begin{array}{c} \tilde{\mathbf{X}}_{HD} \\ \tilde{\mathbf{U}}_{HD} \end{array} \right\} = \left\{ \begin{array}{c} \tilde{\mathbf{X}}_0, \tilde{\mathbf{X}}_1, ..., \tilde{\mathbf{X}}_{N_{HD}} \\ \tilde{\mathbf{U}}_0, \tilde{\mathbf{U}}_1, ..., \tilde{\mathbf{U}}_{N_{HD}} \end{array} \right\} \tag{28}$$

where $N_{HD} + 1$ is the number of points representing the HD discrete solution. Moreover, the HD time vector $\tilde{\mathbf{t}}_{HD}$ must be computed. If we apply Eqs. (22) and (23) to $F(\tau)$, we can write

$$F(\tau) = \sum_{i=0}^{N_{LD}} F_i \prod_{\substack{k=0 \\ k \neq i}}^{N_{LD}} \frac{\tau - \tau_k}{\tau_i - \tau_k}, \ \tau \in [-1, 1] \tag{29}$$

where F_i represents a low-density variable. It can be replaced with the pth row of $\mathbf{T}_{LD}$ 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 high-density representation of our variables

$$\tilde{T}_{HD}^p(\tilde{\tau}_m) = \sum_{i=0}^{N} T_{LD,i}^p \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{\tilde{\tau}_m - \tau_k}{\tau_i - \tau_k}, \ p = 1, ...(n_s + n_c), \ m = 0, ...(N_{HD}) \tag{30}$$

Equation (30) can be extended to all the rows of the matrix $\tilde{\mathbf{T}}_{HD}$, and rewritten in matrix form as

$$\tilde{\mathbf{T}}_{HD} = \mathbf{T}_{LD}\mathbf{P}_{FRP} \tag{31}$$

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

$$\mathbf{P}_{FRP} = \begin{bmatrix} \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_0 - \tau_k}{\tau_0 - \tau_k} & ... & \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_{N_{HD}} - \tau_k}{\tau_0 - \tau_k} \\ ... & ... & ... \\ \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_0 - \tau_k}{\tau_{N_{LD}} - \tau_k} & ... & \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_{N_{HD}} - \tau_k}{\tau_{N_{LD}} - \tau_k} \end{bmatrix} \tag{32}$$

American Institute of Aeronautics and Astronautics

The elements $\tilde{\tau}_m$ represent the high-density discrete pseudotime vector, defined between -1 and 1. Since both the nodes τ_k, where the solutions are computed, and the nodes $\tilde{\tau}_m$, where the solutions are effectively evaluated, 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 linear interpolation process and to the matrix multiplication defined in Eq. (31). 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 by using the following expression:

$$\tilde{t}_m = \frac{\tilde{t}_f - \tilde{t}_0}{2}\tilde{\tau}_m + \frac{\tilde{t}_f + \tilde{t}_0}{2}, \quad m = 0, ..., N_{HD} \tag{33}$$

The initial time $\tilde{t}_0$ is given by the initial time t_0. The final time $\tilde{t}_f$ 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 feedforward guidance solution is completely generated with the application of the Algorithms 1 and 2, and Eqs. (31) and (33).

V.   Numerical Results

A.   Simulation Campaign

For the validation of the method, a Monte Carlo campaign of 1000 cases has been simulated. For each case, the following random dispersions (3σ) have been generated.

$$
\begin{aligned}
\delta h &\in [-250, +250] \text{ m} \\
\delta\theta &\in [-0.5, +0.5] \text{ deg} \\
\delta\phi &\in [-0.5, +0.5] \text{ deg} \\
\delta V &\in [-70, +70] \text{ m/s} \\
\delta\gamma &\in [-0.5, +0.5] \text{ deg} \\
\delta\psi &\in [-0.5, +0.5] \text{ deg}
\end{aligned}
\tag{34}
$$

The type of dispersion assumed implies that in some cases the boundaries defined here are exceeded. In those cases, no extrapolation is performed, but the boundary-trajectories in the database are used to generate the trajectories, and the initial error is not nulled, but reduced to the difference between the initial conditions, and the closest initial conditions that the database contain. For instance, if an initial error of +80 m/s in terms of velocity is experienced, and the database is limited to +70 m/s, the generated trajectory will show an error of only 10 m/s instead of 80 m/s. In all the other cases the initial error that the generated trajectories will have is 0 m/s, and this applied to all of the six states. This choice prevents the use of the algorithm in regions where the behavior of the system has not been directly observed (i.e., during the trajectory-database generation). The solutions have been generated according to the initial conditions by using the proposed AMPI approach. A tracking controller[23] has been added to the guidance scheme. Results are compared with the classical approach based on the tracking of the nominal reference solution. The states, the state errors, the controls, the groundtracks, and the constraints are then compared and reported in Figs. 11-19, together with some histograms, which show the difference in terms of final altitude, velocity and range-to-go.

The multivariate approach generates meaningful trajectories, as can be seen in Fig. 11, and the same can be observed in terms of the constraints: the results are comparable to the classic tracking approach (Fig. 19), and the maximum values are consistent with the ones expected from the nominal scenario. Figure 12 shows the error of the singular states. The generated trajectories significantly reduce the error that has to be handled by the feedback controller, leading to a significant decrease of the dispersions. This is well visible from Figs. 14 and 15, where the trajectories and the final footprints (represented in crossrange and range-to-go) are plotted. The figures show a large improvement of performance when the AMPI is used, w.r.t. the nominal tracking. 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 nominal solution. Another consequence is that the online adaptation of the trajectory simplifies the work of the feedback controller (Figs. 13(a) and 13(b)). Indeed, the proposed technique allows to have smooth AoA and bank angle profiles, and in general a reduced control activity w.r.t. the nominal tracking, as the feedback control needed for
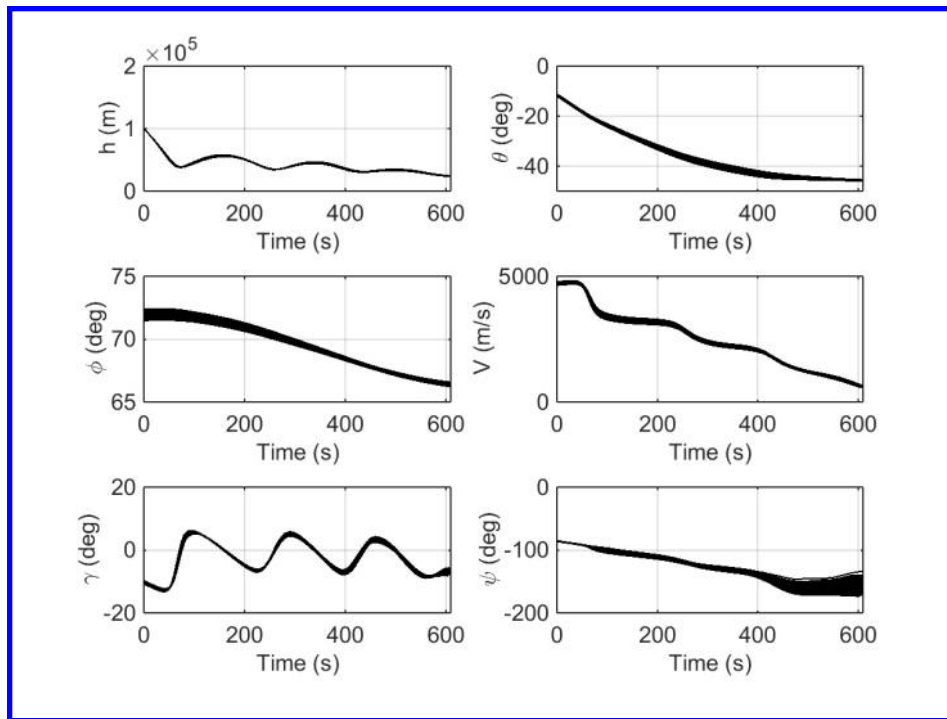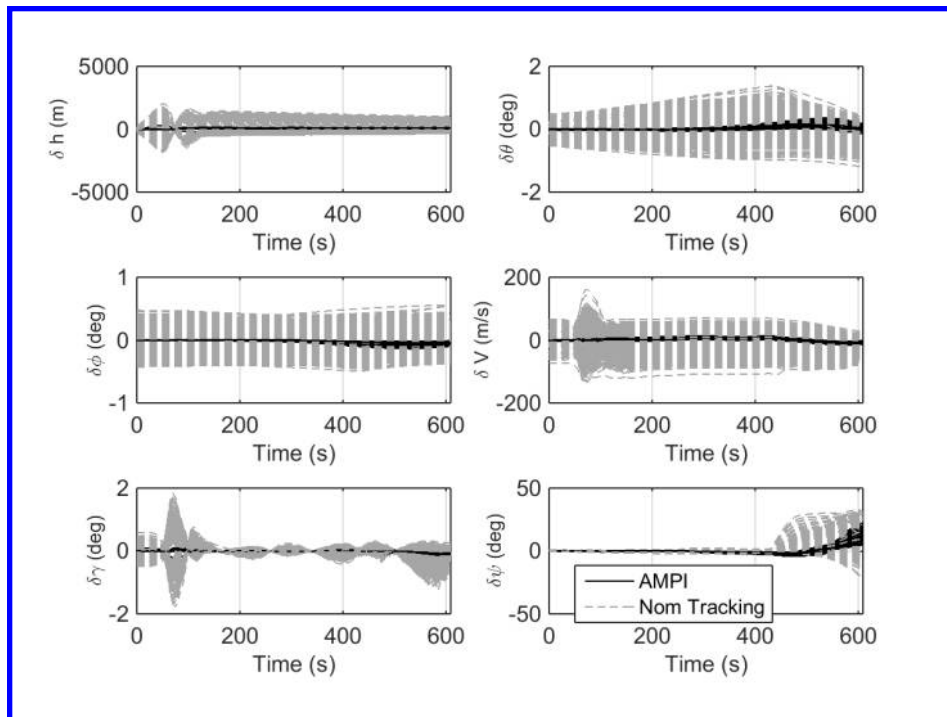
American Institute of Aeronautics and Astronautics

Figure 11: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: states.

Figure 12: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: state errors.

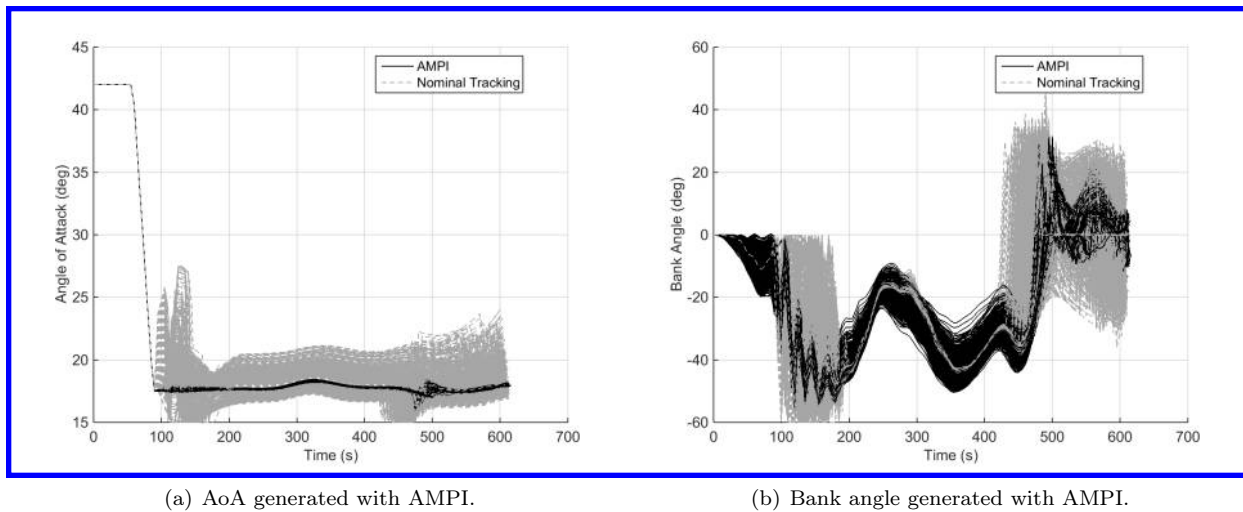(a) AoA generated with AMPI.   (b) Bank angle generated with AMPI.

Figure 13: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: angle of attack and bank angle.
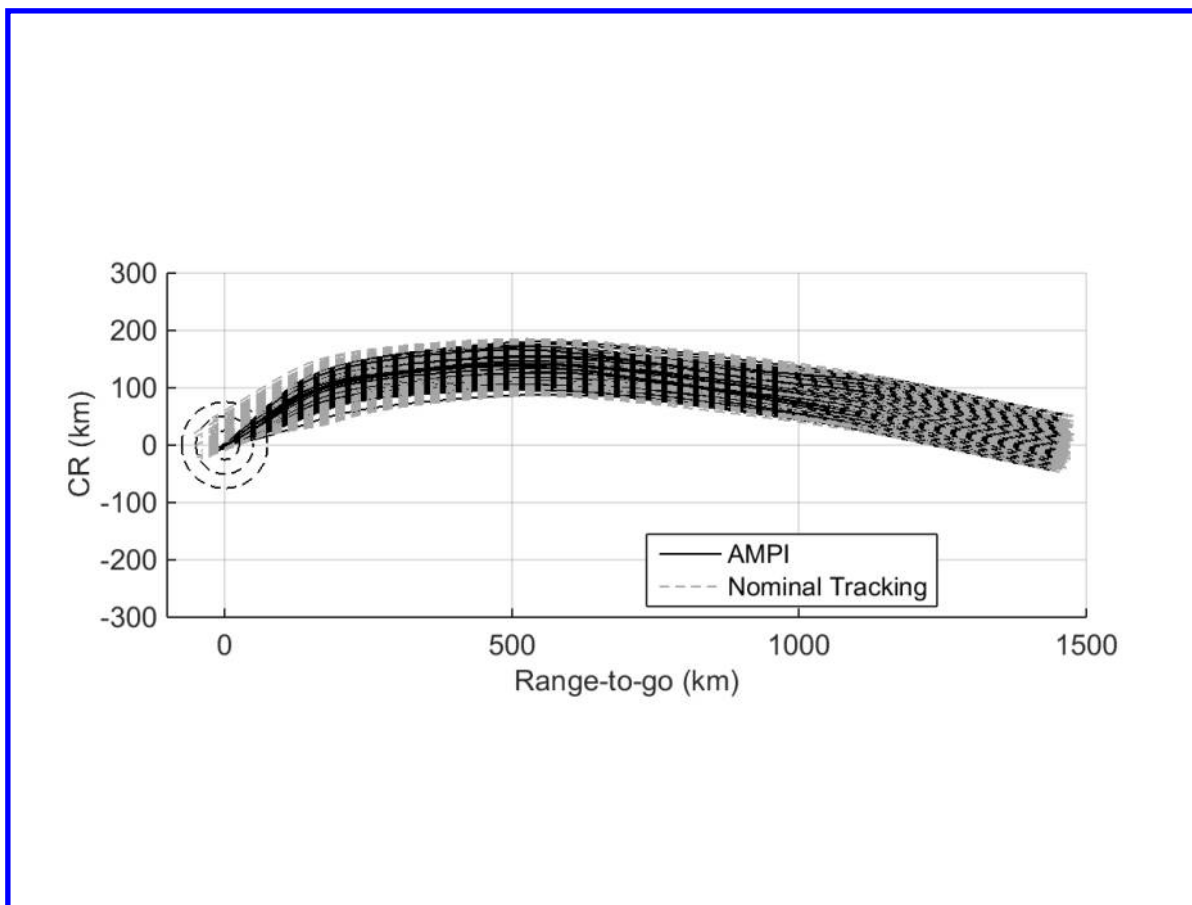

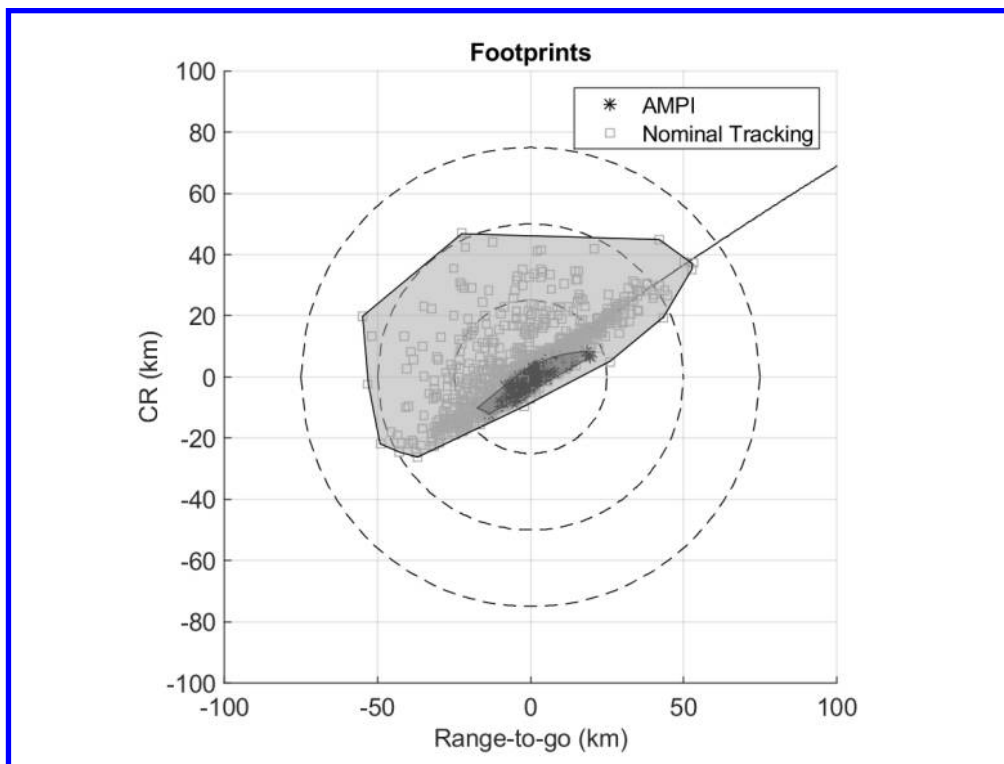Figure 14: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: trajectories.

Figure 15: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: footprints.

the compensation of the errors is significantly reduced. In terms of final altitude, velocity, and range-to-go, results are reported in Figs. 16-18. In terms of altitude a reduced dispersion associated with the use of the AMPI is visible from Fig. 16. In the worst cases, there is a difference of 200 m w.r.t. the reference value when the AMPI is used, while this value goes up to 1.5 km for the nominal tracking. In terms of final velocities, both the systems achieve good performance, and also in this case the AMPI is slightly better (a maximum error of 2 m/s versus 3 m/s). Completely different are the results in terms of range-to-go, as shown in the histogram of Fig. 18. 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 163 cases corresponding to the nominal tracking. The cases associated with the AMPI method go up to 985 cases against 400 if a radius of 13 km is considered. No cases fall outside a radius of 25 km when the proposed technique is applied, against 254 cases obtained outside these boundaries in case the traditional tracking approach is used. Table 5 lists the results of the Monte Carlo campaign.

Table 5: Dispersion Analysis - 1000 MC runs.

Ellipse / Controller	$[25 \times 25]$	$[50 \times 50]$	$[75 \times 75]$	Outside
AMPI	1000	0	0	0
Nominal Tracking	746	242	12	0

It is possible to see that 100% of the cases fall into the finest circle, i.e., within a radius of 25 km versus 74.2% of the cases associated with 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 $\cong 5,032$ km^2. The two areas are depicted in Fig.15. Finally, for both the methods, the constraints are always satisfied, as one can see in the constraints' profiles depicted in Fig.19.

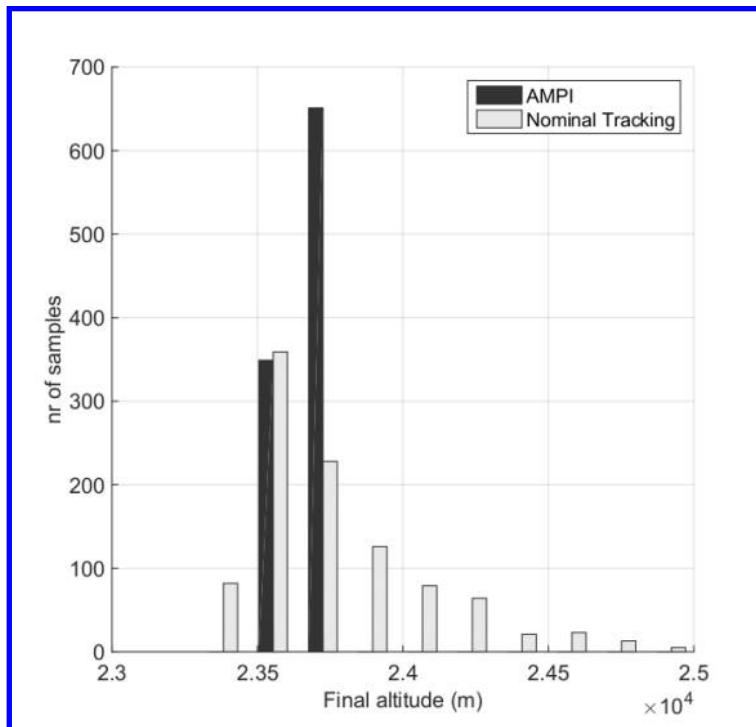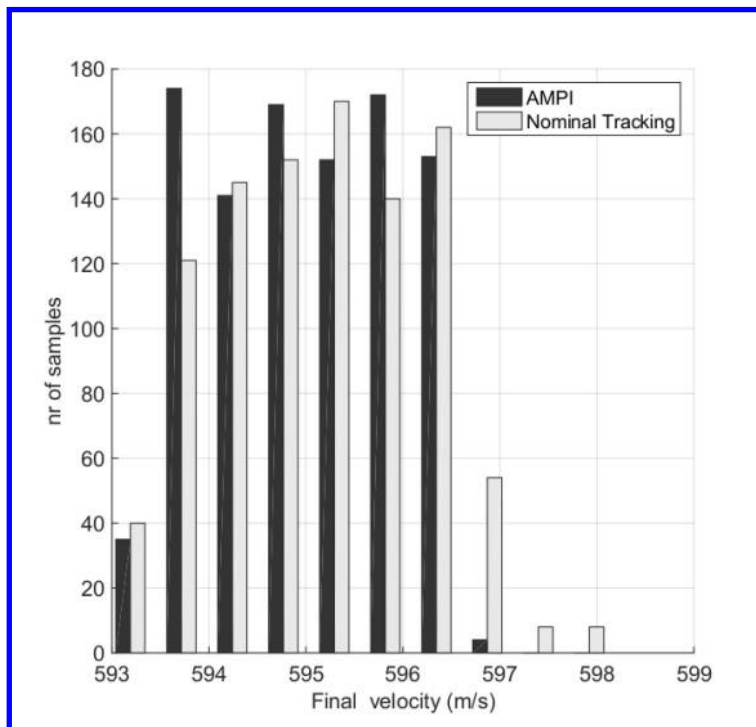American Institute of Aeronautics and Astronautics

Figure 16: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: Final altitudes.

Figure 17: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: Final velocities.
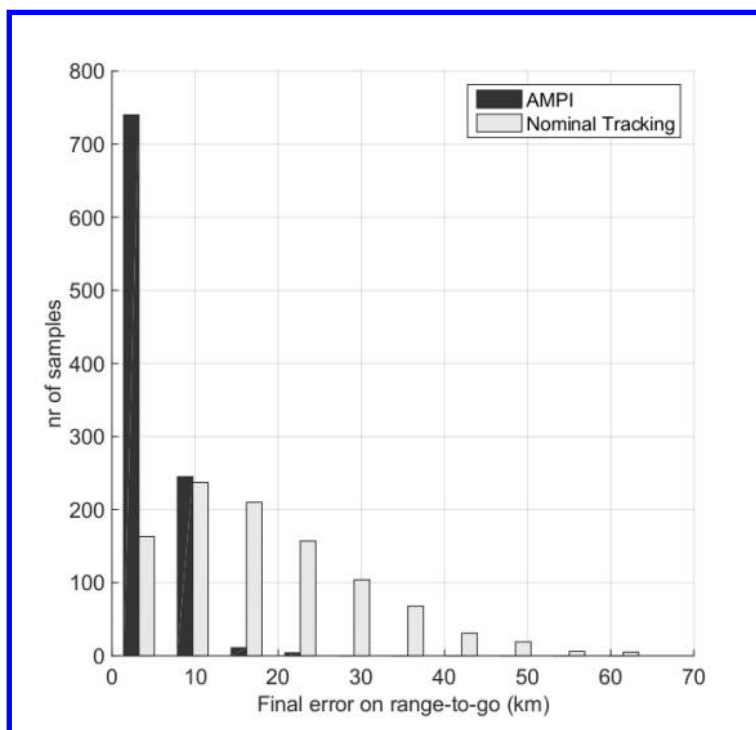
American Institute of Aeronautics and Astronautics

Figure 18: MC campaign (N = 1000): Multivariate interpolated trajectory vs nominal tracked trajectory: range-to-go.
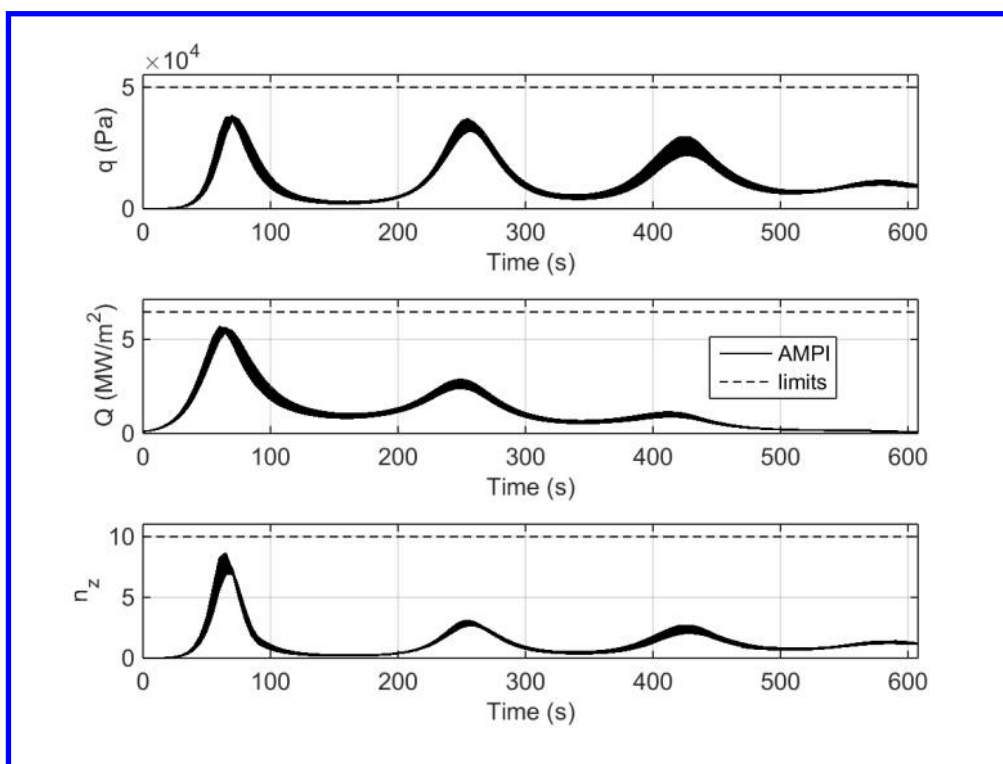

Figure 19: MC campaign (N = 1000): Multivariate interpolated trajectory: constraints.

VI.   Conclusions and Future work

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. Taking advantage from the pseudospectral transcription, the synthesis can be efficiently performed by processing information stored in subspaces of trajectories' database, in this specific case "hexeracts" of trajectories. 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. For instance, for what regards the lateral guidance, the percentage of cases which fall into the finest circle rises from 74.6% to 100%, and the overall dispersion area is reduced to 4.25% of the size obtained with the tracking of the nominal solution. This is significant when we consider that, for the current scenario, the lateral bank-reversal logic cannot guarantee good performance, because of the large initial heading-error. Instead, the AMPI technique allows to drastically improve the lateral performances.

A further advantage coming from the use of this technique is the lossless reduction of the database size. Since the LD-HD conversion is based on the properties of the transcription method, high-density discrete trajectories can be obtained by storing a significantly smaller number of values, as a consequence of the choice of FRP nodes as domain. To give an example, for the case analysed in this work, the database is reduced to $\cong 3\%$ of the size of a database stored in traditional way. This aspect can reduce the requirements for the memory of the onboard CPU. The good performance obtained suggest that the proposed method can be a good choice for scenarios having large dispersions at entry. Moreover, the method is of general validity, and can be also applied to other off-nominal parameters, which can be estimated onboard. Future work includes the analysis of the proposed method in terms of CPU burden, and the impact on the performance of alternative schemes to discretize the parameter space.

References

[1]Bogner I., "Description of Apollo Entry Guidance", *NASA Technical Memorandum CR-110924*, 1966

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

[3]Mease K. D. and Kremer J. P., "Shutte Entry Guidance Revisited Using Nonlinear Geometric Methods", *Journal of Guidance, Control and Dynamics*, Vol.17 No. 6, 1994, doi:10.2514/3.21355

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

[5]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

[6]Mooij E.: "Heat-Flux Tracking for Thermal-Protection System Testing," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA 2014-4141, San Diego, CA, 2014, doi:10.2014/6.2014-4141.

[7]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.

[8]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

[9]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

[10]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

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

[12]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

[13]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

[14]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

[15]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.

[16]Knauer M. and Büskens C., "From WORHP to TransWORHP", *5th International Conference on Astrodynamics Tools and Techniques*, Noordwijk, 2012

[17]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

[18]Lockner E., Oehlschlägel T., Theil S., Knauer M., Tietjen J. and Büskens C., "Real-Time capable trajectory synthesis

American Institute of Aeronautics and Astronautics

via multivariate interpolation methods for a moon landing manoeuvre", *CEAS Space Journal*, DOI 10.1007/s12567-014-0063-z, 2014

[19]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

[20]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

[21]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

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

[23]Sagliano. M., Mooij E. and Theil S., "Sliding Mode Controller for Hypersonic Entry Guidance", *Journal of Guidance, Control and Dynamics*, to be submitted

[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]Vinh, N.X., *Optimal trajectories in atmospheric flight*, Elsevier scientific publishing company, 1981

[29]Abramovitz M. and Stegun I. A., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover, New York, ISBN 978-0-486-61272-0, 1972

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

[31]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

[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

[33]Ross I. M., Sekhavat P., Fleming A. and Gong Q.,"Optimal Feedback Control: Foundations, Examples, and Experimental Results for a New Approach", *Journal of Guidance, Control and Dynamics*, Vol.31 No. 2, pp. 307-321, 2008

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

[35]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

American Institute of Aeronautics and Astronautics