Proceedings of the 3rd CEAS EuroGNC, Specialist Conference on Guidance, Navigation & Control
Co-organized by ONERA, ISAE-SUPAERO & ENAC
Toulouse, France - April 13-15, 2015

Real Time Adaptive Feedforward Guidance for Entry Vehicles

Marco Sagliano, Thimo Oehlschlägel, Stephan Theil, and Erwin Mooij

Abstract One of the most powerful analysis tools to deal with entry guidance problems is the possibility to formulate them as optimal control problems (OCPs). Environmental constraints, actuator limits, and strict requirements on the final conditions can be efficiently transcribed, resulting in a discrete, finite-dimension nonlinear programmming (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 nominal optimal solution can be adapted to the actual flight conditions, which can significantly differ from the nominal scenario. This paper proposes an approach based on multivariate interpolation to generate entry guidance solutions. The real-time capability is ensured in virtue of the lower CPU efforts required to execute the interpolation operation. The approach is here proposed for initial-conditions variations, but can in principle 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 scheduled to be launched in 2016.

1 Introduction

Since the beginning of the Apollo program, entry guidance has been widely treated by engineers and researchers. The first, successful approach, used for several programs (Apollo, Space Transportation System, MSL, [9, 17, 18, 15]), was based on the planning of an entry trajectory in the drag-velocity plane. The rationale for this

Marco Sagliano - Thimo Oehlschlägel - Stephan Theil
Deutsches Zentrum für Luft und Raumfahrt, Robert Hooke Straße 7, 28359, Bremen, Germany, e-mail: Marco.Sagliano@dlr.de

Erwin Mooij
Delft University of Technology, Kluyverweg 1, 2629, HS Delft, The Netherlands e-mail: e.mooij@tudelft.nl

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 the drag-velocity plane. The longitudinal guidance can then be derived either assuming the equilibrium-glide approximation, or extracting the longitudinal states (altitude, speed, flight-path angle) from the drag acceleration and its derivatives. It is possible to demonstrate that similar, but more accurate results can be obtained if the drag-velocity plane is replaced by the drag-energy plane [13, 16].

In either case, approximations, disturbances and modeling errors require the use of a feedback controller 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 Energy Management (TAEM) interface. In parallel to these approaches, the use of techniques based on optimal control [5, 26] has achieved significant improvements. The increased CPU capabilities, together with the development of dedicated algorithms [3], have led to the possibility to transcribe the problem into a discrete, finite-dimension problem (i.e., an NLP problem) which can be efficiently solved with one of the available and well-known NLP solvers [7, 10, 27]. The drawback of this approach is that the computed solution is optimal within the limits of the accuracy of the models, and the deviation of the inflight conditions from 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 mission success.

In this paper an approach based on multivariate pseudospectral interpolation for the generation of feedforward guidance solutions is proposed. Significant steps in this direction have already been performed. Saraf et al. [24] use interpolation schemes applied to extremal drag-energy profiles for generating landing footprints for entry missions. Lockner et. al [11, 12] developed a more extensive approach based on tensor product splines [14], which perform excellent for the lunar landing problem. Arslantas et al. [2] used a similar technique for reachibility-set computations.

In this work, the tensor product spline algorithm is integrated with pseudospectral methods, to exploit the interesting properties that characterize them. The proposed method leads to the possibility to have a near-optimal real-time trajectory synthesis able to deal with multiple, significant off-nominal conditions. The real-time capabilities of the method will be demonstrated by comparing the CPU time required to compute the interpolated and the optimal solutions, generated by solving the relative optimal control problems with the use of pseudospectral methods [20, 23]. Indeed, it is possible to demonstrate that the generation of a full trajectory is reduced to a dot product between the interpolated results and a constant matrix. This matrix is computed during the database generation with no further cost from a computational point of view. This approach gives the possibility to online adapt the entry guidance solution in case significant entry condition variations are experienced. 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 synthesized, by performing a new computation of the database with little or no modifications of the flight software.

The work is organized as follows: In Sec. 2 an overview of the SHEFEX-3 mission is given. The mission requirements will provide the basis to formulate a related optimal-control problem, which is presented in Sec. 3. In Sec. 4 the multivariate interpolation method is briefly reported, and the proposed approach, in combination with the adopted pseudospectral method, is described. A detailed description of the trajectory-database generation is the subject of Sec. 5, while in Sec. 6 the simulation campaign results are reported, and a time comparison between the interpolated trajectories and the optimal trajectories is performed. Section 7, finally, concludes this paper with a discussion of the results obtained.

2 Reference Mission Description - SHEFEX-3

The reference scenario is one of the proposed misson profiles for SHEFEX-3. SHEFEX (SHarp Edge Flying EXperiment) is a DLR-led series of missions for scientific experiments and development of the European technologies for atmospheric reentry. SHEFEX-2 [25] was successfully launched from Norway (Andøya Rocket Range) in June 2012. To go on with the effort to increase the technological level for real space missions, a new challenge arises in the next years with the development of SHEFEX-3. SHEFEX-3, foreseen to be launched in 2016, will be more complex than SHEFEX-2 in terms of the presence of a real guided re-entry phase, while for SHEFEX-2 an autonomous guidance and control phase was only partially foreseen. As a consequence, the mission will be more complex and ambitious, and requires the guidance system to be able to manage significant off-nominal conditions. Also in this case, the current launch site is Andøya, while the terminal area is placed in Greenland. An alternative scenario with the terminal point in the Svalbard Archipelago has already been proposed in [22]. The spacecraft will be launched with a rocket based on the Brazilian engines S50/S44. After the stages separation and the coast phase, the unpowered descent phase follows. An overview 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 the Space Shuttle reentry or the X-33 studies [5, 13, 16]. 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. At that moment a parafoil is deployed for the final descent and landing. The nominal entry and terminal conditions are reported in Table 1. The requirements in terms of dispersions at the terminal-area interface are associated with an ellipse having a semiaxis equal to 0.5 deg in terms of longitude and latitude, with the center placed in the nominal terminal condition. Moreover, a fine and a coarse ellipse are defined to characterize the accuracy of the results. These two ellipses have semiaxis equal to 0.25 deg and 1 deg respectively.

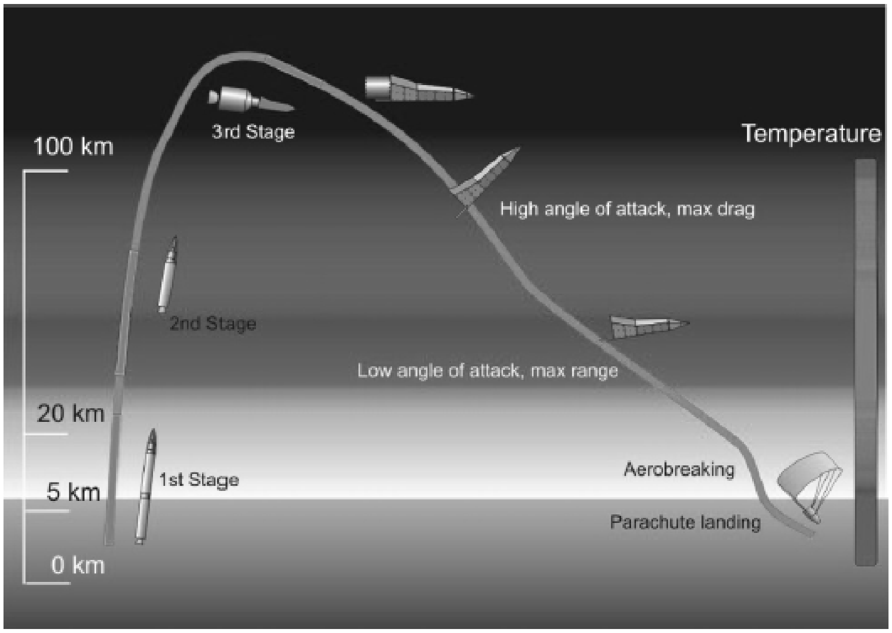


Fig. 1 Shefex-3 Mission Profile.

Table 1 Nominal Entry and Terminal Conditions for SHEFEX-3 guided flight.

State	Initial Value	Terminal Value
Geocentric Altitude h (km)	100.10	20 ± 5
Geocentric Longitude θ (deg)	-11.60	-46.00
Geocentric Latitude ϕ (deg)	71.89	66
Velocity Modulus V (m/s)	4712.26	525 ± 50
Flight-path Angle γ (deg)	-10.31	free
Velocity Azimuth Angle ψ (deg)	-85.92	free

3 Optimal Trajectory Generation

The first step is a proper formulation of the related optimal control problem (OCP). The requirements of the mission include a minimization of the dispersion around the terminal point, at the prescribed altitude and velocity to meet the parafoil conditions. The solution to this OCP problem will provide the reference solution, that is, the reference trajectory and the reference controls, which satisfy all our requirements. Constraints such as dynamic pressure, heat flux and load factor, are also taken into account. The cost function will then be represented as the difference between the current and the desired final states. The gravity model is derived from the WGS84 model as central field with only the J_2 term, as, after a specific analysis, it was observed that the higher-order terms can be neglected for this particular mission [22]. The atmosphere is modeled with the US76 model. The aerodynamics model has been derived by the DLR Aerodynamics Institute. 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. 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 it was not necessary. Reference values for the controls are reported 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 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]

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

$$J = w_\theta(\theta(t_f) - \theta_{f,ref})^2 + w_\phi(\phi(t_f) - \phi_{f,ref})^2 \tag{1}$$

so that the distance w.r.t. the final nominal longitude and latitude is minimized. The weights w_θ and w_ϕ are in this case assumed equal to one. The solution has to satisfy the following equations of motions [13, 18],

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

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. σ and u_σ are the bank angle and the bank-angle rate, respectively. D and L are the drag and lift accelerations, while g is the gravity acceleration. Finally, ω is the Earth's rotation rate, equal to $7.2921 \cdot 10^{-5}$ rad/s.
Both the states and the controls are bounded, i.e.

$$\begin{Bmatrix} 0 \text{ km} \\ -180 \\ -90 \\ 10 \text{ m/s} \\ -45 \\ -180 \end{Bmatrix} \leq \begin{Bmatrix} h \\ \theta \\ \phi \\ V \\ \gamma \\ \psi \end{Bmatrix} \leq \begin{Bmatrix} 120 \text{ km} \\ 180 \\ 90 \\ 7000 \text{ m/s} \\ 30 \\ 180 \end{Bmatrix} \tag{3}$$

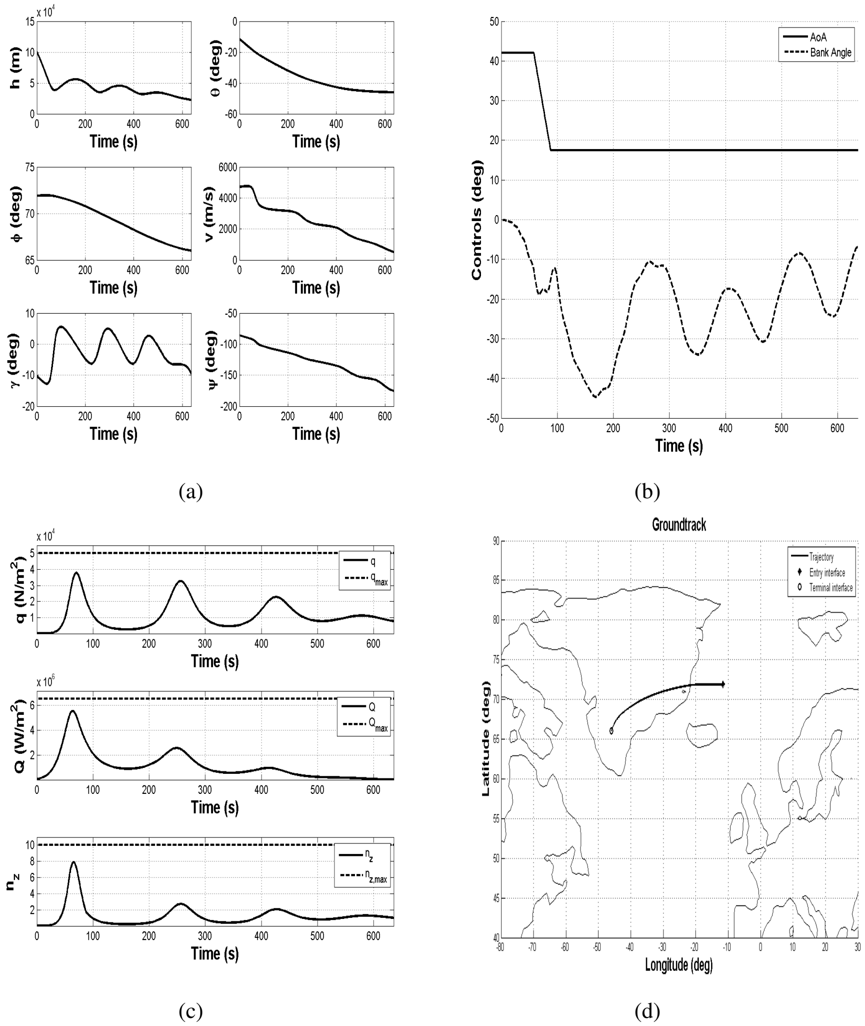
Three constraints are included in the transcription, that is, the dynamic pressure $\overline{q}$, the heat flux $\dot{Q}$ (computed by using the cold-wall model for laminar boundary layer), and the load factor n_z, which can be computed according to

$$\overline{q} = \frac{1}{2}\rho V^2 \tag{4}$$

$$\dot{Q} = k_q \sqrt{\rho} V^3 \tag{5}$$

$$n_z = \frac{|L\cos\alpha + D\sin\alpha|}{g_0} \tag{6}$$

where ρ is the air density, expressed in kg/m^3, k_q is a constant depending on the material and the geometry of the thermal protection system, and g_0 is equal to the gravity acceleration at sea level. Maximum allowed values are $5 \cdot 10^4$ N/m^2, $6.5 \cdot 10^6$ W/m^2 and 10, respectively. With these definitions, the optimal control problem to be solved is completely characterized. Further details about the implemented transcription method and the verification of the solution can be found in [21]-[23].
Figures 2(a)-(d) show the nominal solution of the corresponding OCP. From the analysis of Fig. 2(a), it is possible to see that the vehicle follows a skipped entry. This is due to the limits of the controls, which cannot allow for a shallower entry (i.e., no inverted flight of the vehicle is allowed). Indeed, the flight-path angle oscillates between about -12 deg and 8 deg, and this is mainly a consequence of the initial flight-path angle, and of the limits on the bank angle, which can assume values in the range $[-60, 60]$ deg. Figure 2(b) shows the controls. In terms of planning of trajectory, the angle of attack α is predefined. Specifically it has an upper value equal to 42 deg, and a lower value, equal to 17.5 deg, with a linear transition between the two. The saturation limits both on the bank angle and on the bank angular rate are satisfied too. Figure 2(c) shows the constraints, that is, dynamic pressure, heat-flux, and load factor. All these constraints are within the prescribed limits. Finally, in Fig. 2(d), the nominal groundtrack is plotted, starting from the entry interface. The nominal mission is over once the TAEM condition is achieved. As expected, the terminal position in terms of longitude and latitude is achieved with a very good approximation. The limits on the control rates are also respected.



(a) (b)

(c) (d)

Fig. 2 Nominal solution for Entry Guidance of SHEFEX-3: (a) states, (b) controls, (c) constraints, and (d) groundtrack.

4 Multivariate Interpolation Technique

The purpose of the multivariate interpolation technique is to merge several precomputed trajectories to have a solution corresponding to the actual inflight conditions. These trajectories are associated to different values that a parameter or a series of parameters $\mathbf{P}_c$ can assume. However, extensive trajectory databases can exceed memory limits available onboard. To overcome this problem, the multivariate interpolation process has been integrated with pseudospectral methods to store onboard low-density (LD) discrete solutions. They are online converted into a more useful high-density (HD) discrete solution, able to properly represent the trajectory. The trajectory synthesis via multivariate interpolation can then be seen as a three-phase process:

- Definition and discretization of the parameter space
- Generation of low-density discrete solution
- Generation of high-density discrete solution

In the following subsections, these aspects, representing the multivariate pseudospectral interpolation technique, are described.

4.1 Definition and discretization of the parameter space

Assume a parameterization of the OCP defined in Sec. 3 via a set of d scalar parameters (i.e. initial conditions, physical parameters etc.) $p^1, \ldots, p^d$. Each of the parameters is defined on a compact subset of the real numbers $\mathbb{I}_i \subset \mathbb{R}, i = 1, \ldots, d$, such that the cartesian product

$$\mathbb{R}^d \supset \mathbf{P}_c = \prod_{i=1}^{d} \mathbb{I}_i = \mathbb{I}_1 \times \cdots \times \mathbb{I}_d, \tag{7}$$

defines the parameter space $\mathbf{P}_c$. 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-dimensional domain of a single reference parameter is discretized using a finite number of discrete points. Therefor let

$$\mathbf{p}^i = \left\{ p^i_1, \ldots, p^i_{n_i} \right\} \tag{8}$$

define a strictly monotonically increasing set for each $i = 1, \ldots, d$. The Cartesian product of the sets given by (8) defines a d-dimensional $n_1 \times \cdots \times n_d$-rectangular grid

$$\mathbf{P} = \prod_{i=1}^{d} \mathbf{p}^i = \mathbf{p}^1 \times \cdots \times \mathbf{p}^d, \tag{9}$$

which can be seen as a discretization of the parameter space $\mathbf{P}_c$ defined by (7). The set $\mathbf{P}$ consists of $n_G = \prod_{i=1}^{d} n_i$ elements and can equivalently be represented as a combination of all the grid points $\mathbf{p_i}, \mathbf{i} = (i_1, \ldots, i_d)$, where $\mathbf{p_i} = \left(p^1_{i_1}, \ldots, p^d_{i_d} \right) \in \mathbb{R}^d$, such that

$$\mathbf{P} \cong \{\mathbf{p_i}\}_{i_1=1,\ldots,i_d=1}^{n_1\ ,\ldots,n_d} \,. \tag{10}$$

The parameters space is therefore completely defined.

4.2 Generation of low-density discrete solution

As stated in the previous section, a solution of the OCP defined in Sec. 3 depends not only on the specific choice of a parameter vector $\mathbf{p} \in \mathbf{P}_c$ but also on time. Within this work the computation of a solutions of an OCP is based on the solution of a corresponding NLP. The interpolation approach described in this section shows how to compute a discrete point of an interpolated reference trajectory belonging to a given discrete point in time, while the discretization of the time interval is assumed to be given by the transcription method. In case of a transcription based on flipped Radau pseudospectral (FRP) method the following interpolation algorithm has to be applied to each of the collocation nodes defined by the roots of the FRP (these polynomials are properly defined in Sec. 4.3.1). The set of all collocation nodes can be seen as the domain of the low-density discrete solution.
The solution of the OCP corresponding to an element $\mathbf{p} = (p_1, \dots, p_d)$ of $\mathbf{P}$ is denoted by $f(t,\mathbf{p}) = f(t, p_1, \dots, p_d)$ while $f_\tau(\mathbf{p})$ denotes the solution of the corresponding NLP at $\mathbf{p} \in \mathbf{P}_c$ and a given discrete point in time.
The generation of a trajectory database corresponding to a given grid $\mathbf{P}$ includes the offline computation of n_G solutions $f_{\mathbf{i}} = f(t, \mathbf{p_i})$ of the corresponding NLP.
For each trajectory (i.e. states and control inputs) and each point in time the resulting data base consist of n_G pairs

$$\left(\mathbf{p_i}, f_{\tau,\mathbf{i}}\right)_{i_1=1,\dots,i_d=1}^{n_1\ ,\dots,n_d} = \left(\mathbf{p}^1, \dots \mathbf{p}^d, \mathbf{F}_\tau\right), \tag{11}$$

where the grid points $\mathbf{p_i}$ are called supporting points while the $f_{\tau,\mathbf{i}}$ are called supporting values.
A tensor product spline $s \in \mathbb{S}_{k_1,\mathbf{t}_1} \bigotimes \dots \bigotimes S_{k_d,\mathbf{t}_d}$ on a grid $\mathbf{P}$ at a grid point $\mathbf{p} \in \mathbf{P}$ is in general defined by

$$s(\mathbf{p}) = \sum_{i_1=1}^{m_1} \dots \sum_{i_d=1}^{m_d} c_{i_1,\dots,i_d} B_{i_1,k_1}(p_1) \dots B_{i_d,k_d}(p_d). \tag{12}$$

In (12) $B_{i,k}$ denotes the i^{th} B-spline of order k for a given non-decreasing knot vector $\mathbf{t} = (t_i)_{i=1}^{m+k}$. The coefficients

$$\mathbf{C} = \left(c_{i_1,\dots,\mathrm{\imath}_d}\right)_{i_1=1,\dots,i_d=1}^{m_1\ ,\dots,m_d} \tag{13}$$

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

$$s(\mathbf{p_i}) = f_{\tau,\mathbf{i}}, \forall \mathbf{i} \in 1 \dots n_G. \tag{14}$$

Since every B-spline in (12) 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

[6]. A detailed description of the generalization of the univariate spline interpolation to the so-called tensor product spline approach can be found in [11, 14].
Depending on the choice of k and a suitable knot vector $\mathbf{t}$, the tensor product spline interpolation in (12) 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 usage of different interpolation methods for each of the d dimensions.
Within this work the grid $\mathbf{P}$ consists of two supporting points for each direction, such that $n_i = 2, i = 1, \dots, d$ and

$$\mathbf{P} = \left\{p_1^1, p_2^1\right\} \times \cdots \times \left\{p_1^d, p_2^d\right\}. \tag{15}$$

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

$$\mathbf{t}_i = \left(t_j^i\right)_{j=1}^4 = \left(p_1^i, p_1^i, p_2^i, p_2^i\right), \qquad i = 1, \dots, d \tag{16}$$

defines a piecewise linear interpolation in each direction on the given d-dimensional grid.
For $\mathbf{C} = \mathbf{F}_\tau$ the piecewise linear interpolation method fulfills the interpolation condition given by (14) 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 $\mathbf{p} \in \mathbf{P}_c$ is described by the pseudo code according to Algorithm 1. Algorithm 1 requires the knowledge of the interpolating spline $s \in \mathbb{S}_{2,\mathbf{t}_1} \bigotimes \dots \bigotimes \mathbb{S}_{2,\mathbf{t}_d}$. Details about the computation of the interpolating spline can be found in [11].

Tensor Product Spline Interpolation

```
Data: Given: knot vector $\mathbf{t}$, coefficients $\mathbf{C}$, spline $s \in \mathbb{S}_{2,\mathbf{t}_1} \bigotimes \dots \bigotimes \mathbb{S}_{2,\mathbf{t}_d}$, evaluation point
      $\mathbf{p} \in \mathbf{P}_c$
$\mathbf{A}_0 = \mathbf{C}$;
for i = 1 : $d$
    $\mathbf{A}_i$ = EvalUnivSpline($\mathbf{t}_i, \mathbf{A}_{i-1}, p_i$);
    $\mathbf{A}_i$ = $\mathbf{A}_i'$;
end
$s(\mathbf{p}) = \mathbf{A}_d$;
Result: interpolated values $s(\mathbf{p})$
```

Algorithm 1: Tensor Product Spline Interpolation

The EvalUnivSpline($\mathbf{t}, \mathbf{c}, p$) function in Algorithm 1 denotes the evaluation of a univariate spline with coefficients $\mathbf{c}$ at a point p while the operator $(\cdot)'$ performs a cyclic rotation, such that $\mathbf{A} \in \mathbb{R}^{n_1, n_2, \dots, n_d} \Rightarrow \mathbf{A}' \in \mathbb{R}^{n_2, \dots, n_d, n_1}$.
An interpolation based on Algorithm 1 allows to generate reference trajectories for

the states and the control inputs of the assumed system for each element of the parameter space $\mathbf{p} \in \mathbf{P}_c$. It is important to emphasize that interpolated reference 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.

4.3 Generation of high-density discrete solution

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 big amount of data onboard. Legendre polynomials can be used for this purpose.

4.3.1 Flipped Legendre-Radau polynomials

Legendre polynomials belong to the class of more general Jacobi polynomials, and are defined as follows.

$$L_n(\tau) = \frac{1}{2^n n!}\frac{d^n}{d\tau^n}\left[(\tau^2-1)^n\right] \tag{17}$$

These polynomials are defined over the real axis $\mathbb{R}$. They represent a family of orthogonal polynomials over the interval $[-1,1] \in \mathbb{R}$. It is possible to demonstrate that the family of polynomials $R_n(\tau)$ defined as a combination of the Legendre polynomials of order n and $n-1$

$$R_n(\tau) = L_n(\tau) - L_{n-1}(\tau) \tag{18}$$

is orthogonal too, in the domain $(-1,1]$. The polynomial defined in (18) is called flipped Radau polynomial (FRP), (or alternatively flipped Gauss-Radau polynomial) and is the fundamental entity for the formulation of the flipped Radau pseudospectral method.

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, it is possible to build this approximating function as

$$F(\tau) \cong \sum_{i=0}^{N} F_i P_i(\tau) \tag{19}$$

where the terms $P_i(\tau)$ are defined as

$$P_i(\tau) = \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{\tau - \tau_k}{\tau_i - \tau_k} \tag{20}$$

The following properties, together with the ease of implementation, justify the choice of using the pseudospectral methods for the characterization of the discrete domain.

- 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 [8].
- Removal of the Runge phenomenon.

The removal of the Runge phenomenon[1] 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, with no need to evaluate splines. This approach significantly reduces the onboard memory requirements, as well as the onboard CPU burden.

4.3.2 Computation of Pseudospectral-based high-density 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 $(n_s+n_c) \times (N_{LD}+1)$.

$$\mathbf{T}_{LD} = \left\{ \begin{matrix} \mathbf{X}_{LD} \\ \mathbf{U}_{LD} \end{matrix} \right\} = \left\{ \begin{matrix} \mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_{N_{LD}} \\ \mathbf{U}_0, \mathbf{U}_1, ..., \mathbf{U}_{N_{LD}} \end{matrix} \right\} \tag{21}$$

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

$$\tilde{\mathbf{T}}_{HD} = \left\{ \begin{matrix} \tilde{\mathbf{X}}_{HD} \\ \tilde{\mathbf{U}}_{HD} \end{matrix} \right\} = \left\{ \begin{matrix} \tilde{\mathbf{X}}_0, \tilde{\mathbf{X}}_1, ..., \tilde{\mathbf{X}}_{N_{HD}} \\ \tilde{\mathbf{U}}_0, \tilde{\mathbf{U}}_1, ..., \tilde{\mathbf{U}}_{N_{HD}} \end{matrix} \right\} \tag{22}$$

where $N_{HD}+1$ is the number of points representing the HD discrete solution. Moreover, the HD time vector $\tilde{t}_{HD}$ must be computed. If we combine equations (19) and

[1] 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 [19].

(20), we can write

$$F(\tau) = \sum_{i=0}^{N_{LD}} F_i \prod_{\substack{k=0\\k\neq i}}^{N_{LD}} \frac{\tau - \tau_k}{\tau_i - \tau_k}, \ \tau \in [-1,1] \tag{23}$$

F_i represents a low-density variable. It can be replaced with the p^{th} row of $\mathbf{T}_{LD}$. 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

$$\tilde{T}^p_{HD}(\tilde{\tau}_m) = \sum_{i=0}^{N} T^p_{LD,i} \prod_{\substack{k=0\\k\neq i}}^{N} \frac{\tilde{\tau}_m - \tau_k}{\tau_i - \tau_k}, \ p = 1,...(n_s+n_c), \ m = 0,...(N_{HD}) \tag{24}$$

The relationship (24) can be extended to all the rows of the matrix $\tilde{\mathbf{T}}_{HD}$, and rewritten in matrix form as

$$\tilde{\mathbf{T}}_{HD} = \mathbf{T}_{LD} \cdot \mathbf{P}_{FRP} \tag{25}$$

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

$$\mathbf{P}_{FRP} = \begin{bmatrix} \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_0-\tau_k}{\tau_0-\tau_k} & \cdots & \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_{N_{HD}}-\tau_k}{\tau_0-\tau_k} \\ \cdots & \cdots & \cdots \\ \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_0-\tau_k}{\tau_{N_{LD}}-\tau_k} & \cdots & \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_{N_{HD}}-\tau_k}{\tau_{N_{LD}}-\tau_k} \end{bmatrix} \tag{26}$$

The vector $\tilde{\tau}$ is the high-density discrete pseudotime vector defined between -1 and 1. Since both the nodes τ where the solutions are computed, and the nodes $\tilde{\tau}$ 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 the equation (25). To complete the generation of the HD solution, we still need the HD discrete physical time vector associated to the interpolated solution. It can be computed by using the following expression.

$$\tilde{t}_m = \frac{\tilde{t}_f - \tilde{t}_0}{2}\tilde{\tau}_m + \frac{\tilde{t}_f + \tilde{t}_0}{2}, \ \ m = 0,...,N_{HD} \tag{27}$$

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 Sec. 4 to the final times stored in the trajectory database. The trajectory representing the feedforward guidance solution is completely generated with the application of the algorithm **1** and the equations (25) and (27).

5 Generation of Trajectories Database

Once the mathematical tools needed for the online trajectory generation have been described, we can apply them to a trajectory database. Therefore, the following problem is the trajectory-database generation. For a complex mission such as the atmospheric entry, the inflight conditions can be significantly different w.r.t. the nominal ones, and this aspect directly affects the database size. Therefore, the reference parameters which could vary need to be identified. The driving idea is to perform a mapping of the range we are interested into cover, and to generate a database of trajectories, which fullfill the requirements defined in Sec. 2 for the entire region of interest. This information will be then processed online and used to adapt the feedforward guidance to the current situation as described in Sec. 4. To realize this scheme, we can identify the following steps.

- Identification of region of interest
- Trajectories computation

5.1 Identification of Region of Interest

The first step is the identification of a region of interest which covers, with a reasonable probability, the inflight conditions at the entry interface of SHEFEX-3. Let us 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 (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, whose vertices represent the 2^3 possible combinations of values (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.

$$\begin{aligned} h &\in [h_L, h_U] \\ \theta &\in [\theta_L, \theta_U] \\ \phi &\in [\phi_L, \phi_U] \\ V &\in [V_L, V_U] \\ \gamma &\in [\gamma_L, \gamma_U] \\ \psi &\in [\psi_L, \psi_U] \end{aligned} \tag{28}$$

We can describe this multidmensional uncertainty as a *hexeract* (Fig. 3(d)), which is a member of the hypercubes family, characterized by having dimension equal to

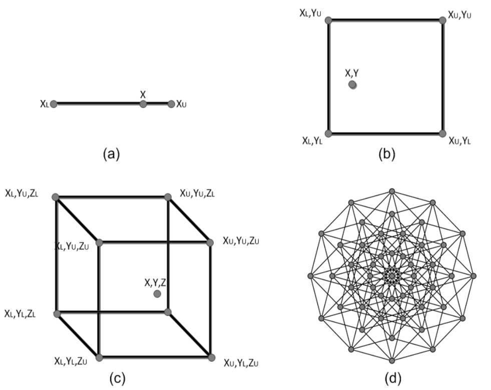


Fig. 3 Geometrical Representation of Regions of Interest: (a) 1-D, (b) 2-D, (c) 3-D, (d) 6-D.

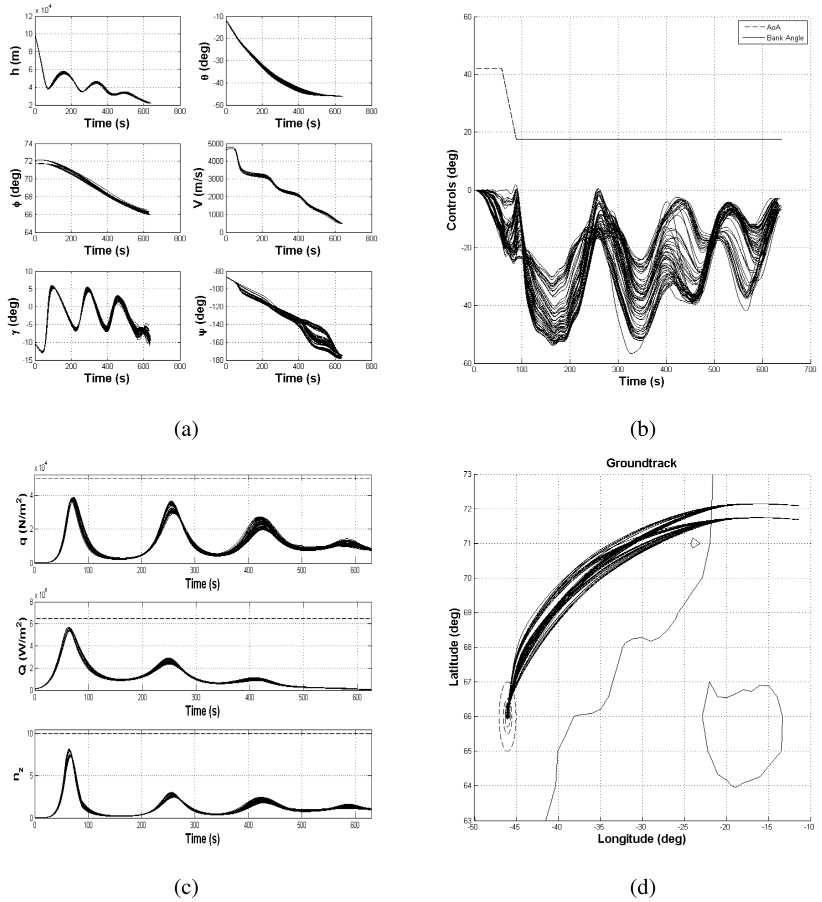
six. The hypervertices of the hexeract will represent the 2^6 possible extreme initial conditions used to solve the optimal-control problems, and the corresponding optimal trajectories represent our database. The multivariate interpolation approach will be able to combine them and provide a solution for any condition enclosed in the hypervolume $\in \mathbb{R}^6$ of the hexeract. The nominal conditions for SHEFEX-3 will represent then the hypercenter of our hexeract. To validate the method, the following region of interest around the nominal initial state has been selected.

$$\begin{array}{c} \delta h \in [-250, +250] \text{ m} \\ \delta\theta \in [-0.2, +0.2] \text{ deg} \\ \delta\phi \in [-0.2, +0.2] \text{ deg} \\ \delta V \in [-50, +50] \text{ m/s} \\ \delta\gamma \in [-0.2, +0.2] \text{ deg} \\ \delta\psi \in [-0.2, +0.2] \text{ deg} \end{array} \tag{29}$$

It is worth to say that the database can be further extended by synthesizing more hexeracts. Since two hexeracts will have a tesseract (i.e., a 5-D hypercube) in common, that is 2^5 trajectories, each additional hexeract will imply the computation of 2^5 more trajectories.

5.2 Trajectories Computation

All trajectories have been generated with SPARTAN (SHEFEX-3 PseudoSpectral Algorithm for Reentry Trajectory ANalysis), a tool developed at DLR based on the flipped Radau pseudospectral method, that exploits the Jacobian structure of the NLP [23], and uses linear and nonlinear automatic scaling methods [21]. It has already been used for computing SHEFEX-3 alternative scenarios [22], and for lunar landing reachability analysis [1, 2]. Each of the computed 2^6 trajectories has been formulated according to the description reported in Sec. 2, except for different initial conditions. Plots representing the states, the controls, the constraints, and the groundtracks obtained are shown in Fig. 4 (a)-(d). The states and the controls reported in 4(a) and 4(b) show that the obtained envelope satisfies the requirements. The constraints in Fig. 4(c) are also within the prescribed boundaries, and their evolution is a direct consequence of the skip entry. The groundtracks reported in Fig. 4(d), show that all trajectories end in the proximity of the final reference longitude and latitude. Small deviations were observed for a few trajectories because of the different initial conditions, but still within the allowed footprint limits.



(a) (b)

(c) (d)

Fig. 4 Trajectories Database: (a) States, (b) Controls, (c) Constraints, (d) Trajectories.

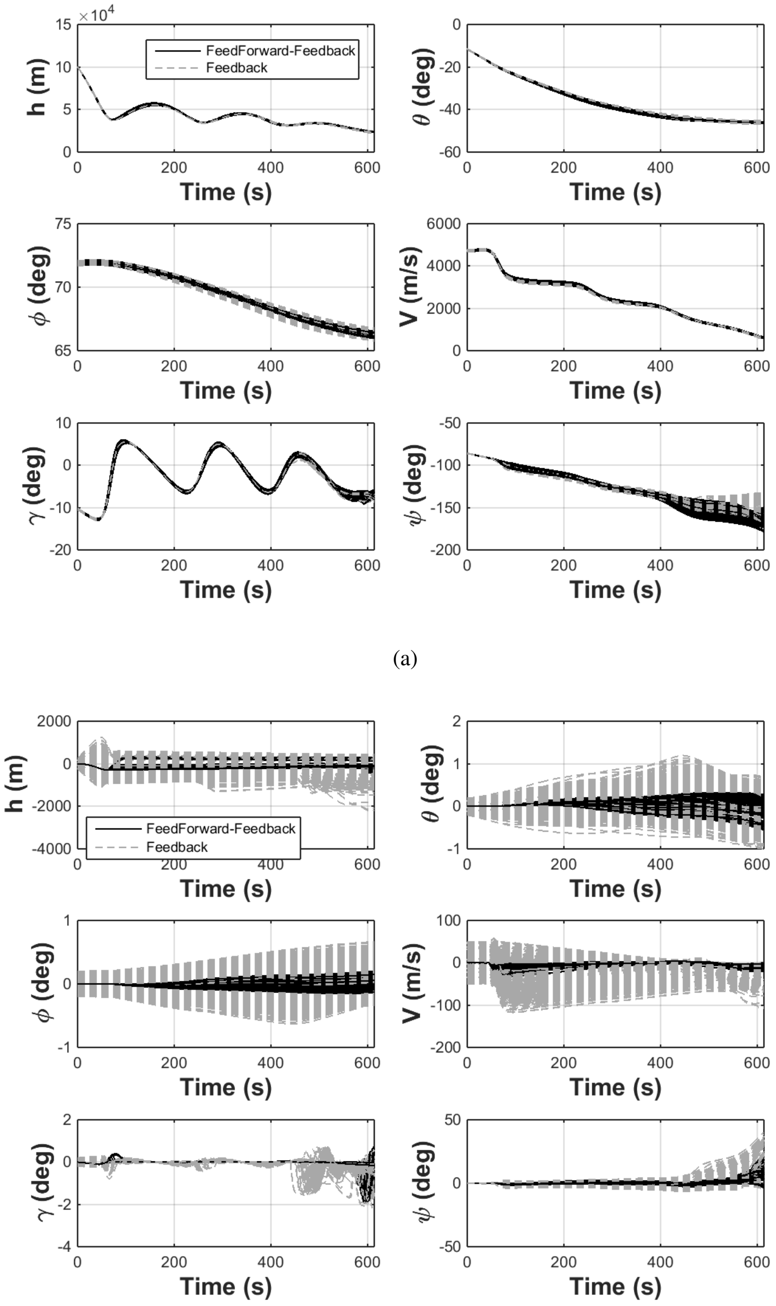
6 Numerical Results

6.1 Simulation Campaign

For the validation of the proposed method, a MonteCarlo campaign of 500 cases has been simulated. For each case, random dispersions within the limits of the database extension have been generated. The feedforward solution is generated according to the initial conditions by using the proposed multivariate pseudospectral interpolation approach. A tracking controller based on [28] has been added to the guidance scheme. Results are compared w.r.t. the tracking of the nominal reference solution. The results are reported in the Fig. 5-7. The multivariate approach generates meaningful trajectories, as it can be seen in the Fig. 5(a), (b). The dispersion error is reduced on all the longitudinal states. All the control limits are satisfied, and the feedforward guidance has the effect to reduce the control variations (Fig. 6(a), (b)). A further advantage is that, with the introduction of the feedforward adaptive guidance, the number of cases where the reversals are needed significantly decreases. The consequence is a further improvement of the longitudinal guidance performances, which are deteriorated by the bank reversals. In terms of constraints (Fig. 8) the maximum values are consistent with the ones associated to the nominal scenario. Moreover, the peaks which are reduced by using the feedforward-feedback approach w.r.t. the corresponding cases which use the sole feedback controller applied to the nominal trajectory. Figures 7(a), (b) show a significant improvement for what regards the dispersion errors (i.e., in terms of latitude and longitude). The online adaptation of the trajectory brings therefore a reduction of the dispersion area, and in general, an improvement of the performances. Specifically, the dispersion area is reduced by about 76%. This information is summarized in Table 3 as well, which reports how the terminal positions obtained with the two methods are distributed in the three dispersion ellipses previously defined. The number of cases which fall into the finest ellipse substantially increases when the adaptive feedforward scheme is used. About 76% of the cases satisfies the stricter requirements against 15% of the cases associated to the tracking of the nominal trajectory with the feedback controller. Moreover, when the feedforward-feedback controller is used, no cases which fall outside the coarse ellipse are observed, while with the feedback controller 1.8% of the cases do not satisfy the dispersion requirements in terms of longitude and latitude.

Table 3 Dispersion Analysis

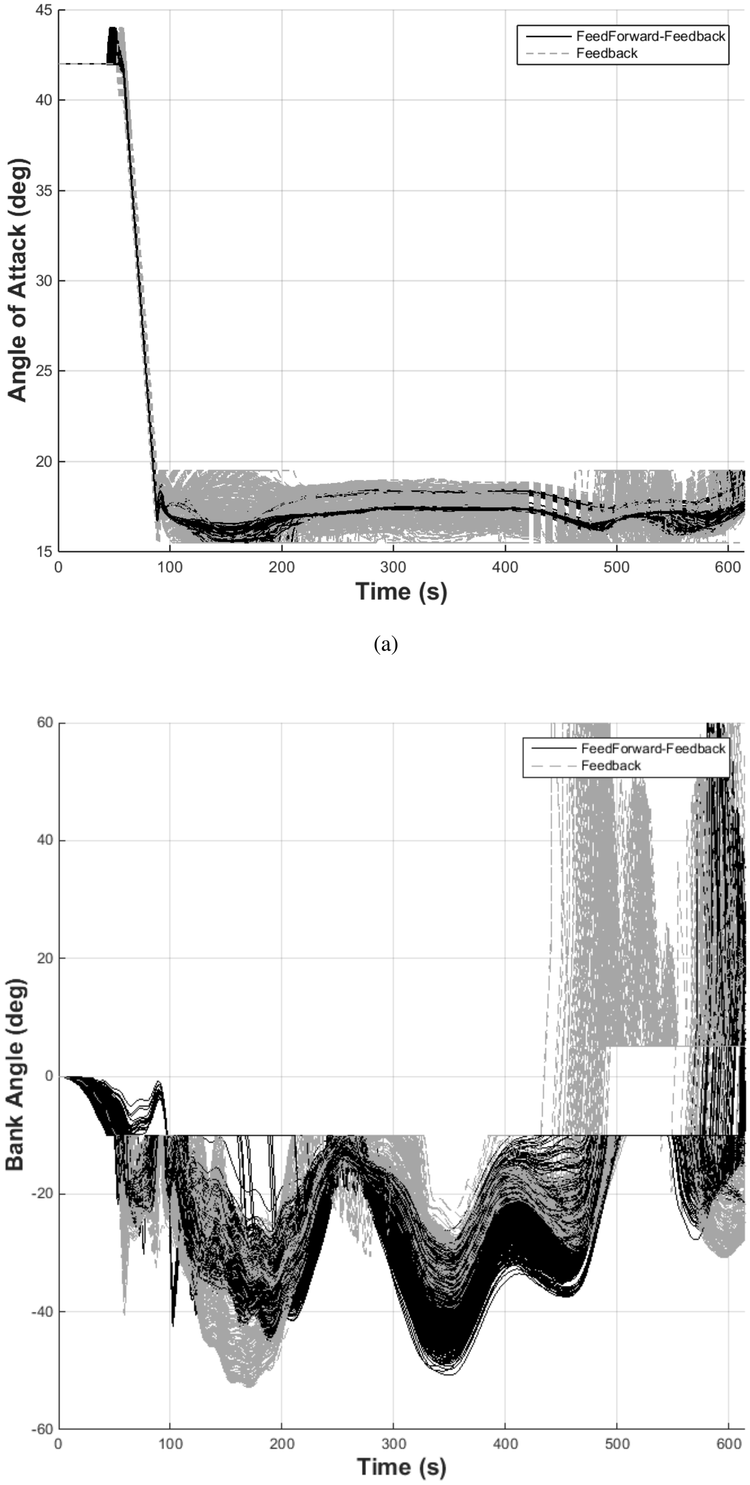
Ellipse / Controller	$[0.25 \times 0.25]$	$[0.5 \times 0.5]$	$[1 \times 1]$	Outside
FeedForward-FeedBack	382	105	13	0
FeedBack	75	221	195	9



(a)

(b)

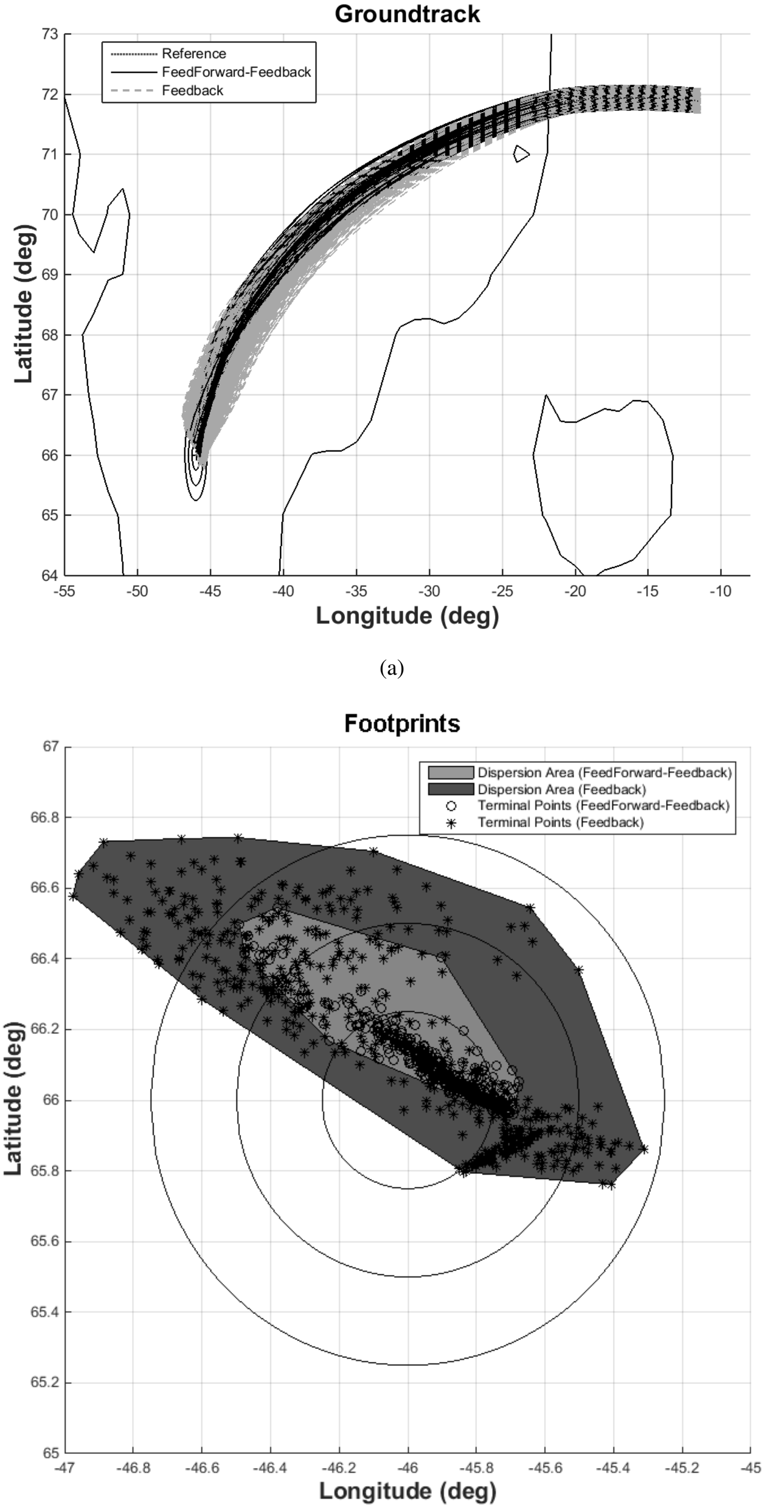
Fig. 5 Multivariate interpolated trajectory vs Nominal tracked trajectory: (a) States, (b) Errors on states.



(a)

(b)

Fig. 6 Multivariate interpolated trajectory vs Nominal tracked trajectory: (a) Angle of Attack, (b) Bank Angle.



(a)

(b)

Fig. 7 Multivariate interpolated trajectory vs Nominal tracked trajectory: (a) Trajectories, (b) Footprints.

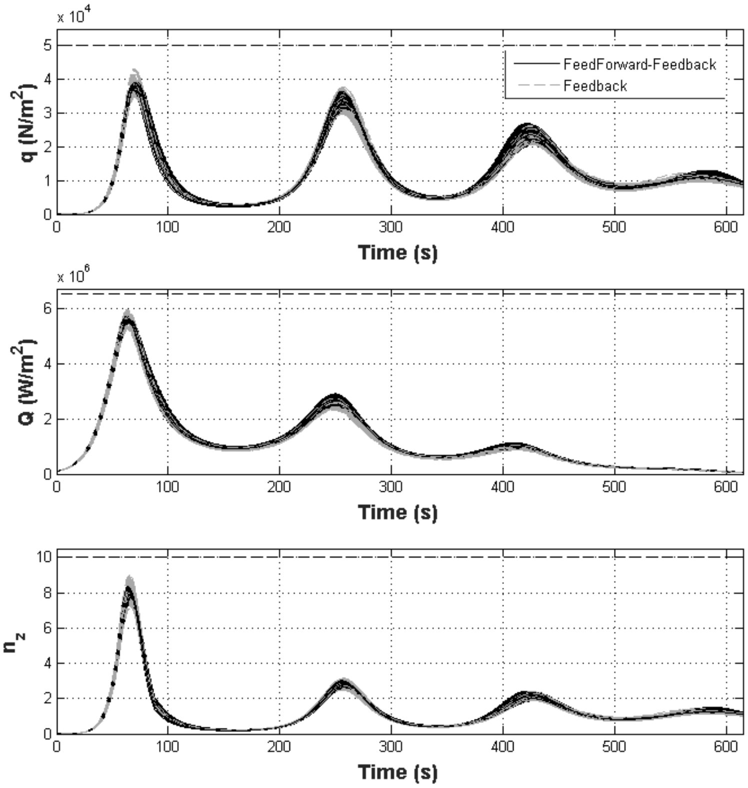


Fig. 8 Multivariate interpolated trajectory vs Nominal tracked trajectory: Constraints.

6.2 Real-Time capability: CPU time comparison

To give an idea about the real-time capability of the proposed approach, the ratio between the required CPU time for synthesizing optimal trajectories and interpolated trajectories for the first 100 cases is reported in Fig. 9. It is possible to see that the interpolation technique is faster than the optimal trajectory generation with a factor varying between about 69 and 619. The mean CPU time ratio is about 195. Therefore, the CPU time required to compute onboard a valid solution is strongly reduced. In absolute terms, the mean time required to generate an optimal trajectory is 28.32 sec, while only 0.147 sec are required to compute the interpolated trajectory[2].

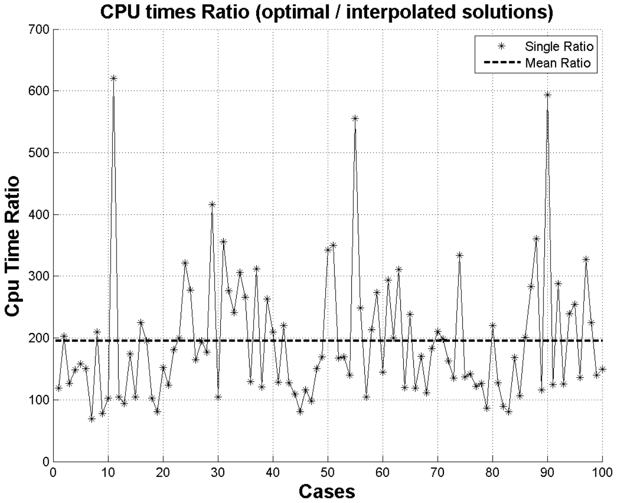


Fig. 9 CPU time analysis: Ratio between time required to compute an optimal trajectory and the time required to compute the corresponding trajectory via multivariate interpolation.

7 Conclusions

In this work the multivariate interpolation approach has been coupled with pseudospectral methods to generate nearly-optimal real-time feedforward guidance solutions for entry scenarios. Taking advantage from the pseudospectral transcription,

[2] All the simulations have been performed with a laptop having an i7M640 CPU with a clock frequency of 2.80 GHz and 4 GB of RAM.

the synthesis can be efficiently performed by processing information stored in a ”hexeract” of trajectories. Specifically, it has been shown that the synthesis can be efficiently represented as a dot product between a low-density interpolated solution and a pre-computed matrix. The MonteCarlo campaign has demonstrated the feasibility of this approach, having as further advantage a significant improvement in the guidance performances, analyzed both for the constraints peaks and the dispersion in terms of final longitude and latitude. A comparison of the times required to synthesize a solution with optimal control, and to compute it by using the multivariate pseudospectral interpolation scheme has confirmed that the method is suitable for real-time applications, since the CPU times is reduced by a factor between 69 and 695. The proposed algorithm can thus be considered as an efficient method for real missions, where the CPU power and the real-time capabilities of the onboard system are key factors for the selection of the guidance solutions.

References

1. Y. E. Arslantas, T. Oehlschlägel, M. Sagliano, S. Theil, C. Braxmaier: Safe Landing Area Determination for a Moon Lander by Reachability Analysis. 17-th International Conference and Control (HSSC) Berlin, Germany, (2014)
2. Y. E. Arslantas, T. Oehlschlägel, M. Sagliano, S. Theil, C. Braxmaier: Approximation of Attainable Landing Area of a Moon Lander by Reachability Analysis. International Astronautical Conference. IAC-14-C.1.7.2, Toronto, Canada, (2014)
3. J. T. Betts: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2nd ed., SIAM, Philadelphia, (2010)
4. I. Bogner: Description of Apollo Entry Guidance. NASA Technical Memorandum CR-110924 (1966)
5. K. P. Bollino: High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles. Ph.D. Dissertation, Mechanical and Astronautical Engineering Dept., Naval PostGraduate School., (2006)
6. C. De Boor: A Practical Guide to Splines. Springer, New York (2001)
7. P. E. Gill, W. Murray, M. A. Saunders: 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)
8. Q. Gong, I. M. Ross, W. Kang, F. Fahroo: 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
9. J., C. Harpold, C. A. Graves Jr.: Shuttle Entry Guidance. Journal of the Astronautical Sciences, Vol.27 No. 3, (1979)
10. M. Knauer, C. Büskens C: From WORHP to TransWORHP. 5th International Conference on Astrodynamics Tools and Techniques, Noordwijk (2012)
11. E. Lockner: Echtzeitfähige Trajektorien-Synthese mittels multivariater Interpolationsverfahren am Beispiel eines Mondlandemanövers. Diploma thesis, University of Bremen (2011)
12. E. Lockner, T. Oehlschlägel, S. Theil, M. Knauer, J. Tietjen, C. 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)
13. P. Lu , J. M. Hanson: Entry Guidance for the X-33 Vehicle. Journal of Spacecraft and Rockets, Vol.35 No. 3, (1998)
14. T. Lyche, K. Morken: Spline Methods. Department of Informatics, University of Oslo. http://www.uio.no/studier/emner/matnat/ifi/INF-MAT5340 (2011)

15. K. D. Mease, J.P. Kremer: Shutte Entry Guidance Revisited Using Nonlinear Geometric Methods. Journal of Guidance, Control and Dynamics, Vol.17 No. 6 (1994)
16. K. D. Mease, D. T. Chen, P. Teufel, H. Schöneberger: Reduced-Order Entry Trajectory Planning for Acceleration Guidance. Journal of Guidance, Control and Dynamics, Vol.25 No. 2, (2002)
17. E. Mooij: Robustness Analysis of an Adaptive Re-entry Guidance System, AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA 2005-6146, San Francisco, CA (2005)
18. E. Mooij: Characteristics Motion of Re-entry Vehicles, AIAA Atmospheric Flight Mechanics (AFM) Conference, AIAA 2013-4603, Boston, MA (2005), doi:10-2514/6.2013-4603
19. C. Runge: Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. Zeitschrift fü Mathematik und Physik Vol. 46, 224-243 (1901)
20. I. M. Ross, P. Sekhavat, A. Fleming, Q. Gong: Optimal Feedback Control: Foundations, Examples, and Experimental Results for a New Approach. Journal of Guidance, Control and Dynamics, Vol.31 No. 2 (2008)
21. M. Sagliano: 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. M. Sagliano, M. Samaan, S. Theil, E. Mooij: 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
23. M. Sagliano, S. Theil: 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
24. A. Saraf, J.A. Levitt, K.D. Mease, M. Ferch: Landing footprint computation for entry vehicles. AIAA Guidance, Navigation and Control Conference and Exhibit, Providence, RI, (2004)
25. S. R. Steffes: Development and Analysis of SHEFEX-2 Hybrid Navigation System Experiment. Fachbereich Produktionstechnik, Universität Bremen, Bremen, (2012)
26. B. Singh, R. Bhattacharya: Optimal Guidance of Hypersonic Vehicles Using B-Splines and Galerkin Projection. AIAA Guidance, Navigation, and Control (GNC) Conference. AIAA 2008-7263, Honolulu, HA, (2008)
27. A. Wächter, L.T. Biegler, On the implementation of an interior-point filter linesearch algorithm for large-scale nonlinear programming, Math. Program. 106(1) Springer-Verlag, New York (2006)
28. M. Sagliano, S. Theil, E. Mooij: Robust Nonlinear Tracking Controller for Hypersonic Entry Vehicles. To be submitted (2015)