Real-Time Atmospheric Entry Trajectory Computation Using Parametric Sensitivities

6th International Conference on Astrodynamics Tools and Techniques – ID 121

David Seelbinder
German Aerospace Center
Institute of Space Systems
✉ david.seelbinder@dlr.de

Prof. Christof Büskens
University Bremen
Center for Technomathematics
✉ bueskens@math.uni-bremen.de

Knowledge for Tomorrow
Guided Entry Phase

*Schiaparelli enters atmosphere*
- Time: 0 sec
- Altitude: 123 km
- Speed: 21,000 km/h

*Heatshield protection during atmospheric deceleration*
- Time of maximum heating: 1 min 32 sec
- Altitude: 45 km
- Speed: 19,000 km/h

*Parachute deploys*
- Time: 3 min 21 sec
- Altitude: 11 km
- Speed: 1,700 km/h

*Front shield separates, radar turns on*
- Time: 4 min 2 sec
- Altitude: 7 km
- Speed: 320 km/h

*Parachute jettisoned with rear cover*
- Time: 5 min 22 sec
- Altitude: 1.2 km
- Speed: 240 km/h

*Thruster ignition*
- Time: 5 min 23 sec
- Altitude: 1.1 km
- Speed: 250 km/h

*Thrusters off, freefall*
- Time: 5 min 52 sec
- Altitude: 2 m
- Speed: 4 km/h

*Touchdown*
- Time: 5 min 53 sec
- Altitude: 0 m
- Speed: 10 km/h

Image credit: ESA
Guided Entry Phase

- **Schiaparelli enters atmosphere**
  - Time: 0 sec
  - Altitude: 156 km
  - Speed: 21,000 km/h

- **Heatshield protection during atmospheric deceleration**
  - Time of maximum deceleration: 3 min 12 sec
  - Altitude: 45 km
  - Speed: 19,000 km/h

- **Parachute deploys**
  - Time: 3 min 2 sec
  - Altitude: 11 km
  - Speed: 1,700 km/h

- **Front shield separates, radar turns on**
  - Time: 4 min 2 sec
  - Altitude: 7 km
  - Speed: 320 km/h

- **Parachute jettisoned with rear cover**
  - Time: 5 min 22 sec
  - Altitude: 1.2 km
  - Speed: 140 km/h

- **Thruster ignition**
  - Time: 5 min 23 sec
  - Altitude: 1.1 km
  - Speed: 250 km/h

- **Thrusters off; freefall**
  - Time: 5 min 52 sec
  - Altitude: 2 m
  - Speed: 4 km/h

- **Touchdown**
  - Time: 5 min 53 sec
  - Altitude: 0 m
  - Speed: 0 km/h

Image credit: ESA
Agenda

- Trajectory computation
  - Offline sensitivity analysis
  - Online solution approximation
- Two-Degree of Freedom Guidance System
- Results of Monte Carlo Campaign
- Results of Processor-in-the-loop test
- Summary
Offline Process: Problem Formulation and Transcription

Formulation as parametric Optimal Control Problem:

\[
\begin{align*}
\text{OCP}(p) & = \min_{z} \quad J(x, u, p) = g(x(t_0), x(t_f), p) + \int_{t_0}^{t_f} l(x(t), u(t), p) \, dt \\
\text{w.r.t.} & \quad \dot{x}(t) = f(x(t), u(t), p) \\
& \quad \psi_0(x(t_0), p) = 0 \\
& \quad \psi_f(x(t_f), p) = 0 \\
& \quad C(x(t), u(t), p) \leq 0 \\
& \quad t \in [t_0, t_f]
\end{align*}
\]
Offline Process: Problem Formulation and Transcription

- Direct optimization
  - Discretization (Runge-Kutta-4, Linear control interpolation)
  - Transcription into a parametric Nonlinear Program (shooting technique)

\[
\begin{align*}
\min_z & \quad F(z, p_0) \\
\text{w.r.t.} & \quad g_l \leq G(z, p_0) \leq g_u
\end{align*}
\]

- Choice of a nominal parameter set \( p_0 \)
Obtain nominal optimal solution $z_0 = z^* (p_0)$
Let $z_0$ fulfill strong second order sufficient conditions
Karush-Kuhn-Tucker-matrix is invertible
Sensitivity differentials (SD) are computable

\[
\begin{bmatrix}
\frac{dz}{dp} [p_0] \\
\frac{d\eta^a}{dp} [p_0]
\end{bmatrix} = - \begin{bmatrix}
\nabla_z^2 L(z_0, \eta_0^a, p_0) & \nabla_z G^a (z_0, p_0)^T \\
\nabla_z G^a (z_0, p_0) & 0
\end{bmatrix}^{-1} \begin{bmatrix}
\nabla_{zp}^2 L(z_0, \eta_0^a, p_0) \\
\nabla_p G^a (z_0, p_0)
\end{bmatrix}
\]
Online Solution Approximation: p-Step

- Parameter vector \( p \)
  - Perturbation of the initial condition \( \psi_0 \)
  - Perturbation of model parameters: mass, aerodynamic coefficients

- Approximation of optimal solution for disturbed parameters \( p \) using Taylor expansion

\[
z^*(p) \approx z_1 := z_0 + \frac{dz}{dp} \left[ p_0 \right] \cdot (p - p_0)
\]
Online Solution Approximation: p-Step

- Parameter vector $p$
  - Perturbation of the initial condition $\psi_0$
  - Perturbation of model parameters: mass, aerodynamic coefficients

- Approximation of optimal solution for disturbed parameters $p$ using Taylor expansion

\[ z^*(p) \approx z_1 := z_0 + \frac{dz}{dp}[p_0] \cdot (p - p_0) \quad \Rightarrow \quad \text{error in the active constraints} \quad \| G^a(z_i, p) \| > 0 \]
Offline Process: Sensitivity Analysis II

\[ \min_z \quad F(z, p_0) \]

w.r.t.

\[ g_l \leq G(z, p_0) \leq g_u \]
Offline Process: Sensitivity Analysis II

Additional parameter vector \( q \) with nominal value \( q_0 = 0 \)

\[
\begin{align*}
\min_z & \quad F(z, p_0) \\
\text{w.r.t.} & \quad g_l \leq G(z, p_0) - q_0 \leq g_u
\end{align*}
\]

\( \frac{dz}{dq} \) can be computed analog to \( \frac{dz}{dp} \)
Online Solution Approximation: p-Step and q-Step

- Parameter vector $p$
  - Perturbation of the initial condition $\psi_0$
  - Perturbation of model parameters: mass, aerodynamic coefficients

- Approximation of optimal solution for disturbed parameters $p$ using Taylor expansion

$$z^*(p) \approx z_1 := z_0 + \frac{dz}{dp}[p_0] \cdot (p - p_0)$$

error in the active constraints $\|G^a(z_i, p)\| > 0$
Online Solution Approximation: p-Step and q-Step

- Parameter vector $p$
  - Perturbation of the initial condition $\psi_0$
  - Perturbation of model parameters: mass, aerodynamic coefficients

- Approximation of optimal solution for disturbed parameters $p$ using Taylor expansion

$$z^*(p) \approx z_1 := z_0 + \frac{dz}{dp} [p_0] \cdot (p - p_0)$$

- $\frac{dz}{dq}$ is used to iteratively correct the constraint error and at the same time improve the order of optimality of the approximation

\[\text{while } \|G^a(z_i, p)\| > \varepsilon\]

$$q_i = G^a(z_i, p)$$

For $\|q - q_0\| < \delta$ iteration converges against a fixpoint $z_\infty$ at which

$$\|G^a(z_\infty, p)\| = 0$$
For trajectory computation at time $t$ the sensitivity differentials must be known for the initial condition $\psi_0^t := x^*(t), t \in [t_0, t_f)$
Offline Process: Build Sensitivity Catalog

For trajectory computation at time $t$ the sensitivity differentials must be known for the initial condition $\psi^t_0 := x^*(t)$, $t \in [t_0, t_f)$

Repeat sensitivity analysis at discrete points $t_i \in [t_0, t_f)$, $0 < i \leq l$ of the nominal trajectory $x^*(t)$
Sensitivity on Discrete Points of the Nominal Trajectory

- Example: Sensitivity of $\mu$ against perturbations in $h$ at $x^*(t_i), 0 \leq i \leq k$
Sensitivity on Discrete Points of the Nominal Trajectory

- Example: Sensitivity of $\mu$ against perturbations in $h$ at $x^*(t_i)$, $0 \leq i \leq k$

- Interpolation between SD is feasible based on the continuity of the OCP

- Sensitivity surfaces

- Online: At time $\bar{t}$ surface is evaluated at $(t, \bar{t})$, $\bar{t} \leq t \leq t_f$

$$\frac{d\mu}{dh}(t, \bar{t}) \approx \frac{d\mu}{dh_{\bar{t}}}(t)$$

to obtain an approximation of the SD against a perturbation affecting the system at the instant $\bar{t}$
Guidance System Overview

- Two-degree-of-freedom design
  - Fast inner tracking loop (20 Hz)
  - Slow outer trajectory loop (0.05 Hz)

- Trajectory computation outputs
  - near optimal discrete \( u^*, x^* \) for the entire remaining process
  - optimal bank angle profile \( \mu_{ref} \) and drag profile \( D_{ref} \) obtained from \( u^*, x^* \)

- Drag tracking controller based on Mease et. al.
Monte Carlo: Perturbed Environment

- Comparatively large EIP state errors (using a uniform error distribution)
- Atmospheric perturbations:
  - Random temperature profile between warm and cold conditions
  - Random sinusoidal density perturbations of up to 50% amplitude
- Aerodynamic coefficients perturbed by up to 10%
- Guidance input:
  - true state, lift and drag falsified with white noise
  - Atmospheric perturbation parameters are estimated using an extended kalman filter

State Pert. | Pert.  
---|---
$h_0$ | +/- 3 km
$\lambda_0$ | +/- 0.3265°
$\varphi_0$ | +/- 0.1632°
$v_0$ | +/- 200 m/s
$\gamma_0$ | +/- 1°
$\chi_0$ | +/- 1°

Param Pert. | Pert.  
---|---
$\rho$ | Temp. +/- 50%
$c_L$ | +/- 10 %
$c_D$ | +/- 10 %
wind | +/- 200 m/s
mass | +/- 20 kg
Monte Carlo: Results

\(|\mu| + 3\sigma|\) hori. dist.: \textbf{12.3 km}

\(|\mu| + 3\sigma|\) alt. error: \textbf{2.4 km}

(3.5 DOF, 2500 MC cases)

<table>
<thead>
<tr>
<th>Result</th>
<th>Mean ($\mu$)</th>
<th>Std. Dev. ($\sigma$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eucl. dist.</td>
<td>4.1 km</td>
<td>2.6 km</td>
</tr>
<tr>
<td>Hori. dist.</td>
<td>4 km</td>
<td>2.7 km</td>
</tr>
<tr>
<td>DR error</td>
<td>- 2.6 km</td>
<td>3.8 km</td>
</tr>
<tr>
<td>CR error</td>
<td>- 0.4 km</td>
<td>1.1 km</td>
</tr>
<tr>
<td>Alt. error</td>
<td>- 0.4 km</td>
<td>0.7 km</td>
</tr>
<tr>
<td>Vel. error</td>
<td>4 m/s</td>
<td>6 m/s</td>
</tr>
</tbody>
</table>
Processor-in-the-loop

- Test on RASTA-101 with 80 MHz LEON2 processor
- GNC c-code from autocoding from Embedded Matlab
- TASTE toolset: Onboard SW interface definition, communication setup and target compilation
- Used dense NLP formulation
  - grid length 70
  - ~25 MB sensitivity data
Processor-in-the-loop

- Test on RASTA-101 with 80 MHz LEON2 processor
- GNC c-code from autocoding from Embedded Matlab
- TASTE toolset: Onboard SW interface definition, communication setup and target compilation
- Used dense NLP formulation
  - grid length 70
  - ~25 MB sensitivity data

Trajectory computation time: < 1 sec.
Summary

- Applied sensitivity analysis to discretized optimal control process for different initial conditions
- Repeated online trajectory computation based on sensitivity interpolation, Taylor expansion and iterative constraint correction
- Two-degree-of-freedom guidance system using drag tracking
- Promising results in 3.5 DOF Monte Carlo campaign
- Real-time capability proven by PIL test on LEON2 processor
Summary

➢ Applied sensitivity analysis to discretized optimal control process for different initial conditions

➢ Repeated online trajectory computation based on sensitivity interpolation, Taylor expansion and iterative constraint correction

➢ Two-degree-of-freedom guidance system using drag tracking

➢ Promising results in 3.5 DOF Monte Carlo campaign

➢ Real-time capability proven by PIL test on LEON2 processor

Thank you for your attention!

david.seelbinder@dlr.de

This research was conducted under No. 4000107257/12 NL/GLC/al of the ESA Networking and Partnering Initiative.
Monte Carlo and PIL Results

<table>
<thead>
<tr>
<th>State</th>
<th>Pert.</th>
<th>State Pert.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_0$</td>
<td>$\pm 3$ km</td>
<td>$\pm 3$ km</td>
</tr>
<tr>
<td>$\lambda_0$</td>
<td>$\pm 0.3265^\circ$</td>
<td>$\pm 0.3265^\circ$</td>
</tr>
<tr>
<td>$\varphi_0$</td>
<td>$\pm 0.1632^\circ$</td>
<td>$\pm 0.1632^\circ$</td>
</tr>
<tr>
<td>$v_0$</td>
<td>$\pm 200$ m/s</td>
<td>$\pm 200$ m/s</td>
</tr>
<tr>
<td>$\gamma_0$</td>
<td>$\pm 1^\circ$</td>
<td>$\pm 1^\circ$</td>
</tr>
<tr>
<td>$\chi_0$</td>
<td>$\pm 1^\circ$</td>
<td>$\pm 1^\circ$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Param</th>
<th>Pert.</th>
<th>Param Pert.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>Temp. $\pm 50%$</td>
<td>Temp. $\pm 50%$</td>
</tr>
<tr>
<td>$c_L$</td>
<td>$\pm 10%$</td>
<td>$\pm 10%$</td>
</tr>
<tr>
<td>$c_D$</td>
<td>$\pm 10%$</td>
<td>$\pm 10%$</td>
</tr>
<tr>
<td>wind</td>
<td>$\pm 200$ m/s</td>
<td>$\pm 200$ m/s</td>
</tr>
<tr>
<td>mass</td>
<td>$\pm 20$ kg</td>
<td>$\pm 20$ kg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result</th>
<th>Mean ($\mu$)</th>
<th>Std. Dev. ($\sigma$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eucl. dist.</td>
<td>4.1 km</td>
<td>2.6 km</td>
</tr>
<tr>
<td>Hori. dist.</td>
<td>4 km</td>
<td>2.7 km</td>
</tr>
<tr>
<td>DR error</td>
<td>-2.6 km</td>
<td>3.8 km</td>
</tr>
<tr>
<td>CR error</td>
<td>-0.4 km</td>
<td>1.1 km</td>
</tr>
<tr>
<td>Alt. error</td>
<td>-0.4 km</td>
<td>0.7 km</td>
</tr>
<tr>
<td>Vel. error</td>
<td>4 m/s</td>
<td>6 m/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>i7, 2.9 GHz</th>
<th>LEON2, 80 MHz</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sen. interp.</td>
<td>12 ms</td>
<td>324 ms</td>
<td>~27</td>
</tr>
<tr>
<td>p-step</td>
<td>0.05 ms</td>
<td>3 ms</td>
<td>~60</td>
</tr>
<tr>
<td>q-step</td>
<td>0.19 ms</td>
<td>8 ms</td>
<td>~42</td>
</tr>
<tr>
<td>$G(z,p)$</td>
<td>0.88 ms</td>
<td>32 ms</td>
<td>~36</td>
</tr>
</tbody>
</table>
Real-Time Production Code Generation and Testing

- Solve OCP with FD Transcription
- Compute Sen-Sets\(0\ldots n\) For selected Transcription
- Data Processing Prepare Guidance Input

- GNC Algorithms Implemented in Embedded Matlab
- HF Simulator Simulink
- C-Code Generation for GNC Matlab Coder / RTW
- Real-Time-Code Generation Matlab Coder / RTW / TASTE
- RASTA Leon2 Onboard Computer (runs GNC)
- dSpace RT Plattform (runs environment)
- Monte Carlo Software in the Loop
- GNC Performance Analysis Landing Site Dispersion, etc.
- Monte Carlo Processor in the Loop
Closed Loop Guidance Design

Phase 1
State/Pert. Estimation

Estimate $p$

Estimate Independent Variable

Phase 2
Preparation

Sen-Interpolation and Problem Size Reduction

Phase 3
Computation

PSA based Trajectory Computation

Phase 4
Forward Generation

Feed Forward Command Generation

Opt. Drag Profile Generation

Phase 5
Error Estimation

Drag Error Estimation

Phase 6
Tracking

Drag Tracking Dynamic Inversion Controller

Phase 7
Command Merging

Guidance Command Generation

Guidance Cmd

Attitude Control

Navigation

Navigation
Parameter Dependent NLP Resulting From Control Discretization ("Single Shooting")

Minimize \[
\Phi(u,p) = g(x_1, x_N(u), p)
\]

subject to \[
\psi(x_1, x_N(u), p) - q^1 = 0
\]
\[
S(x_i(u), u_i, p) - q^2 \leq 0 \quad i = 1, \ldots, N
\]
Parameter Dependent NLP Resulting From Control Discretization ("Single Shooting")

Minimize \[ \Phi(u, p) = g(x_1, x_N(u), p) \]
subject to \[ \psi(x_1, x_N(u), p) - q^1 = 0 \]
\[ S(x_i(u), u_i, p) - q^2 \leq 0 \quad i = 1, \ldots, N \]

Parameter Dependent NLP Resulting From Control and State Discretization ("Multiple Shooting")

Minimize \[ \Phi(u, x, p) = g(x_1, x_N, p) \]
subject to \[ \psi(x_1, x_N, p) - q^1 = 0 \]
\[ S(x_i, u_i, p) - q^2 \leq 0 \quad i = 1, \ldots, N \]
\[ \delta_j(x_i, u_{i\ldots k}, x_{i+1}, p) - q^3 = 0 \quad j = 1, \ldots, SI \quad k = 1, \ldots, CpSI \]

Mind: Only optimizable ("free") variables should be included in the NLP...
Initial Condition Dependency of Sensitivity Differentials

\[ OCP_{\tau_i} \quad \text{and} \quad OCP_{\tau_{i+1}} \]

- OCP\(_i\) and OCP\(_{i+1}\) are closely related by the Bellman Principle
Initial Condition Dependency of Sensitivity Differentials

\[ OCP_{\tau_i} \]

\[ \begin{bmatrix}
    \nabla_z^2 L(z_0, \eta_0^a, p_0) & \nabla_z G^a(z_0, p_0) \\
    \nabla_z G^a(z_0, p_0) & 0
\end{bmatrix} \]

\[ K_{\tau_{i+1}} = \begin{bmatrix}
    \nabla_z^2 L(z_0, \eta_0^a, p_0) & \nabla_z G^a(z_0, p_0) \\
    \nabla_z G^a(z_0, p_0) & 0
\end{bmatrix} \]

- OCP\(_i\) and OCP\(_{i+1}\) are closely related by the Bellman Principle
- In the NLP representation of the OCP, gridded optimization variables become fixed up to \(\tau_{cur}\)!
- Fixed variables can be removed \(\Rightarrow\) K-Matrix is time dependent
Estimation of Sensitivity Matrix Dimensions for PSA Update

<table>
<thead>
<tr>
<th></th>
<th>SS</th>
<th>FD</th>
</tr>
</thead>
<tbody>
<tr>
<td># NLP variables: N</td>
<td>u*D</td>
<td>(u+x)*D</td>
</tr>
<tr>
<td># Constraints: M</td>
<td>x</td>
<td>(x-1)*D</td>
</tr>
<tr>
<td>Size(dz/dp) = N*p</td>
<td>u<em>D</em>p</td>
<td>(u+x)<em>D</em>p</td>
</tr>
<tr>
<td>Size(dz/dq) = N*M</td>
<td>u<em>x</em>D</td>
<td>(u<em>x+x²)</em>(D² - D)</td>
</tr>
</tbody>
</table>

Example: Entry Problem: x=10, u=1, D=50, no path constraints!

SS size(dz/dq) = 500
FD size(dz/dq) = 269500

Lower bound on multiplication operations required per update iteration

⇒ SS clearly preferable as it will allow for a much higher trajectory generation frequency