REAL-TIME ATMOSPHERIC ENTRY TRAJECTORY COMPUTATION USING PARAMETRIC SENSITIVITIES

D. Seelbinder

German Aerospace Center (DLR)
Institute of Space Systems
Robert Hooke Straße 7
28359 Bremen

C. Büskens

University of Bremen
Center for Technomathematics
Bibliothekstraße 1
28359 Bremen

ABSTRACT

The real-time generation of optimal trajectories and controls for nonlinear systems is a technology of interest to many applications. The online solution of an optimal control problem (OCP) is often not computationally feasible on current embedded systems hardware. We present a method to generate a near-optimal control sequence and the corresponding state trajectory based on the parametric sensitivity analysis (PSA) of nonlinear programs (NLPs) which does not require performing the classical gradient based NLP process online and hence reduces the computational load. The OCP is transcribed into a parametric NLP which is solved offline for a nominal set of parameters. Additionally the parametric sensitivities of the optimal solution with respect to different types of perturbations are computed at discrete points along the nominal trajectory. The sensitivities are used online in a Taylor expansion of the nominal solution and an iterative feasibility and optimality restoration to compute a new near-optimal control sequence and trajectory from an off-nominal state to the terminal set without solving a NLP. This process is repeated iteratively in the neighborhood of the nominal trajectory. The proposed method is demonstrated for the guided, hypersonic entry of a small capsule into the Martian atmosphere. The PSA algorithm is used as feed forward command and trajectory generation to provide the input for a drag-energy tracking controller.

***Index Terms*—** Optimal Control, Sensitivity, Re-entry, Guidance

1. INTRODUCTION

This work develops an atmospheric entry guidance system combining methods from sensitivity analysis for trajectory computation and drag control for trajectory tracking.

A theoretical basis for the sensitivity analysis of optimal control processes is formulated by Maurer and Pesch [1][2]. In this work a direct optimization approach is used to solve an OCP and the parametric sensitivities of the discretized OCP, i.e. a nonlinear program, are analyzed [3]. Previous work on this topic includes e.g. [4][5] which is expanded to enable a repeated trajectory computation during flight.

Drag tracking is a technique developed during the Space Shuttle era. Mease and Kremer [6][7] applied nonlinear methods to drag tracking by using feedback linearization. This has since been subject to many other authors e.g. [8][9][10].

The paper is structured as follows: In section 2 the control problem is formulated which is maintained throughout the paper as an example. Section 3 gives an overview over the proposed guidance system before the central ideas of the trajectory generation (section 4) and the trajectory tracking (section 5) are presented. In section 6 a performance analysis of the guidance system is given, including the results of a processor in the loop test (section 6.2). In section 7 properties of the guidance system are discussed before in section 8 some conclusions are given.

2. PROBLEM FORMULATION

The control problem under consideration is the entry of a small capsule into the Martian atmosphere. It is controlled by rotating the lift vector using bank angle modulation. The capsule is assumed to be trimmed at a fixed angle of attack. The entry scenario is characterized by a steep flight path angle, a high ballistic coefficient and a low lift-to-drag ratio of the entry capsule. The scenario is similar to missions investigated in ESA Mars Precision Lander studies. The nominal entry interface point (EIP) is given in table 1. Considered are the dynamics for planetary atmospheric flight over a rotating spherical planet with the independent variable being specific energy E. The potential energy is normalized to zero at the planets surface.

$$E = \frac{v^2}{2} - \left(\frac{Gm_p}{r_p + h} - \frac{Gm_p}{r_p}\right) \tag{1a}$$

$$\dot{E} = -vD \tag{1b}$$

G is the gravitational constant, m_p is the planet mass, r_p is the planet radius and D is the drag acceleration (3b). The energy dynamics for altitude h, longitude λ, latitude φ, air-speed v, flight path angle γ and azimuth χ (with respect to the north) are given by $(\cdot)' = \frac{\mathrm{d}(\cdot)}{\mathrm{d}E} = \frac{\mathrm{d}(\cdot)}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}E}$.

$$h' = -\frac{1}{D}\sin\gamma \tag{2a}$$
$$\lambda' = -\frac{1}{D}\frac{\cos\gamma\sin\chi}{(r_p+h)\cos\varphi} \tag{2b}$$
$$\varphi' = -\frac{1}{D}\frac{\cos\gamma\cos\chi}{r_p+h} \tag{2c}$$
$$v' = \frac{D+g\sin\gamma}{Dv} + \Omega_1 \tag{2d}$$
$$\gamma' = -\frac{L}{D}\frac{\cos\mu}{v^2} - \left(\frac{v}{r_p+h} - \frac{g}{v}\right)\frac{\cos\gamma}{Dv} + \Omega_2 \tag{2e}$$
$$\chi' = -\frac{L}{D}\frac{\sin\mu}{v^2\cos\gamma} - \frac{1}{D}\cos\gamma\sin\chi\tan\varphi + \Omega_3 \tag{2f}$$

The terms $\Omega_1, \Omega_2, \Omega_3$ depending on the planet rotational rate are not written out, but they are included in the subsequently defined optimization problem. The control enters the system as the bank angle μ. The lift- and drag accelerations are

$$L = \frac{1}{2}\rho(h)v^2 c_L(M)\frac{s}{m} \tag{3a}$$
$$D = \frac{1}{2}\rho(h)v^2 c_D(M)\frac{s}{m} \tag{3b}$$

where the atmospheric density ρ is given as a function of altitude. The aerodynamic coefficients c_D, c_L are given as a function of Mach $M = \frac{v}{c(h)}$ with c being the speed of sound given as a function of altitude. Lastly s is the aerodynamic reference surface and m is the entry mass. The average lift-to-drag ratio of the capsule is $L/D \approx 0.2$. The angle of attack assumed to be constant.

The guidance and control task is to steer the capsule to the parachute opening point specified in table 1 while respecting path constraints on the heat flux $\dot{Q}$, dynamic pressure q and load factor n. The values of used constants are given in table 2.

$$\dot{Q} = k_p\sqrt{\frac{\rho}{r_n}}v^3 \leq 1600\,\frac{\mathrm{kW}}{\mathrm{m}^2} \tag{4a}$$
$$q = \frac{1}{2}\rho v^2 \leq 17\ \mathrm{kPa} \tag{4b}$$
$$n = \frac{\sqrt{D^2+L^2}}{g_e} \leq 15 \tag{4c}$$

3. GUIDANCE SYSTEM OVERVIEW

The guidance system is composed of two main loops: In the outer loop adapted trajectories are computed at a low rate, while the inner loop tracks the most recent trajectory at a high rate. The trajectory generation yields a discrete near optimal control sequence with corresponding state trajectory and drag-energy profile. In the inner loop downrange is controlled by tracking the computed drag profile. The tracking controller commands a vertical lift to drag ratio which is converted into a command for the bank angle magnitude. The crossrange is managed with secondary priority and is controlled using the bank angle sign which is computed based on the heading error. The control system architecture is shown in figure 1. The symbols $p_1, \ldots, p_9$ are parameters of the sensitivity analysis described in section 4.1.4.

Table 1. Nominal Entry Interface and Parachute Opening Conditions

State		Value	State		Value
h_0	=	120000 m	h_f	=	10000 m
λ_0	=	0°	λ_f	=	11.3°
φ_0	=	25°	φ_f	=	23.3°
v_0	=	5440.8 m/s	v_f	$\leq$	450 m/s
γ_0	=	$-14.5°$	γ_f		free
χ_0	=	97.4°	χ_f		free

Table 2. Model Parameter

		Value			Value
k_p	=	1.9027E-04 $\sqrt{\mathrm{kg}}$/m	r_n	=	0.7 m
r_p	=	3393940 m	g_e	=	9.806 m/s^2

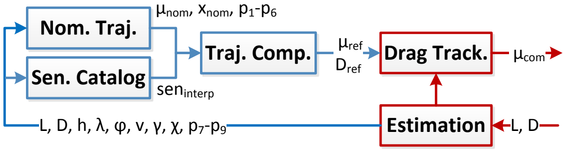


Fig. 1. Guidance system overview

4. OPTIMAL TRAJECTORY COMPUTATION

The trajectory computation strategy is divided into a preparatory offline phase and a real-time capable online phase. The general approach is to view the entry problem as optimal control problem (OCP) and solve it by discretization and transcription into a nonlinear program (NLP). A parametric sensitivity analysis of the optimal NLP solution is performed. The solution and its sensitivities are then used in the online phase for trajectory recomputation.

4.1. Offline Phase

4.1.1. *Optimal Control Process*

The optimal control process immediately follows from the problem definition. All mission constraints are represented as hard constraints, this leaves the objective function to be chosen freely. The path constraint boundaries (4) include large margins and remain inactive for reasonable objective functions. Hence the objective can be used to shape the control function.

To incentivize a low terminal velocity v_f a quadratic penalty term is added. A smooth bank angle profile is desirable to ensure the realized bank angle can always closely match the command. Hence the bank angle time derivative $\dot{\mu}$ is treated as the control variable and quadratically penalized in the objective function. The bank angle μ becomes an auxiliary state. To avoid the singularities at bank angles of $\mu = 0$ and $\mu = \pi$ a penalty term is added on the vertical lift ratio.

$$\begin{aligned} J(x,u) &= w_1 v_f^2 \\ &\quad + w_2 \int_{E_0}^{E_f} w_e \left[w_3 (\cos\mu)^2 + w_4 (\dot{\mu})^2 \right] \,\mathrm{d}E \end{aligned} \tag{5}$$

The variables w_1, w_2, w_3, w_4 are positive constant weighting factors and w_e is a positive function

$$w_e(E) = \frac{1}{vD} \tag{6}$$

that decreases the cost when the absolute value of the energy time rate (1b) is high. This leads the system to utilize more lift in regions where the energy dynamics are fast. This decreases the sensitivity of the control against most perturbations and thus results into an increase of the controllable region.

4.1.2. *Transcription*

The transcription uses well known methods and will not be addressed here formally. It is based on the multiple shooting algorithm with a 4th order Runge-Kutta discretization scheme. A grid adaption strategy is used to find a grid that equally distributes the local discretization error with the goal of keeping the global discretization error low. The result of the transcription is the discretized OCP in standard NLP form (7). The optimization variables z, $\dim(z) = N$ are the control variables at every grid point and the state variables at the shooting nodes. The path constraints are discretized at every grid node. Together with the initial and final conditions and the defect constraints introduced by the multiple shooting algorithm the joint constraint function $G : \mathbb{R}^N \rightarrow \mathbb{R}^M$ is formed. The objective is a function $F : \mathbb{R}^N \rightarrow \mathbb{R}$ which corresponds to (5) where the integral is approximated using the trapezoidal quadrature.

$$\min_{z} \qquad F(z) \tag{7a}$$

$$w.r.t. \quad g_l \leq G(z) \leq g_u \tag{7b}$$

The terminal energy point is free (note that the terminal speed is only bounded above). During the transcription the independent variable is normalized and the terminal energy is included as an additional optimization variable.

Let the solution exist on a grid T with length l of the normalized energy $\tau \in [0,\ 1]$.

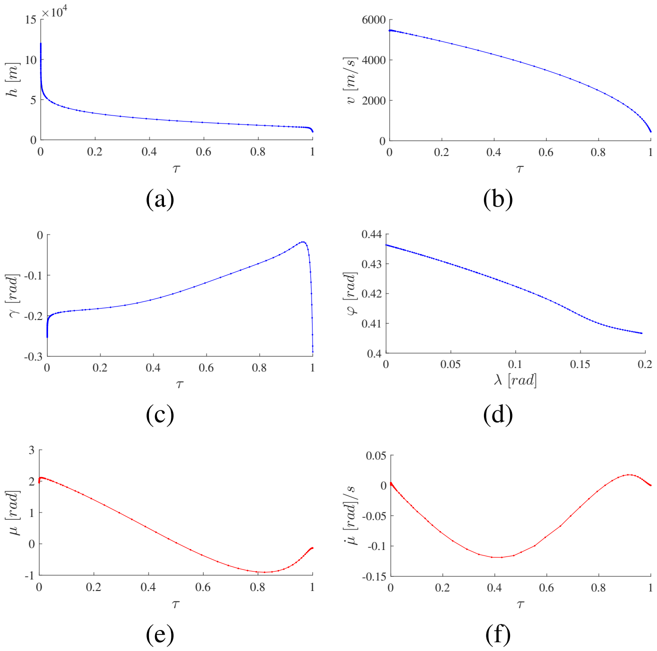
$$T = \{\tau_i \,|1 \leq i \leq l, \tau_1 = 0, \tau_l = 1, \tau_i < \tau_{i+1}\} \tag{8}$$

The notation $x_{(i,j)}$ refers to state j, $1 \leq j \leq 6$, at grid point τ_i, $1 \leq i \leq L$. In the following the state is assumed to be known at all grid points, either from being a multiple shooting node or from integration of the control function.

4.1.3. *Nominal Solution*

Let $\overset{\star}{z}$ denote the nominal optimal solution. The optimal bank angle profile $\overset{\star}{\mu}(\tau)$ and the corresponding state trajectories are shown in figure 2. Note that a large portion of the altitude decrease happens in the high energy region. Initially the energy loss is marginal $\dot{E} \approx 0$, $\dot{E} < 0$, the velocity slightly increases as the altitude rapidly decreases. This stiff characteristic can be problematic for integration, requiring a fine solution grid.

In all investigated problem variants ample margins were left before the path constraints would become active. During the online computation the trajectory is checked for path constrained violation. A violation is detected in advance, but a separate guidance mode is required to ensure that the violation is prevented.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 2. Nominal optimal solution of the altitude h (a), velocity v (b), flight path angle γ (c), ground track (d), bank angle μ (e) and bank angle derivative $\dot{\mu}$ (f).

4.1.4. Parametric Sensitivity Analysis

The optimal solution $\overset{\star}{z}$ is seen as function of a parameter set p. With sufficiently continuous functions F, G the differentiability of the solution $\overset{\star}{z}(p)$ with respect to p can be achieved locally around a chosen set of nominal parameters p_0 [1][2]. The problem is defined as parametric nonlinear program NLP(p) (9). The parameters p may nonlinearly affect the objective function and the constraint function.

$$\min_{z} \quad F(z,p) \tag{9a}$$

$$w.r.t. \quad g_l \leq G(z,p) \leq g_u \tag{9b}$$

In the entry problem the vector p, $\dim(p) = 9$ includes a perturbation of the initial state, the vehicle mass, and the lift- and drag-acceleration. The parameterized initial conditions are

$$h_0 + p_1 - x_{(1,1)} = 0 \tag{10a}$$

$$\lambda_0 + p_2 - x_{(1,2)} = 0 \tag{10b}$$

$$\varphi_0 + p_3 - x_{(1,3)} = 0 \tag{10c}$$

$$v_0 + p_4 - x_{(1,4)} = 0 \tag{10d}$$

$$\gamma_0 + p_5 - x_{(1,5)} = 0 \tag{10e}$$

$$\chi_0 + p_6 - x_{(1,6)} = 0. \tag{10f}$$

The perturbed model parameters are

$$\tilde{m} = m + p_7 \tag{11a}$$

$$\tilde{L} = \frac{1}{2}\rho(h)v^2 c_L(M)\frac{s}{\tilde{m}}\,(1 + p_8) \tag{11b}$$

$$\tilde{D} = \frac{1}{2}\rho(h)v^2 c_D(M)\frac{s}{\tilde{m}}\,(1 + p_9) \tag{11c}$$

The solution for the nominal parameter set

$$p_0 = (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9) = 0 \tag{12}$$

is denoted $z_0 = \overset{\star}{z}(p_0)$.

To obtain the nominal optimal solution z_0 the problem is solved for p_0 using an NLP solver. Once z_0 is known and if strong second order sufficient conditions (SSOSC) [3] hold, a parametric sensitivity analysis can be performed as follows: The Lagrangian function of (9) is defined as

$$L(z,\eta,p) = F(z,p) + \eta^{\mathsf{T}} G(z,p) \tag{13}$$

with the lagrangian multipliers $\eta \in \mathbb{R}^M$. Consider the set of constraints G^a that is active at the solution and the associated Lagrangian multipliers η^a.

$$G^a := \{G_i \mid G_i(z_0, p_0) = 0, i = 1, ..., M\} \tag{14}$$

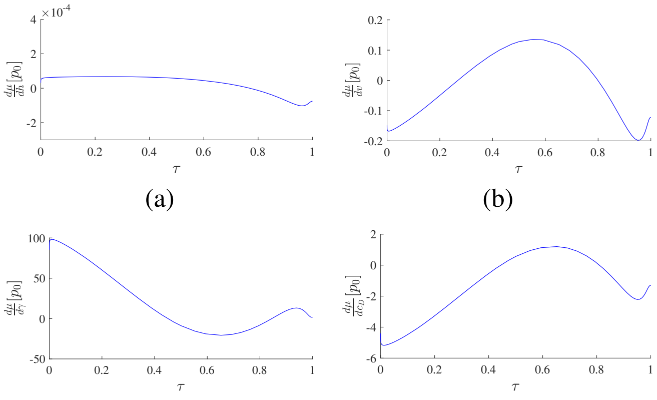
If z_0, η_0 is regular an explicit formulation of the sensitivity differentials of the primal- and dual-variables can be given

$$\begin{pmatrix} \frac{dz}{dp}[p_0] \\ \frac{d\eta^a}{dp}[p_0] \end{pmatrix} = -\nabla_{(z,\eta^a)} K^{-1} \begin{pmatrix} \nabla^2_{zp} L[z_0, \eta^a_0, p_0] \\ \nabla_p G^a[z_0, p_0] \end{pmatrix} \tag{15}$$

where

$$\nabla_{(z,\eta^a)} K = \begin{pmatrix} \nabla^2_z L[z_0, \eta^a_0, p_0] & (\nabla_z G^a[z_0, p_0])^{\mathsf{T}} \\ \nabla_z G^a[z_0, p_0] & 0 \end{pmatrix} \tag{16}$$

Under SSOSC the Hessian $\nabla^2_z L[z_0, \eta^a_0, p_0]$ evaluated at the solution is guaranteed to be positive definite and thus the right hand side of (15) is computable. The differentials (15) are called parametric sensitivities of the solution. They are first order total derivatives of the optimal solution with respect to the defined parameters p in a neighborhood around p_0. Note that the parametric sensitivities take feasibility and optimality into account. Figure 3 shows examples of the sensitivity of the bank angle against different perturbations.



(a) (b)

(c) (d)

Fig. 3. Sensitivities of the bank angle μ against a perturbation in initial altitude h (a), velocity v (b), flight path angle γ (c) and constant multiplicative error in the drag acceleration (d).

4.2. Online Phase

4.2.1. Perturbed Solution Approximation

Given the knowledge of the nominal solution z_0 and the parametric sensitivities (15), a linear approximation $\tilde{z}_1(p)$ of the perturbed solution $z(p)$ for a set of disturbed parameters p in a neighborhood $U(p_0)$ around p_0 can be obtained using a Taylor expansion of first order.

$$\tilde{z}_1(p) := z_0 + \frac{dz}{dp}[p_0]\,(p - p_0) \approx z(p) \tag{17}$$

The approximation (17) causes an error in the active constraints which can be computed exactly as

$$G^a(\tilde{z}_1, p) = \epsilon_1 \neq 0 \tag{18}$$

Now consider linear perturbations $q \in \mathbb{R}^M$ in the constraint function of (9)

$$g_l \leq G(z,p) \; - \; q \leq g_u. \tag{19}$$

The linear constraint perturbations q are a specialization of the general perturbations p. Obviously q is also of the same type as the resulting constraint error ϵ_1. Assume that when performing the offline sensitivity analysis (15) in addition to the sensitivities with respect to p also the sensitivities with respect to q are obtained for $q_0 = 0$ in an analog way. The approximate solution $\tilde{z}_1(p)$ can be significantly improved [5] by taking into account the error that the initial Taylor expansion (17) causes in the constraint function. A better approximation than (17) can be obtained by

$$\tilde{z}_{k+1}(p) = \tilde{z}_k(p) + \frac{dz}{dq^a}[q_0]\; G^a(\tilde{z}_k, p) \tag{20}$$

starting with $k = 1$. Policy (20) defines an iterative procedure. It has been shown [5] that if functions F and G of the underlying NLP are three times continuously differentiable with respect to their arguments and $p = p_0 + \Delta p \in U(p_0)$ the iteration (20) converges linearly against a fixpoint $\tilde{z}_\infty(p)$. At the fixpoint the active constraints are fulfilled exactly.

$$\|G^a(\tilde{z}_\infty(p), p)\| = 0 \tag{21}$$

Already after the first iteration of (20) the error in the objective function reduces by one order to $\mathcal{O}(\|\Delta p\|^3)$.

The approximation $\tilde{z}(p)$ includes an adapted control sequence and, depending on the transcription, a state trajectory for the entire problem. After termination of the iteration scheme the drag profile $D_{ref}(\tilde{z}(p))$ for the adapted trajectory is computed. It is input to the drag tracking controller along with the corresponding adapted bank angle profile $\mu_{ref} \in \tilde{z}(p)$.

4.2.2. *Repeated Trajectory Computation during Flight*

The parametric sensitivities obtained through (15) are a total solution derivative valid for a specific initial condition. If the trajectory computation is to be repeated at different initial conditions, a corresponding set of sensitivities is required. A possible approach to is to repeat the offline sensitivity analysis at several discrete points of the nominal trajectory, each yielding a set of parametric sensitivities.

Let $z_0[\tau_i, p_0]$, $1 \le i < l$ denote the nominal optimal solution on the subinterval $[\tau_i, 1]$. The solution on the interval $[\tau_i, 1]$ is the solution of the same optimal control process, but for the initial condition $x_{0_i} = \overset{\star}{x}_{(i,:)} \subset z_0[0, p_0]$. The cost function F is separable and the *Principle of Optimality* holds, hence $z_0[\tau_{i+1}, p_0] \subset z_0[\tau_i, p_0]$.

The parametric sensitivity differentials computed for the optimal solution on each subinterval $[\tau_i, 1]$ are denoted by

$$\frac{\mathrm{d}z}{\mathrm{d}p}[\tau_i, p_0] \tag{22}$$

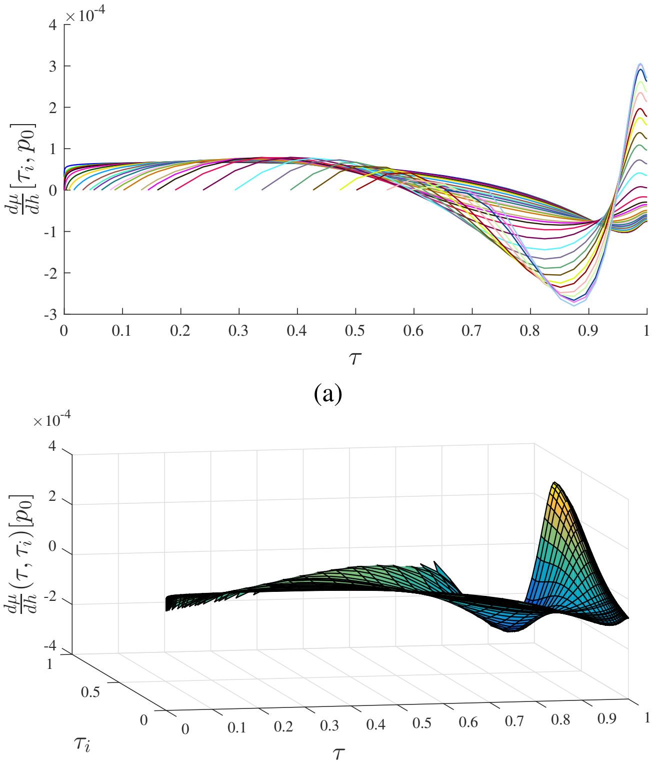
where τ_i refers to the changing initial condition. Figure 4.a shows the sensitivities of the bank angle μ against perturbations in altitude at different initial conditions τ_i, $1 \le i \le k$.

$$\frac{\mathrm{d}\mu}{\mathrm{d}h}[\tau_1, p_0],\; \ldots,\; \frac{\mathrm{d}\mu}{\mathrm{d}h}[\tau_k, p_0] \tag{23}$$

Observe that

$$\frac{\mathrm{d}\mu}{\mathrm{d}h}[\tau_{i+1}, p_0] \not\subseteq \frac{\mathrm{d}\mu}{\mathrm{d}h}[\tau_i, p_0] \tag{24}$$

and that the sensitivity increases as $\tau_i \to 1$. A perturbation of constant strength requires a stronger change of the control function to be compensated on a shortening process interval.



(a)

(b)

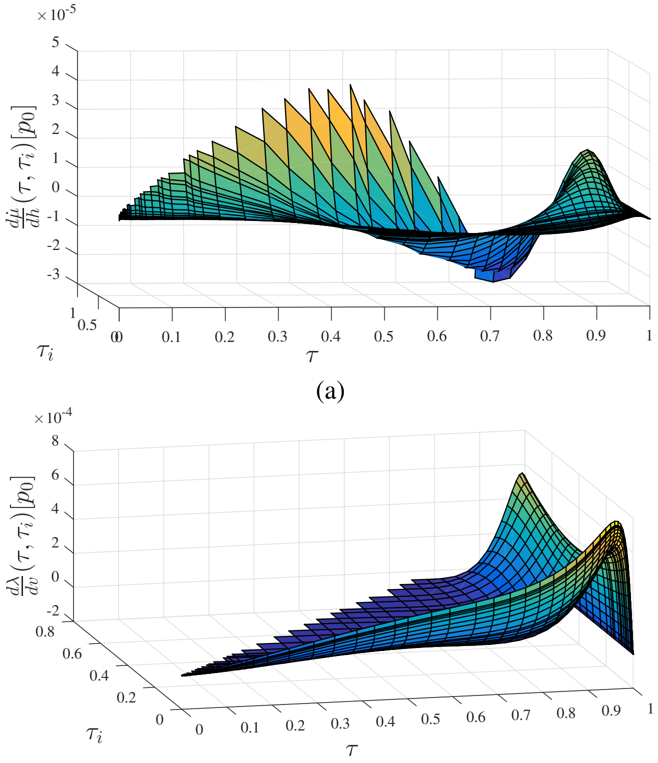
Fig. 4. Sensitivity of the bank angle μ against perturbations in altitude h at τ_i, $1 \le i \le k < l$ along the nominal trajectory. Figure (a) shows the data basis $\frac{\mathrm{d}\mu}{\mathrm{d}h}[\tau_i, p_0]$ for the interpolation $\frac{\mathrm{d}\mu}{\mathrm{d}h}(\tau, \tau_i)[p_0]$ shown in (b).

If the parametric sensitivities sets are computed on a dense grid of initial conditions occurring along the nominal trajectory it is possible to interpolate between them based on the continuity assumptions on F and G and the underlying optimal control process. The obtained surface (25) (figure 4.b) is a function of the initial condition at τ_i and the independent variable τ with $\tau_i \le \tau \le 1$.

$$\frac{\mathrm{d}\mu}{\mathrm{d}h}(\tau, \tau_i)[p_0] \tag{25}$$

At normalized energy $\bar{\tau}$ the surface can be evaluated at $\tau_i = \bar{\tau}$ to obtain the approximate sensitivity for the bank angle on the remaining interval $[\bar{\tau}, 1]$. The surface interpolation has to be performed separately for each state, control channel and perturbation. Figure 5 shows additional examples.

The interpolated sensitivities are used as input for the real-time iteration scheme. A more detailed description of this strategy will appear soon.



(a)

(b)

Fig. 5. (a) Sensitivity of the bank angle rate $\dot{\mu}$ against perturbations in altitude h. (b) Sensitivity of the longitude λ against perturbations in velocity v along the nominal trajectory.

5. TRAJECTORY TRACKING

Drag tracking is heritage from the space shuttle era. It has proven to be very robust against perturbations in atmospheric density, which is a major concern for Mars entry. The limited availability of measurement data is the driving design factor. The drag acceleration can be related to the accelerations measured by an inertial measurement unit (IMU) hence the control concept is based on a direct measurement, which is a unique advantage as other state information must be assumed to be integrated.

The used tracking law is due to Mease et. al and related work [6][7]. The tracking controller is based on exact input-output linearization of the drag dynamics in the energy domain as feedforward and a PID controller as stabilizing feedback. The concept is briefly summarized, e.g. Isidori [11] provides a comprehensive introduction into feedback linearization techniques.

By neglecting the planet rotation and substituting the vertical lift over drag ratio (vLoD) as the control variable

$$u = \frac{L}{D}\cos\mu \tag{26}$$

the longitudinal entry dynamics can be written as a control affine system, with drag as the output variable. In the energy domain this system is of full relative degree and the dynamics can be transformed into a new set of coordinates using a diffeomorphism. In the new coordinates the second energy derivative of drag can be written as

$$D'' = a + bu \tag{27}$$

where a and b are terms depending on D, D', E. The linearizing control is then given by

$$u_{drag} = \frac{1}{b}(-a + \nu) \tag{28}$$

where ν is the outer loop PID control. The commanded vLoD is converted to the bank angle magnitude while the sign is determined by a corridor on the heading error or it can be obtained from the trajectory solution directly depending on the frequency of the trajectory updates. The tracking controller takes into account the estimated perturbations p_7, p_8, p_9 when calculating (28).

Tests have shown that the controller is very sensitive to early drag error. To guard against overreaction in the thin layers of the atmosphere the drag based command u_{drag} is weighted against the reference command u_{ref} obtained from the trajectory generation on a schedule of sensed drag acceleration using a sigmoid function S which is zero at zero drag and one at 60% of the expected peak drag acceleration and after.

$$u_{com} = u_{ref} + S(D)(u_{drag} - u_{ref}) \tag{29}$$

This way the guidance system relies more on the trajectory generation in the beginning and smoothly transitions to the drag control law. The scheduling also increases the probability that the commanded bank angle can be actually realized during the high drag segment of the flight. The bank angle is then obtained by

$$|\mu_{com}| = \arccos\left(\frac{D}{L}u_{com}\right). \tag{30}$$

6. PERFORMANCE ANALYSIS

6.1. Monte Carlo Simulation

The proposed guidance system was tested in a 3.5 degree of freedom simulation, using a pseudo attitude control for the bank angle rotation. The atmosphere data is obtained from the European Mars Climate Database [12]. The nominal trajectory assumes average UV conditions and average dust distribution. Considered are errors in the atmospheric density, aerodynamic coefficients, wind speed, mass and the initial state. The atmosphere properties are randomized between cold and warm scenarios and in addition random sinusoidal perturbations with amplitudes of up to 50% of the nominal value are added. Table 3 shows the intervals and maximum amplitudes of the introduced perturbations. The intervals around λ_0 and φ_0 roughly correspond to a maximum error of 20 km in downrange and 10 km in crossrange direction at the EIP. No error covariance has been assumed, the errors are equally distributed.

Table 3. Perturbed Environment

State		Value	Parameter	max. Amp.
h_0	+/-	3000 m	ρ	temp. + 50%
λ_0	+/-	0.3265°	c_L	10%
φ_0	+/-	0.1632°	c_D	10%
v_0	+/-	200 m/s	wind speed	200 m/s
γ_0	+/-	1°	mass (const.)	up to 20 kg
χ_0	+/-	1°		

The guidance inputs are the full state and the lift- and drag acceleration. The inputs are the true values falsified with white noise. An extended Kalman Filter is used to estimate the p-perturbation in the lift- and drag-acceleration. The perturbed mass was assumed to be known.

The trajectory computation is executed at 0.05 Hz, the tracking is performed at 20 Hz. The most recently computed satisfactory trajectory is tracked. If the corrector iteration (20) is not convergent the trajectory is discarded.

Table 4 lists the achieved mean errors and the standard deviations based on 2500 Monte Carlo cases. The guidance has a slight tendency to undershoot, the $(\mu + 3\sigma)$ horizontal miss distance is 12.3 km. Figure 6 shows the dispersion at parachute opening.

Table 4. Terminal Dispersion

Type	Mean (μ)	Std. Deviation (σ)
Euclidean Dist.	4132 m	2678 m
Horizontal Dist.	3999 m	2759 m
Downrange Error	-2636 m	3895 m
Crossrange Error	-435 m	1138 m
Altitude Error	-408 m	690 m
Velocity Error	4 m/s	6 m/s

6.2. Processor in the Loop

The guidance was run on a 80 MHz LEON2 processor board. Basis for the test was a discretization with a grid length of $l = 70$ using a dense NLP formulation. The GNC C-code has been obtained via autocoding from the development implementation. Table 5 shows a comparison between the required computation time on a standard Linux desktop PC and the LEON2 running RTEMS. The binaries for both systems have been compiled from the same source code. The binary size of the sensitivity catalog was roughly 25 MB.

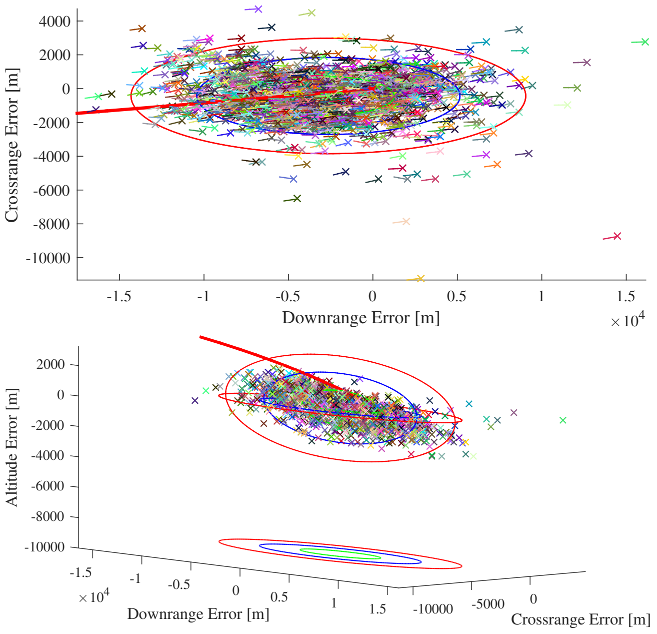


Fig. 6. Dispersion ellipse at parachute opening.

Table 5. Runtime Comparison

Computation	i7, 2.9 GHz	LEON2, 80 MHz	Factor
Sen. interp.	12 ms	324 ms	≈ 27
p-step (17)	0.05 ms	3 ms	≈ 60
q-step (20)	0.19 ms	8 ms	≈ 42
$G(z, p)$	0.88 ms	32 ms	≈ 36

The p- and q-correction steps can be computed almost instantly. The evaluation of the constraint function G takes somewhat longer. In the majority of cases less than ten corrector iterations (20) are required to reduce the constraint violation to an acceptable level. The most costly operation is the interpolation of the required sensitivity differentials. An alternative is to trigger the trajectory generation at predetermined points, using fix sets of sensitivities. But even including the interpolation, the trajectory computation can be performed in less than a second, proofing the real-time capability of the approach on flight equivalent hardware.

6.3. Numerical Treatment

The transcription and the analysis of the nominal trajectory and the sensitivity differentials is performed by the code *Sen-*

sitivity Analysis Framework (SAF) by D. Seelbinder. The solution of the parametric NLP and the computation of the sensitivity differentials is performed by the NLP solver WORHP [13].

The numeric quality of the sensitivity differentials strongly depends on the knowledge of the true sparsity structure of the Hessian matrices in (15). Standard finite difference routines do not provide sufficient accuracy. The required derivatives are computed using the automatic differentiation library ADOL-C [14].

The online part of the guidance algorithm is implemented in Embedded Matlab. Automatic code generation is used to obtain the guidance algorithm in C-code. For the processor in the loop test the tool chain TASTE [15] is used to specify the interfaces of the partition between the GNC algorithm and the simulation environment. The setup of the communication architecture and the final compilation for the LEON2 processor is also performed by TASTE, using the generated C-code.

7. DISCUSSION

The p-correction step (17) is a direct feedback [16][17] on the parameters p, requiring p to be known at computation time. As a consequence there are no perturbation parameters explicitly for the atmospheric density model $\rho(h)$. It has been found problematic to meaningfully parametrize the atmospheric disturbances and robustly distinguish between perturbations of the aerodynamic coefficients and the inner functions of the density model, using only parameters that can be estimated at flight time.

Unforeseeable or unknown perturbations are countered indirectly by computing the constraint violation and reducing it via the q-correction step (20). Relying only on this strategy, without a perturbation model and corresponding p-step (17), results into a lower order solution approximation [5]. Although not strictly necessary, an accurate perturbation estimation combined with a meaningful perturbation model improve the solution approximation with respect to optimality.

The sensitivity differentials (15) are based on a linearization [3] of the Karush-Kuhn-Tucker conditions around the nominal value p_0. The linearization is valid in a neighborhood $U(\tau, p_0)$ in which the real-time iteration scheme (17)(20) converges to a fixpoint. For nonlinear problems the extend of $U(\tau, p_0)$ can only be approximated numerically. Maximally $U(\tau, p_0)$ extends until a change of the active set of constraints is required, but depending on the nonlinearity of the system it can be smaller. It is a major task to formulate the optimization problem such that the optimal solution is insensitive against perturbations and that the margin between the nominal solution and inactive inequality constraints is large. Possible extensions of the theory are discussed in [5].

Loosely speaking, most expected disturbed EIP conditions lie within $U(0, p_0)$. The sensitivity of the control against perturbations at the EIP $\frac{\mathrm{d}\mu}{\mathrm{d}p}[0, p_0]$ is small. Conversely small changes in the early control function have a strong effect on the terminal state. For that reason the method is well suited to adapt to state errors at the EIP or to compensate for static parameter changes, like the mass. As the system progresses $\tau \to 1$ the neighborhood $U(\tau, p_0)$ shrinks. This is dramatically noticeable after the peak deceleration and when the flight path angle starts to decrease. At this point strong perturbations cannot be compensated anymore. It is an interesting future task to further investigate the resulting controllability tube and possibly compare it with the backwards reachable set.

The iteration scheme does not rely on the online computation of derivatives and therefore requires less computational power than methods which perform online optimization. If single shooting is used as transcription method the integration of the trajectory performed in $G(z, p)$ in (20) directly yields knowledge of path- and terminal constraint satisfaction. A dense problem formulation is also advantageous for minimizing the amount of optimization variables and required sensitivity differentials. In the case of multiple shooting the evaluation of $G(z, p)$ yields the defects at the shooting nodes. A comparison between the convergence regions for single-shooting and for full discretization is a future task.

8. CONCLUSION

For nonlinear dynamic processes that are too fast to allow for the online solution of NLPs, or when computational resources are at a premium, parametric sensitivity analysis combined with the real-time iteration scheme (17)(20) is a promising candidate to approximate feasible, near optimal solutions at very low computational cost. The iteration scheme is a versatile tool in that it can be applied to arbitrary NLPs for which strong second-order sufficient conditions hold at the nominal solution. The practical limit of the approach is defined by the extend of the region of convergence $U(\tau, p_0)$, which is depended on the solution structure.

The Mars entry study demonstrates that the trajectory generation is capable of adapting to strong state and parameter disturbances at the EIP. The combination with drag tracking provides the required robustness against the unforeseeable changes of the atmosphere that are difficult to handle using predictive methods. Therefore the two approaches complement each other well. Processor in the loop testing has proven that the proposed method is real-time capable on flight equivalent hardware.

9. ACKNOWLEDGMENT

This research was conducted under No. 4000107257/12 NL/GLC/al of the ESA Networking Partnering Initiative.

10. REFERENCES

[1] Maurer, Helmut and Hans Josef Pesch: *Solution differentiability for nonlinear parametric control problems*. SIAM Journal on Control and Optimization, 32(6):1542–1554, 1994.

[2] Maurer, H. and H.J. Pesch: *Solution differentiability for parametric nonlinear control problems with control-state constraints*. Journal of Optimization Theory and Applications, 86(2):285–309, 1995, ISSN 0022-3239.

[3] Büskens, Christof and Helmut Maurer: *Sqp-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control*. Journal of Computational and Applied Mathematics, 120:85 – 108, 2000, ISSN 0377-0427.

[4] Büskens, Christof: *Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustands-Beschränkungen*. Dissertation, Westfälische Wilhelms-Universität Münster, 1998.

[5] Büskens, Christof: *Echtzeitoptimierung und Echtzeitoptimalsteuerung parametergestörter Probleme*. Habil. Thesis, 2002.

[6] Mease, Kenneth D. and Jean Paul Kremer: *Shuttle entry guidance revisited*. AIAA Guidance, Navigation and Control Conference, August 1992.

[7] Mease, Kenneth D. and Jean Paul Kremer: *Shuttle entry guidance revisited using nonlinear geometric methods*. Journal of Guidance, Control, and Dynamics, 17(6):1350–1356, 1994.

[8] Bharadwaj, Sanjay, Anil V. Rao, and Kenneth D. Mease: *Entry trajectory tracking law via feedback linearization*. Journal of Guidance, Control, and Dynamics, 21(5), 1998.

[9] Tu, Kuang Yang, Mohammed S. Munir, Kenneth D. Mease, and David S. Bayard: *Drag-based predictive tracking guidance for mars precision landing*. Journal of Guidance, 23(4), 2000.

[10] Saraf, A., J. A. Leavitt, D. T. Chen, and K. D. Mease: *Design and evaluation of an acceleration guidance algorithm for entry*. Journal of Spacecraft and Rockets, 41(6), 2004.

[11] Isidori, Alberto: *Nonlinear Control Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 1995.

[12] Lewis, S. R., M. Collins, P. L. Read, F. Forget, F. Hourdin, R. Fournier, C. Hourdin, O. Talagrand, and J. P. Huot: *A climate database for mars*. Journal of Geophysical Research, 1999.

[13] Wassel, D. and C. Büskens: *Modeling and Optimization in Space Engineering*, volume 73 of *Springer Optimization and Its Applications*, chapter The ESA NLP Solver WORHP. Springer Verlag, 2013.

[14] Walther, A. and A. Griewank: *Getting started with adol-c*. In Naumann, U. and O. Schenk (editors): *Combinatorial Scientific Computing*, chapter 7, pages 181–202. Chapman-Hall CRC Computational Science, 2012.

[15] Perrotin, Maxime, Eric Conquet, Pierre Dissaux, Thanassis Tsiodras, and Jerome Hugues: *The taste toolset: turning human designed heterogeneous systems into computer built homogeneous software*. European Congress on Embedded Real-Time Software (ERTS 2010), 2010.

[16] Pesch, Hans Josef: *Real-time computation of feedback controls for constrained optimal control problems. part 1: Neighbouring extremals*. Optimal Control Applications and Methods, 10(2):129–145, 1989.

[17] Pesch, Hans Josef: *Real-time computation of feedback controls for constrained optimal control problems. part 2: A correction method based on multiple shooting*. Optimal Control Applications and Methods, 10(2):147–171, 1989.