Interplanetary Trajectories High-Order Guidance Based on Differential Algebra

R. Armellin, P. Di Lizia, 
*Politecnico di Milano*

K. Makino and M. Berz 
*Michigan State University*
Motivation

- **Space trajectory design** is always affected by uncertainties
  - Uncertainties due to navigation systems (errors on the knowledge of the vehicle position and velocity)
  - Uncertainties in modeling both the environment and the system performances (e.g. atmosphere density, vehicle aerodynamic parameters, perturbations)
- A **guidance strategy** is required in order to achieve missions goals regardless the presence of such uncertainties
Outline

- Notes on Differential Algebra
- High-Order DA guidance algorithm
- Algorithm applications
  - Aerocapture maneuver
  - Low-thrust Earth-Mars transfer
- Conclusions and final remarks
Differential Algebra: Some History

- Differential Algebra (DA) is an automatic differentiation technique
- DA was first developed by Martin Berz in the late ‘80s:
  - **1986**
    Definition of the algebra of Taylor Polynomials in the so-called Truncated Power Series Algebra (TPSA).
  - **1987**
    Introduction of methods to treat common elementary functions and the operations on them
  - **1989**
    Introduction of the analytic operations of differentiation and integration (Differential Algebra)
  - **1998**
    Validated Remainder Enhanced Differential Algebra (Taylor Models)

Implementation in COSY-Infinity
The basic idea is to bring the treatment of functions and the operations on them to the computer in a similar way as the treatment of numbers.

- Real numbers are approximated by floating point numbers.
- For each $\ast$, adjoint $\otimes$ can be crafted on floating point numbers.

$T$ is the extraction of Taylor coefficients (equivalence relation).
- The new space can be endowed with corresponding operations.
Minimal Differential Algebra

First order Differential Algebra

- Consider the set of all ordered pairs of reals \((a_0, a_1)\)
- Define the operations:

\[
\begin{align*}
(a_0, a_1) + (b_0, b_1) &:= (a_0 + b_0, a_1 + b_1) \\
t \cdot (a_0, a_1) &:= (t \cdot a_0, t \cdot a_1) \\
(a_0, a_1) \cdot (b_0, b_1) &:= (a_0 \cdot b_0, a_0 \cdot b_1 + a_1 \cdot b_0) \\
(a_0, a_1)^{-1} &:= (1/a_0, -a_1/a_0^2)
\end{align*}
\]

- The previous algebra allows the automatic computation of derivatives. E.g.:
  - Assume to have \(f\) and \(g\), and to put their values and derivatives at the origin in \(_1D_1:\ ((f(0), f'(0))\) and \((g(0), g'(0))\)
  - Evaluate:

\[
(f(0), f'(0)) \cdot (g(0), g'(0)) = (f(0) \cdot g(0), f'(0) \cdot g(0) + f(0) \cdot g'(0))
\]
This observation can be used to compute derivatives of many functions starting from the **ordered pair corresponding to the identity function** \([x + x_0] \rightarrow (x_0, 1)\)

E.g.:

\[
\begin{align*}
    f(x) &= \frac{1}{x + 1/x} \quad & f'(x) &= \frac{1}{(x + 1/x)^2} - \frac{1}{x + 1/x} \\
    f(3) &= \frac{3}{10} \quad & f'(3) &= -\frac{2}{25}
\end{align*}
\]

Evaluating \(f\) in \(x + 3 \rightarrow (3, 1)\) using the previous algebra:

\[
    f((3, 1)) = \frac{1}{(3, 1) + 1/(3, 1)} = \frac{1}{(3, 1) + (1/3, -1/9)} = \frac{1}{(10/3, 8/9)} = \left(\frac{3}{10}, -\frac{2}{25}\right)
\]

**Important implementation advantages:**

- ordered pairs \(\rightarrow\) new variable type
- algebra \(\rightarrow\) operator overloading
General Differential Algebra $nD_v$

1. $1D_1$ can be generalized to $nD_v$ for function of $v$ variables and the arbitrary order $n$

$$1D_1 \quad \rightarrow \quad nD_v$$

$$(a_0, a_1) \quad \rightarrow \quad (\ldots, c_{j_1, \ldots, j_v}, \ldots)$$

The vector in $nD_v$ is a collection of all the Taylor coefficients of the function $f$ w.r.t the $v$ variables up to the order $n$.

2. $nD_v$ can be further extended to treat any transcendental function (sin, cos, exp, log, etc.).

3. Real algebra is substituted by Taylor polynomial algebra.

4. Starting from the Taylor polynomial of the identity function, the DA computation of $f$ returns its Taylor expansion.
High-Order Guidance Algorithm

- Start from a trajectory computed in nominal conditions typically by solving an optimization problem.

- The nominal trajectory is completely described by:
  - A set of $n$ ODE describing the system dynamics
  - The initial condition $x_0 \in \mathbb{R}^n$
  - A set of $m$ parameters $p$ (it can include also part of the initial condition)
  - A nominal $j$ dimensional control vector $u(t)$ described by the $j \times k$ collocation points of cubic splines

- The goal of robust guidance algorithm is to find the corrections in the control law $u$ to constrain a subset of the final state $x_f^c$ to its nominal value regardless of the uncertainties on $p$. 
Consider

- $m$ uncertain parameters $p$
- $g \leq n$ constraints on the final state $x_f^c$

**Initialize** the uncertain parameter vector as DA

$[p] = p + \delta p$

**Initialize** $g$ collocation points of the control as DA variables $[u^g] = u^g + \delta u^g$

By means of DA numerical integration obtain the $n$-th order **Taylor expansion of the flow** of the ODE w.r.t. $m+g$ DA variables

$\delta x_f^c = M_{x_f^c}(\delta u^g, \delta p)$ \hspace{1cm} (1)

The map describes the variation of the constrained part of the final state as a function of controls and parameters
High-Order Guidance Algorithm

- The m-th dimensional identity map in the parameters $\delta p = \mathcal{I}_p(\delta p)$ is added to (1) resulting in

\[
\begin{pmatrix}
\delta x^c_f \\
\delta p
\end{pmatrix}
= \begin{pmatrix}
\mathcal{M}^{x^c_f} \\
\mathcal{I}_p
\end{pmatrix}
\begin{pmatrix}
\delta u^g \\
\delta p
\end{pmatrix}
\]

- The map is then inverted

\[
\begin{pmatrix}
\delta u^g \\
\delta p
\end{pmatrix}
= \begin{pmatrix}
\mathcal{M}^{x^c_f} \\
\mathcal{I}_p
\end{pmatrix}^{-1}
\begin{pmatrix}
\delta x^c_f \\
\delta p
\end{pmatrix}
\]

- It is then evaluated for $\delta x^c_f = 0$

\[
\begin{pmatrix}
\delta u^g \\
\delta p
\end{pmatrix}
= \begin{pmatrix}
\mathcal{M}^{x^c_f} \\
\mathcal{I}_p
\end{pmatrix}^{-1}
\begin{pmatrix}
0 \\
\delta p
\end{pmatrix}
\]

- The set of the first $g$ equations is the sought n-th order Taylor guidance, i.e. $\delta u^g = \mathcal{M}_{u^g}(\delta p)$. 

Roberto Armellin
Aerocapture Maneuver (AC)

Classic Orbit Circularization

- Target orbit
- \( V_\infty \)\: Incoming hyperbolic leg
- \( \Delta V \)
- \( V_t \)
- \( V_{R_p} \)

Aerocapture

- Exit trajectory
- \( V_{R_a} \)
- Atmospheric phase
- \( \Delta V \)
- \( V_t \)
- \( V_\infty \)\: Incoming hyperbolic leg
AC Dynamics and Control

- **Two-body problem outside Mars atmosphere**

- Atmospheric dynamics of the vehicle within Mars atmosphere

- Aerodynamic forces

  \[
  D = \frac{1}{2} \rho V^2 SC_D \\
  L = \frac{1}{2} \rho V^2 SC_L
  \]

- **Control parameters**
  - \( \sigma \) bank angle
  - \( \alpha \) angle of attack

- Optimization problem: find the value of \( \alpha \) and the \( \sigma \) profile to **minimize the \( \Delta V \)** required to achieve a Mars polar circular orbit of 200 km of altitude (\( v_\infty = 5 \text{ km/s} \))

\[
\begin{align*}
\dot{R} &= V \sin \gamma \\
\dot{\vartheta} &= \frac{V \cos \gamma \cos \psi}{R \cos \varphi} \\
\dot{\varphi} &= \frac{V \cos \gamma \sin \psi}{R} \\
\dot{V} &= \frac{D}{m} - g \sin \gamma \\
V \dot{\gamma} &= \frac{L \cos \sigma}{m} - g \cos \gamma + \frac{V^2 \cos \gamma}{R} \\
V \dot{\psi} &= \frac{L \sin \sigma}{m \cos \gamma} - \frac{V^2 \tan \varphi \cos \gamma \cos \psi}{R},
\end{align*}
\]
AC Optimal Nominal Solution

- **Multiple-shooting** transcription and **SQP optimizer**
- The optimal **bank profile** is described by means of **10 collocation points**
- The optimal trim angle $\alpha = -19.3$ deg
- The required $\Delta V = 38.25$ m/s is the **TPS mass fraction** is **7.05 %**
AC High-Order Guidance

Taylor polynomial describing how the bank profile changes to satisfy the constraint on the final altitude

\[ \delta \sigma_2 = M_{\sigma_2}(\delta C_L) \]

\[ \delta \sigma_2 = M_{\sigma_2}(\delta C_D) \]

\[ \delta \sigma_2 = M_{\sigma_2}(\delta \rho_0) \]

![Diagram showing controlled and uncontrolled trajectories](image-url)
The guidance law reduces to a polynomial evaluation

The collocation point to change is selected by means of a sensitivity analysis

The solutions are near optimal

- $\Delta V = 107 - 109 \text{ m/s}$
- TPS fraction = 6.35 - 7.44 \%
Low-Thrust Earth-Mars Transfer

- Simple two-body controlled dynamics
  \[
  \begin{align*}
  \dot{r} &= v \\
  \dot{v} &= -\frac{\mu}{r^3} r + u,
  \end{align*}
  \]

- Rendezvous with Mars and \( C_3 = 0 \)

- Variables: acceleration profiles on 3 axes, departure epoch, and time of flight

- 4 collocation points for each acceleration components

- Optimization problem: find the control law, the departure epoch, and the time of flight that maximize of the spacecraft final mass fraction

\[
\frac{m(t)}{m_0} = e^{-\int_{t_0}^{t} \frac{u \, d\tau}{I_{sp} g_0}}
\]
Low-Thrust Earth-Mars Transfer

- **Simple-shooting** transcription and **SQP optimizer**
- 210 days **impulsive** transfer as **first guess** solution
- Optimal solution
  - Departure epoch: 1213.78 MJD2000
  - Time of flight: 513.21 days
  - $m(t)/m_0 = 0.801$ (Isp = 3000 s)
High-Order Guidance

- **Uncertain departure epoch**, 10 days box
- DA evaluation of Earth ephemerides translates into an *uncertain the departure state* (on the 6 components)
- **Corrections** on two collocation points for each acceleration components are computed to guarantee the rendezvous with Mars at the final time
Conclusions and Final Remarks

- DA algebra reduces the guidance problem to a simple polynomial evaluation
- Several sources of uncertainties can be taken into account
- Feedback control laws can be obtained
- Constraints on the controls variation cannot be explicitly imposed
- Exploit remaining control collocation points to gain guidance optimality
- Automatic selection of the collocation points to be modified
Interplanetary Trajectories High-Order Guidance Based on Differential Algebra

R. Armellin, P. Di Lizia,
*Politecnico di Milano, Milano, Italy*

and M. Berz
*Michigan State University, East Lansing, MI*

Taylor Methods and Computer Assisted Proofs
Barcelona, June, 3 – 7, 2008