New Trends in Astrodynamics and Applications VI

Courant Institute of Mathematical Sciences

New York, June 6-8, 2011

POLITECNICO DI MILANO

MICHIGAN STATE UNIVERSITY

High-Order Optimal Station Keeping of Geostationary Satellites

P. Di Lizia, R. Armellin, F. Topputo, M. Lavagna, F. Bernelli Zazzera

Department of Aerospace Engineering, Politecnico di Milano, Milano, Italy

M. Berz

Department of Physics and Astronomy, East Lansing, Michigan, USA

Motivations and Goal

- Geostationary satellites move from their nominal path due to
  - Non-spherical gravitational field
  - Third-body perturbations
  - Solar radiation pressure

  ⇨ Station keeping manoeuvres

- Operative life strictly depends on ΔV for station keeping (SK)
  - Recent interest in low-thrust electric propulsion

Impulsive maneuvers ⇨ Continuous maneuvers

- Continuous SK maneuvers are designed by solving an Optimal Feedback Control Problem

- Classical methods are based on linear techniques
  - Pros: fast and easier implementation onboard
  - Cons: inaccurate for large deviations

Motivations and Goal

- Interest in nonlinear control techniques
  - Accurate optimal feedback
  - Tend to be computationally expensive
- Available nonlinear optimal feedback control methods
  - State-dependent (SDRE) or approximating sequence (ASRE) of Riccati equations methods (Cimen and Banks)
  - High order expansion of the generating functions (Scheeres, Park)
- **Goal**: Alternative approach based on Differential Algebra
  - Fast computation of high order optimal feedback control laws
  - High order expansion of ODE flow
  - High order expansion of the solution of the Optimal Control Problem

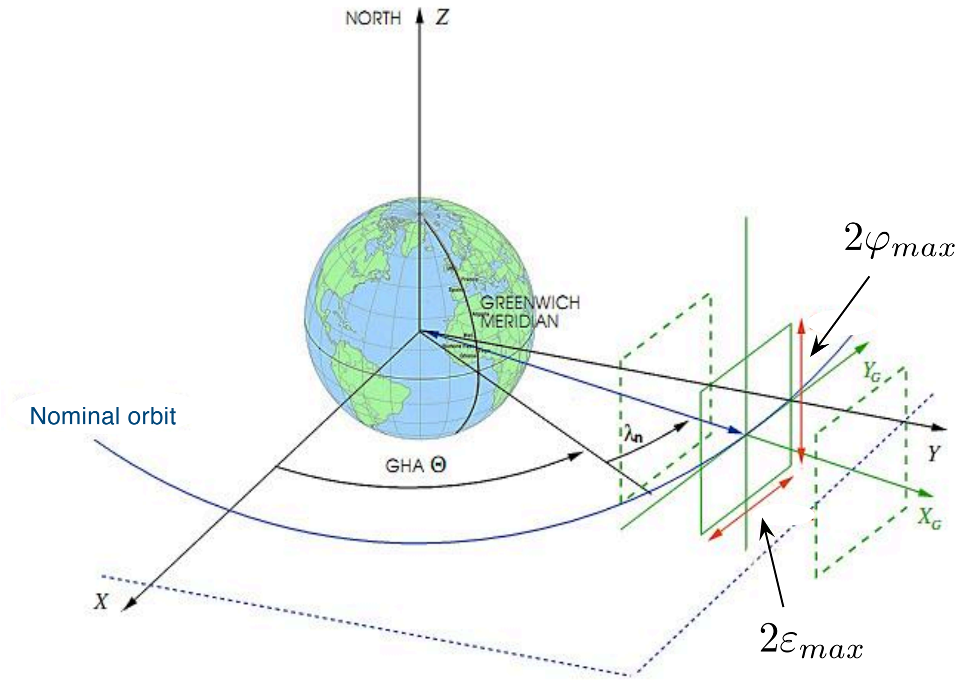
Outline

- Station keeping problem and dynamical models
- Notes on Differential Algebra
- High order expansion of ODE flow
- Optimal station keeping problem
- High order expansion of the optimal station keeping problem

Non-spherical gravitational field	3rd-body perturbation	Solar radiation pressure	
✔			
✔	✔	✔	
✔	✔	✔	→ *Fast correction*

- Conclusions and future work

Station Keeping Problem



- λ_n : nominal longitude

$$\boxed{\lambda_n = 60 \text{ deg}}$$

- φ : latitude
- ε : longitude error

$$\boxed{\varepsilon = \lambda - \lambda_n}$$

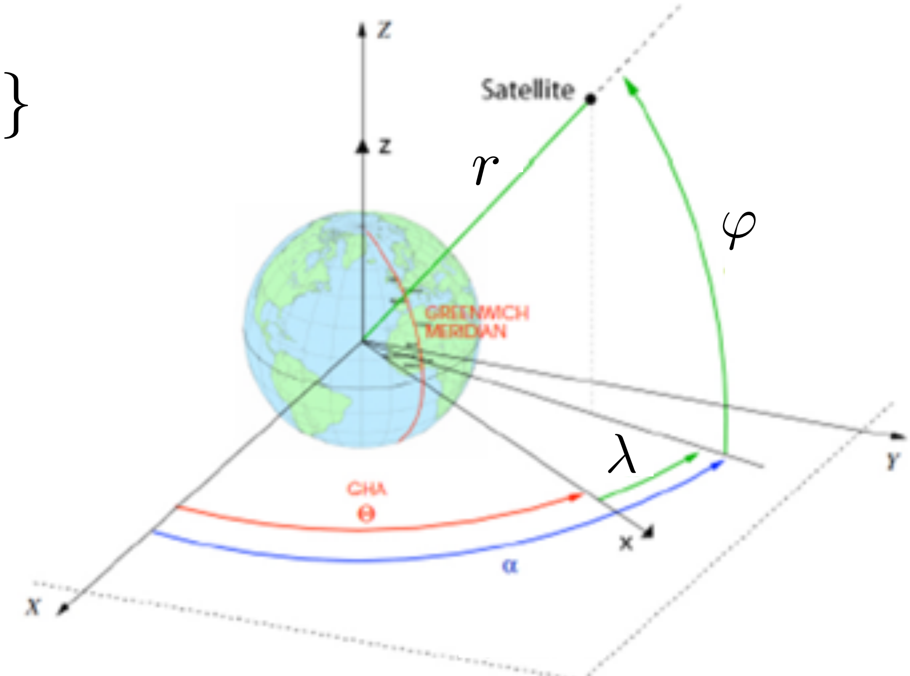
- Given λ_n ⇨ Keep the spacecraft inside the admissible box:

$$-\varepsilon_{max} \le \varepsilon \le \varepsilon_{max}\,, \qquad \varepsilon_{max} = 0.05 \text{ deg}$$

$$-\varphi_{max} \le \varphi \le \varphi_{max}\,, \qquad \varphi_{max} = 0.05 \text{ deg}$$

Station Keeping Dynamical Model

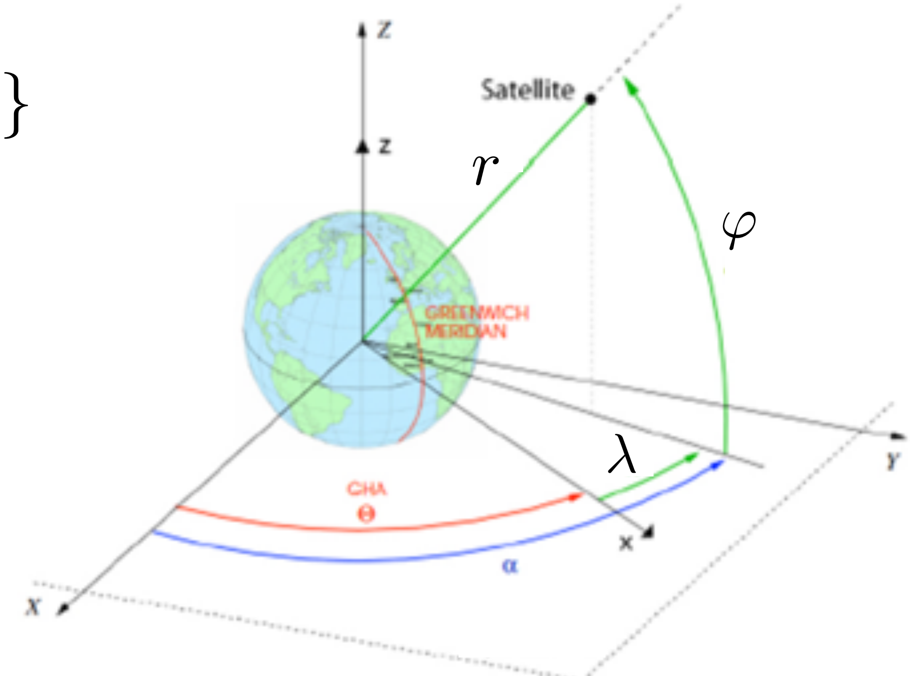
- ECEF reference frame
- Spherical coordinates $\{r, \varepsilon, \varphi\}$
- Kepler's dynamics +
  - Non-spherical gravitational field
  - Third-body perturbations
  - Solar radiation pressure



$$
\begin{cases}
\dot{r} = v \\
\dot{\varepsilon} = \xi \\
\dot{\varphi} = \eta \\
\dot{v} = -\dfrac{\mu}{r^2} + r\eta^2 + r(\xi + \omega)\cos^2\varphi + a_{p_r}(r, \varepsilon, \varphi) + u_r(t) \\
\dot{\xi} = 2\eta(\xi + \omega)\tan\varphi - 2\dfrac{v}{r}(\xi + \omega) + \dfrac{1}{r\cos\varphi} a_{p_\varphi}(r, \varepsilon, \varphi) + \dfrac{1}{r\cos\varphi} u_\varepsilon(t) \\
\dot{\eta} = -2\dfrac{v}{r}\eta - (\xi + \omega)^2 \sin\varphi\cos\varphi + \dfrac{1}{r} a_{p_\varepsilon}(r, \varepsilon, \varphi) + \dfrac{1}{r} u_\varphi(t)
\end{cases}
$$

Station Keeping Dynamical Model

- ECEF reference frame
- Spherical coordinates $\{r, \varepsilon, \varphi\}$
- Kepler's dynamics +
  - Non-spherical gravitational field
  - Third-body perturbations
  - Solar radiation pressure

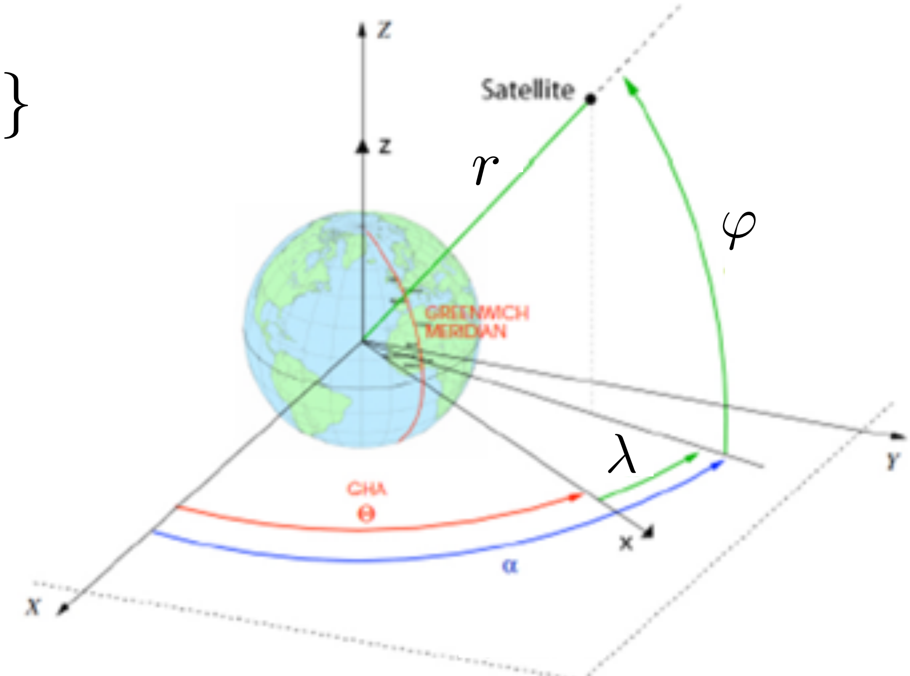


$$
\begin{cases}
\dot{r} = v \\
\dot{\varepsilon} = \xi \\
\dot{\varphi} = \eta \\
\dot{v} = -\dfrac{\mu}{r^2} + r\eta^2 + r(\xi + \omega)\cos^2\varphi + a_{p_r}(r,\varepsilon,\varphi) + u_r(t) \\
\dot{\xi} = 2\eta(\xi+\omega)\tan\varphi - 2\dfrac{v}{r}(\xi+\omega) + \dfrac{1}{r\cos\varphi}a_{p_\varphi}(r,\varepsilon,\varphi) + \dfrac{1}{r\cos\varphi}u_\varepsilon(t) \\
\dot{\eta} = -2\dfrac{v}{r}\eta - (\xi+\omega)^2\sin\varphi\cos\varphi + \dfrac{1}{r}a_{p_\varepsilon}(r,\varepsilon,\varphi) + \dfrac{1}{r}u_\varphi(t)
\end{cases}
$$

perturbations

Station Keeping Dynamical Model

- ECEF reference frame
- Spherical coordinates $\{r, \varepsilon, \varphi\}$
- Kepler's dynamics +
  - Non-spherical gravitational field
  - Third-body perturbations
  - Solar radiation pressure

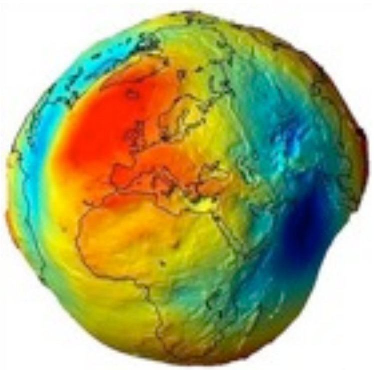


$$
\begin{cases}
\dot{r} = v \\
\dot{\varepsilon} = \xi \\
\dot{\varphi} = \eta \\
\dot{v} = -\dfrac{\mu}{r^2} + r\eta^2 + r(\xi+\omega)\cos^2\varphi + a_{p_r}(r,\varepsilon,\varphi) + u_r(t) \\
\dot{\xi} = 2\eta(\xi+\omega)\tan\varphi - 2\dfrac{v}{r}(\xi+\omega) + \dfrac{1}{r\cos\varphi}a_{p_\varphi}(r,\varepsilon,\varphi) + \dfrac{1}{r\cos\varphi}u_\varepsilon(t) \\
\dot{\eta} = -2\dfrac{v}{r}\eta - (\xi+\omega)^2\sin\varphi\cos\varphi + \dfrac{1}{r}a_{p_\varepsilon}(r,\varepsilon,\varphi) + \dfrac{1}{r}u_\varphi(t)
\end{cases}
$$

control

Station Keeping Dynamical Model

- Non-spherical gravitational field

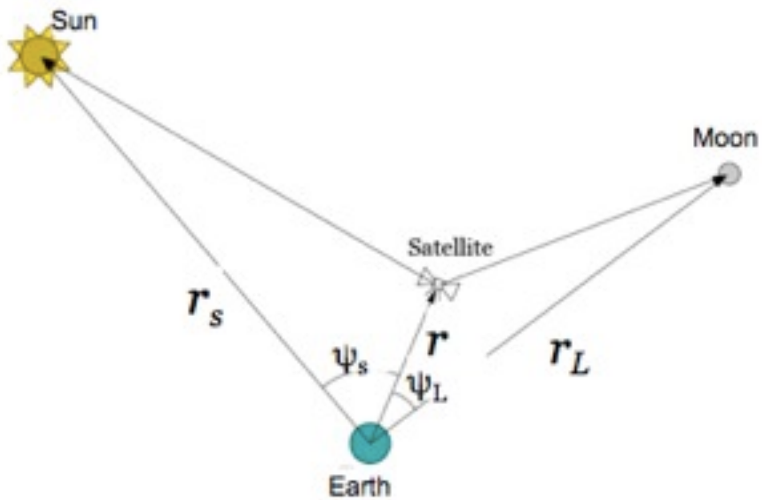


  - Gravitational potential model

$$U = \frac{\mu}{r}\sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{R_T}{r}\right)^{l} P_{l,m}[\sin\varphi]\left\{C_{l,m}\cos m\lambda + S_{l,m}\sin m\lambda\right\}$$

  - Truncation: $l = m = 3 \Rightarrow \mathbf{a}_{gg}(\mathbf{x})$

- 3-rd body perturbation

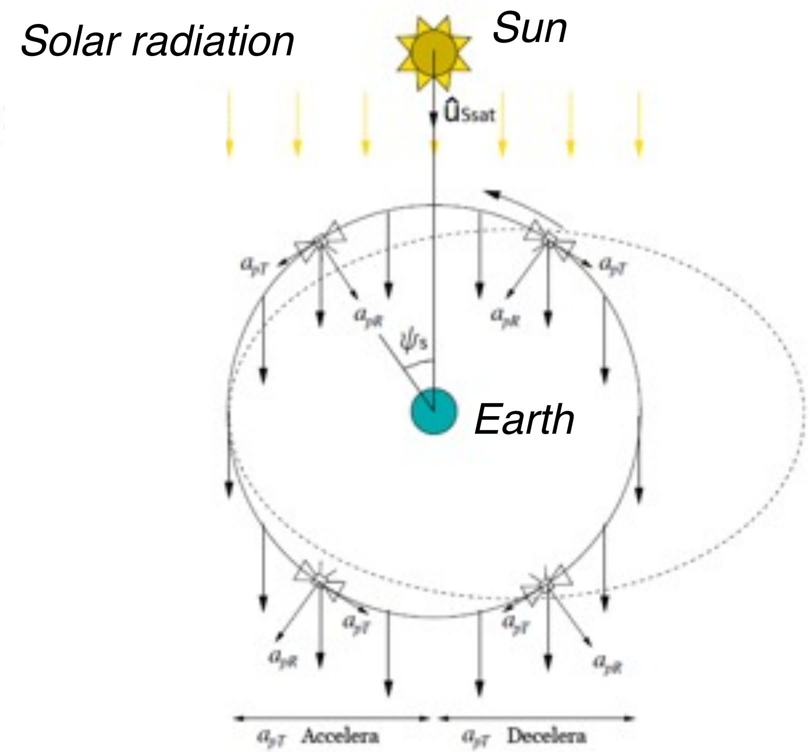


  - Gravitational potential model

$$U_3 = \frac{\mu_3}{r_3}\left[1 + \sum_{k=2}^{\infty}\left(\frac{r}{r_3}\right)^{k} P_k(\cos\psi)\right]$$

  - Truncation: $k = 2 \Rightarrow \mathbf{a}_{3b}(\mathbf{x}, t)$

Station Keeping Dynamical Model



- Solar radiation pressure
  - acceleration:

$$\mathbf{a}_{sp} = P_{sr}(1+\beta)\frac{A}{m}\hat{\mathbf{u}}_{Ssat} = \mathbf{a}_{sp}(\mathbf{x}, t)$$

  - where:

$$P_{sr} = \frac{C_S}{c} = \frac{1353[W/m^2]}{299792458[m/s]}$$

A: surface $\perp$ to radiation

Observation

- An ephemeris model is used for Earth, Moon, and Sun positions
- Kepler + $\mathbf{a}_{gg}$ ⇒ *autonomous dynamics*
- Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$ ⇒ *non-autonomous dynamics*

Notes on Differential Algebra

- Differential Algebra (DA) is an automatic differentiation technique

Algebra of real numbers	$\Rightarrow$	Algebra of Taylor polynomials

- Unlike standard automatic differentiation tools, the analytic operations of differentiation and antiderivation are introduced

- DA can be easily implemented in a computer environment (COSY-Infinity, Berz and Makino, 1998)

- Given any sufficiently regular function f of v, DA enables the computation of its Taylor expansion up to an arbitrary order n

High Order Expansion of ODE Flow

- Consider the ODE initial value problem:

$$\dot{x} = f(x), \; x(0) = x_0$$

- Any integration scheme is based on algebraic operations, involving the evaluation of f at several integration points

- Initialize x_0 as a DA $[x_0] = x_0 + \delta x_0$
- Operate in the DA framework

Taylor expansion of the ODE flow

$$x_f = \mathcal{M}_{x_f}(\delta x_0)$$

- Example: explicit Euler's scheme

$$x_{k+1} = x_k + f(x_k) \cdot h$$

High Order Expansion of ODE Flow

- Consider the ODE initial value problem:

$$\dot{x} = f(x), \; x(0) = x_0$$

- Any integration scheme is based on algebraic operations, involving the evaluation of f at several integration points

- Initialize x_0 as a DA $[x_0] = x_0 + \delta x_0$
- Operate in the DA framework

Taylor expansion of the ODE flow

$$x_f = \mathcal{M}_{x_f}(\delta x_0)$$

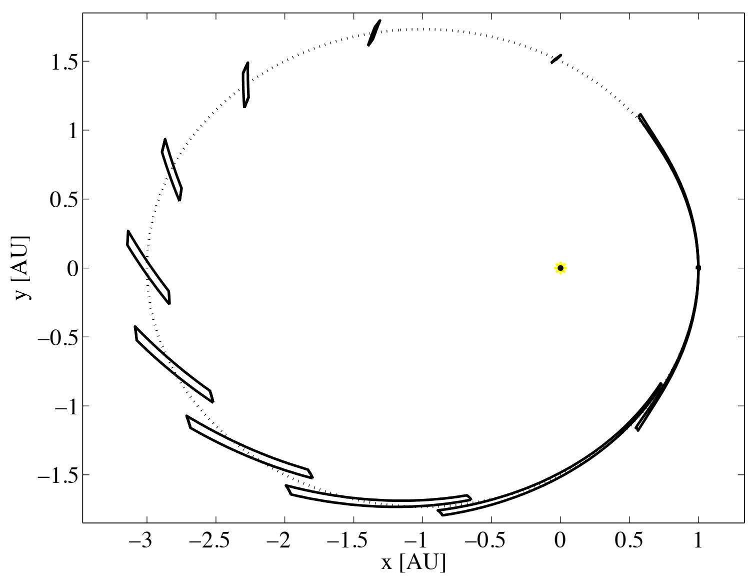
- Example: explicit Euler's scheme

$$[x]_{k+1} = [x]_k + f([x]_k) \cdot h$$

⇒ $[x]_{k+1}$ is the n-th order Taylor expansion of the ODE flow

High Order Sensitivity Analysis

- Example: 2-Body Problem
  - Eccentricity: 0.5 - Starting point: pericenter
  - Integration scheme: Runge-Kutta (variable step, order 8)
  - DA-based ODE flow expansion order: 5
- Uncertainty box on the initial position of 0.01 AU



- Any sample in the uncertainty box can be propagated using the 5th order polynomial

Fast Monte Carlo simulations

Optimal Station Keeping Problem

- Consider the dynamics: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \tilde{\mathbf{f}}(\mathbf{x}, t) + B(\mathbf{x})\,\mathbf{u}$
- Minimizes: $J = \frac{1}{2}\left(\mathbf{x}(t_f) - \mathbf{x}_f\right)^T Q\left(\mathbf{x}(t_f) - \mathbf{x}_f\right) + \int_{t_0}^{t_f} \frac{1}{2}\,\mathbf{u}^T\,\mathbf{u}$
- Initial condition: $\mathbf{x}(t_0) = \mathbf{x}_0$
- Optimal control theory reduces the OCP to the BVP:
  - differential: $\begin{cases} \dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, t) + B(\mathbf{x})\,\mathbf{u} \\ \dot{\boldsymbol{\lambda}} = -\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} \end{cases}$
  - algebraic: $\mathbf{u} + B(\mathbf{x})^T \boldsymbol{\lambda} = 0$
  - subject to: $\mathbf{x}(t_0) = \mathbf{x}_0\;,\quad \boldsymbol{\lambda}(t_f) = Q\left(\mathbf{x}(t_f) - \mathbf{x}_f\right)$

Optimal Station Keeping Problem

- Consider the dynamics: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \tilde{\mathbf{f}}(\mathbf{x}, t) + B(\mathbf{x})\,\mathbf{u}$
- Minimizes: $J = \frac{1}{2}\left(\mathbf{x}(t_f) - \mathbf{x}_f\right)^T Q \left(\mathbf{x}(t_f) - \mathbf{x}_f\right) + \int_{t_0}^{t_f} \frac{1}{2}\,\mathbf{u}^T\,\mathbf{u}$
- Initial condition: $\mathbf{x}(t_0) = \mathbf{x}_0$
- Optimal control theory reduces the OCP to the BVP:
  - differential: $\begin{cases} \dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, t) + B(\mathbf{x})\,\mathbf{u} \\ \dot{\boldsymbol{\lambda}} = -\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} \end{cases}$
  - algebraic: $\mathbf{u} + B(\mathbf{x})^T \boldsymbol{\lambda} = 0 \quad \Rightarrow \quad \boxed{\mathbf{u} = -B(\mathbf{x})^T \boldsymbol{\lambda}}$
  - subject to: $\mathbf{x}(t_0) = \mathbf{x}_0\ , \quad \boldsymbol{\lambda}(t_f) = Q\left(\mathbf{x}(t_f) - \mathbf{x}_f\right)$

High Order Optimal Station Keeping

- The BVP is reduced to a TPBVP on the ODE system:

$$\begin{cases} \dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, t) - B\,B^T \boldsymbol{\lambda} \\ \dot{\boldsymbol{\lambda}} = -\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} \end{cases}$$

- subject to: $\mathbf{x}(t_0) = \mathbf{x}_0$

$$\boldsymbol{\lambda}(t_f) = Q\left(\mathbf{x}(t_f) - \mathbf{x}_f\right)$$

High Order Optimal Station Keeping

- The BVP is reduced to a TPBVP on the ODE system:

$$\begin{cases} \dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, t) - B\,B^T \boldsymbol{\lambda} \\ \dot{\boldsymbol{\lambda}} = -\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} \end{cases}$$

- subject to: $\mathbf{x}(t_0) = \mathbf{x}_0$

$$\boldsymbol{\lambda}(t_f) = Q\,(\mathbf{x}(t_f) - \mathbf{x}_f)$$

- Differential Algebra is applied to expand the solution of the TPBVP up to an arbitrary order w.r.t. $\delta\mathbf{x}_0$:
  - Consider a reference $\mathbf{x}_0$ = nominal geostationary satellite state
  - Consider the reference $\boldsymbol{\lambda}_0 = 0$ $\Rightarrow$ $\boxed{\mathbf{u}_0 = 0}$
  - Initialize the initial state and costate as a DA variable:

$$[\mathbf{x}_0] = \mathbf{x}_0 + \delta\mathbf{x}_0\,, \quad [\boldsymbol{\lambda}_0] = \boldsymbol{\lambda}_0 + \delta\boldsymbol{\lambda}_0$$

High Order Optimal Station Keeping

- Expand the ODE flow w.r.t. $\delta\mathbf{x}_0$ and $\delta\boldsymbol{\lambda}_0$

$$\begin{pmatrix} [\mathbf{x}_f] \\ [\boldsymbol{\lambda}_f] \end{pmatrix} = \begin{pmatrix} \mathcal{M}_{\mathbf{x}_f} \\ \mathcal{M}_{\boldsymbol{\lambda}_f} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix}$$

- Build the map of defects on the final boundary condition:

$$[\mathbf{C}_f] = Q\left([\mathbf{x}_f] - \mathbf{x}_f\right) - [\boldsymbol{\lambda}_f] = \mathcal{M}_{\mathbf{C}_f}(\delta\mathbf{x}_0, \delta\boldsymbol{\lambda}_0)$$

where $\mathbf{x}_f$ is the desired final state

- Build the following map and invert it:

$$\begin{pmatrix} [\mathbf{C}_f] \\ \delta\mathbf{x}_0 \end{pmatrix} = \begin{pmatrix} \mathcal{M}_{\mathbf{C}_f} \\ \mathcal{I}_{\mathbf{x}_0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix} \Rightarrow \begin{pmatrix} \delta\mathbf{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix} = \begin{pmatrix} \mathcal{M}_{\mathbf{C}_f} \\ \mathcal{I}_{\mathbf{x}_0} \end{pmatrix}^{-1} \begin{pmatrix} [\mathbf{C}_f] \\ \delta\mathbf{x}_0 \end{pmatrix}$$

- Impose $[\mathbf{C}_f] = 0 \Rightarrow \boxed{\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)}$

High Order Optimal Station Keeping

$$\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0) \tag{1}$$

- Given any $\delta\mathbf{x}_0$, the evaluation of map (1) delivers the corresponding $\delta\boldsymbol{\lambda}_0$ $\Rightarrow$ optimal station keeping control law

High Order Optimal Station Keeping

$$\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0) \tag{1}$$

- Given any $\delta\mathbf{x}_0$, the evaluation of map (1) delivers the corresponding $\delta\boldsymbol{\lambda}_0$ $\Rightarrow$ optimal station keeping control law

Observation

- Consider the costate dynamics: $\dot{\boldsymbol{\lambda}} = -(\partial\mathbf{f}/\partial\mathbf{x})^T\boldsymbol{\lambda}$

High Order Optimal Station Keeping

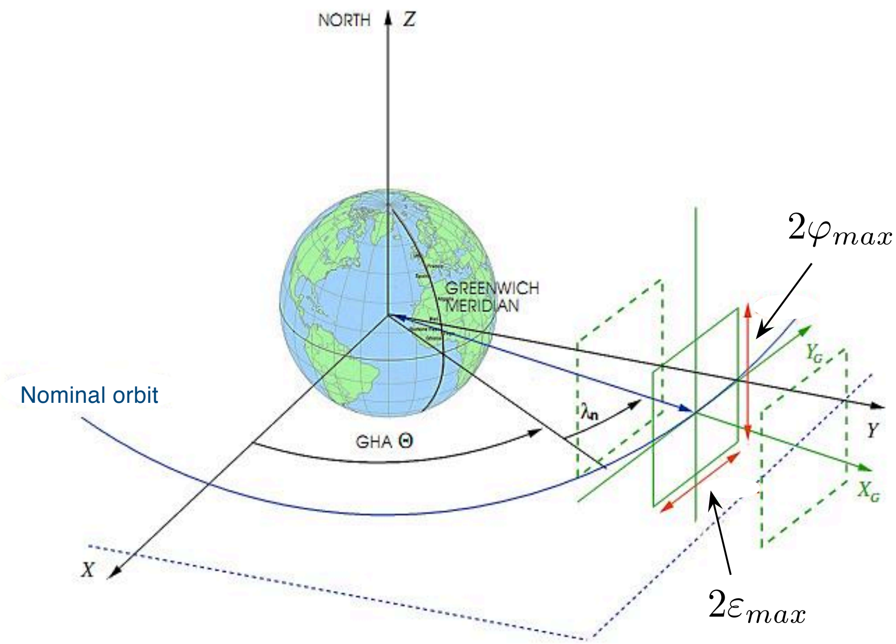
$$\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0) \tag{1}$$

- Given any $\delta\mathbf{x}_0$, the evaluation of map (1) delivers the corresponding $\delta\boldsymbol{\lambda}_0$ $\Rightarrow$ optimal station keeping control law

Observation

- Consider the costate dynamics: $\dot{\boldsymbol{\lambda}} = -\boxed{(\partial\mathbf{f}/\partial\mathbf{x})^T}\boldsymbol{\lambda}$
- DA is used to avoid the analytical computation of $(\partial\mathbf{f}/\partial\mathbf{x})$:
  - Suppose the ***n***-th order solution of the TPBVP is of interest
  - Initialize $\mathbf{x}$ as an ($\boldsymbol{n}$+1)-st order DA number: $[\mathbf{x}] = \mathbf{x} + \delta\mathbf{x}$
  - Compute the ($\boldsymbol{n}$+1)-st order expansion of $\mathbf{f}$: $[\mathbf{f}] = \mathbf{f}([\mathbf{x}]) = \mathcal{M}_{\mathbf{f}}^{n+1}(\delta\mathbf{x})$
  - Use the differentiation: $[\partial\mathbf{f}/\partial\mathbf{x}] = \partial\mathcal{M}_{\mathbf{f}}^{n+1}/\partial\mathbf{x} = \boxed{\mathcal{M}_{\partial\mathbf{f}/\partial\mathbf{x}}^{n}(\delta\mathbf{x})}$

Application: Geostationary Satellite



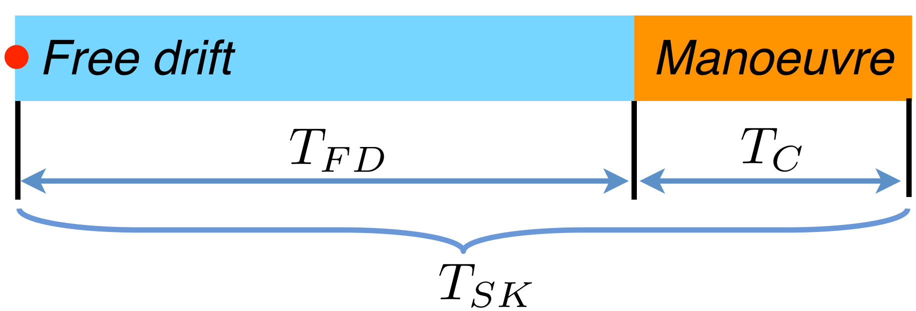
- Nominal longitude: $\lambda_n = 60$ deg
- $\mathbf{x}_0 = \{R_{GEO}, 0, 0, 0, 0, 0\}$
- Admissible box:

  $$-0.05 \text{ deg} \leq \{\varepsilon, \varphi\} \leq 0.05 \text{ deg}$$

- Initial longitude error: -0.04 deg
- Satellite properties:

  $$m = 3000 \text{ kg}\,,\ \ A = 100 \text{ m}^2$$

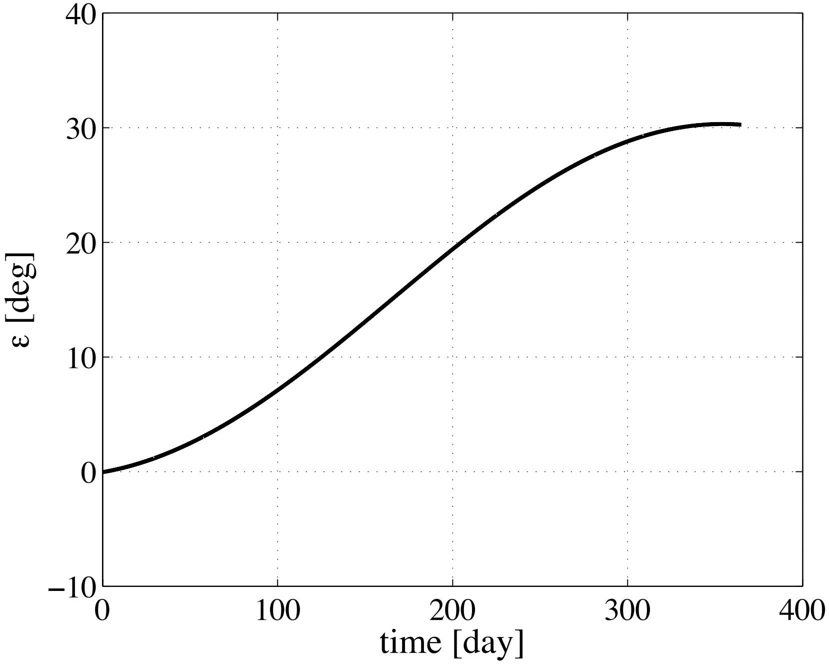
- SK strategy:



  - Free drift: $T_{FD} = 2.5$ day
  - SK manoeuvre: $T_C = 0.5$ day
  - $\mathbf{x}_f \equiv$ initial condition

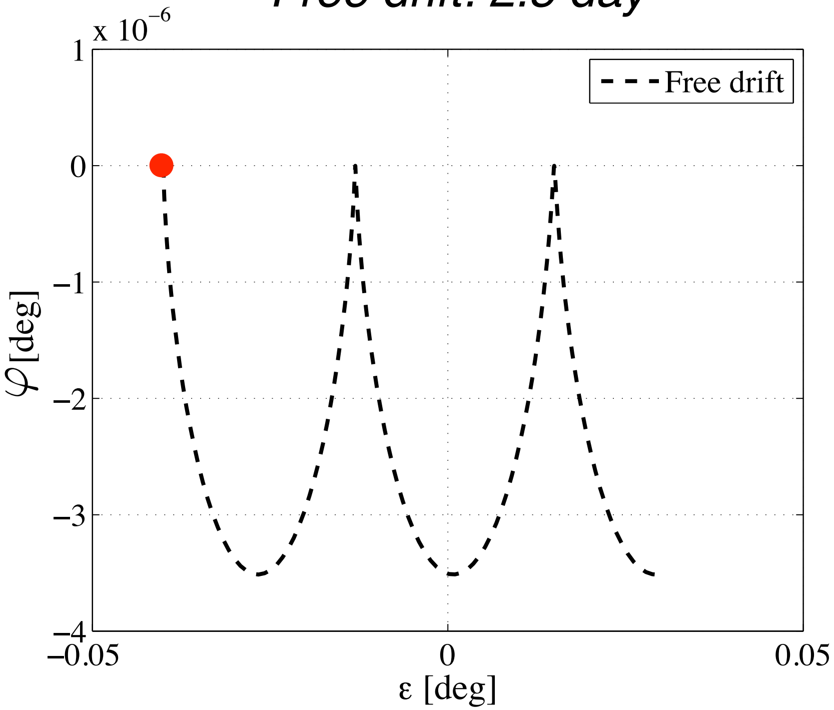
Application: Kepler + $\mathbf{a}_{gg}$

Free drift: 1 year



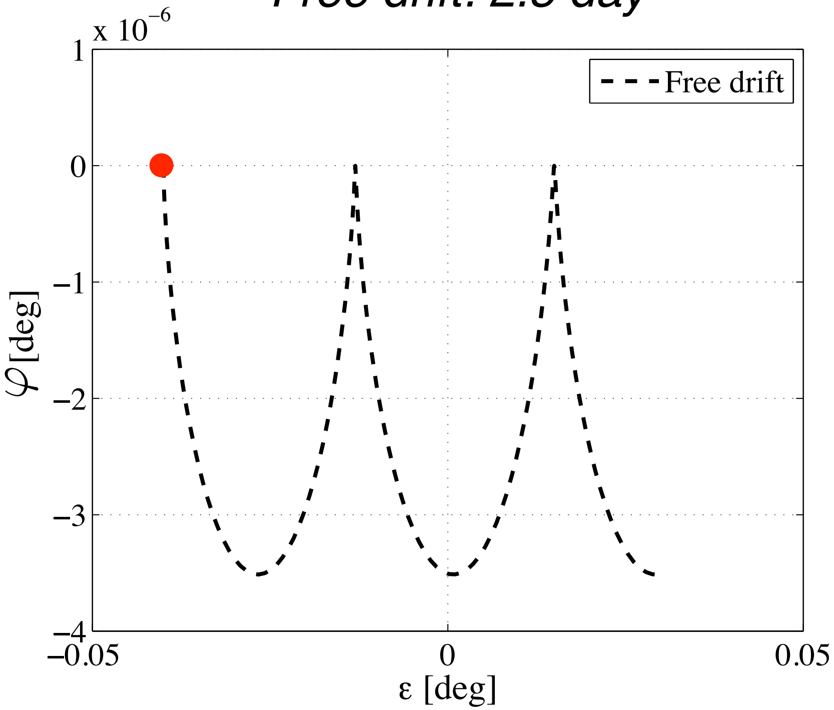
Application: Kepler + $\mathbf{a}_{gg}$

Free drift: 2.5 day



Application: Kepler + $\mathbf{a}_{gg}$

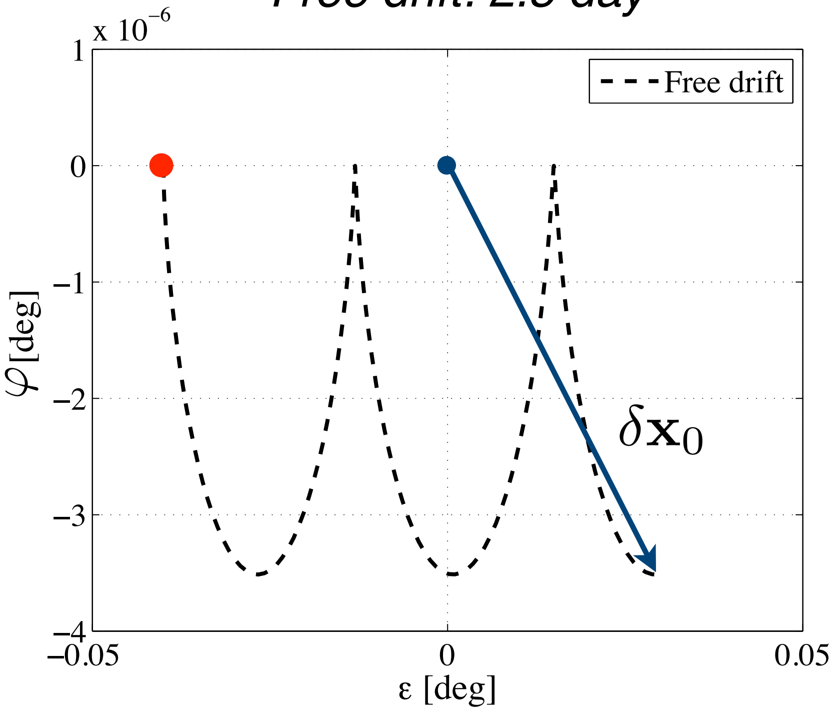
Free drift: 2.5 day



- DA-based 4-th order expansion $\Rightarrow$ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 8.4 s (Mac OS X, 2 GHz Intel Core Duo)

Application: Kepler + $\mathbf{a}_{gg}$

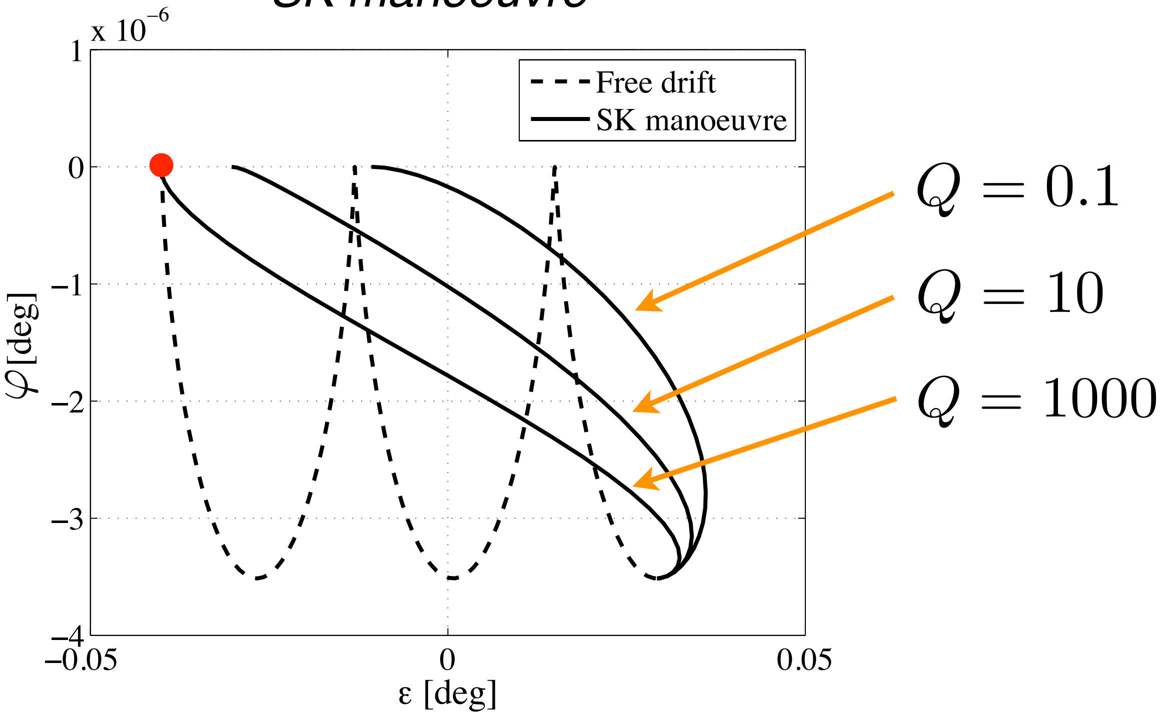
Free drift: 2.5 day



- DA-based 4-th order expansion $\Rightarrow$ $\delta \boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta \mathbf{x}_0)$
  - Computational time: 8.4 s (Mac OS X, 2 GHz Intel Core Duo)

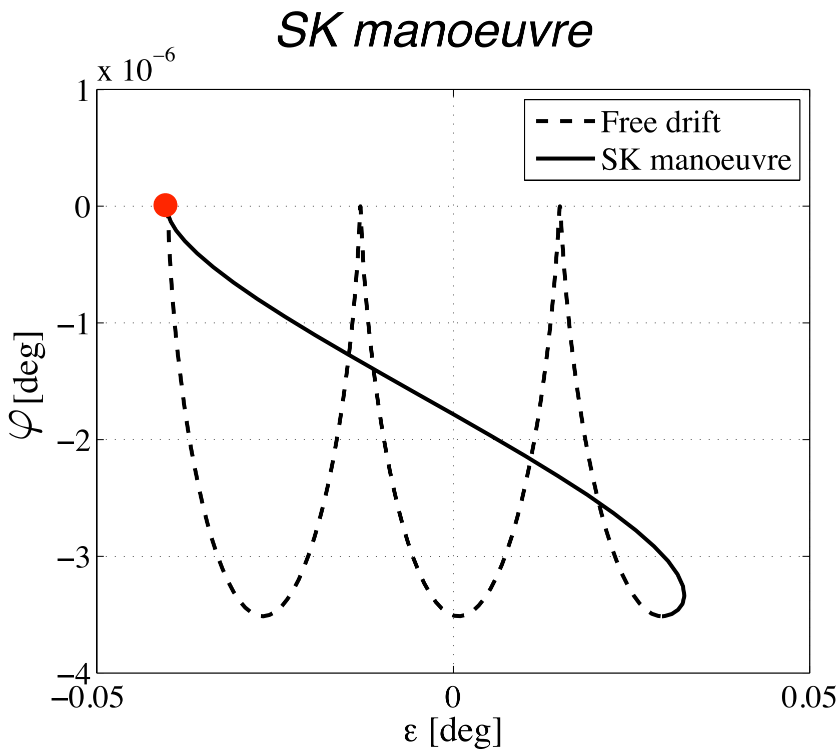
Application: Kepler + $\mathbf{a}_{gg}$

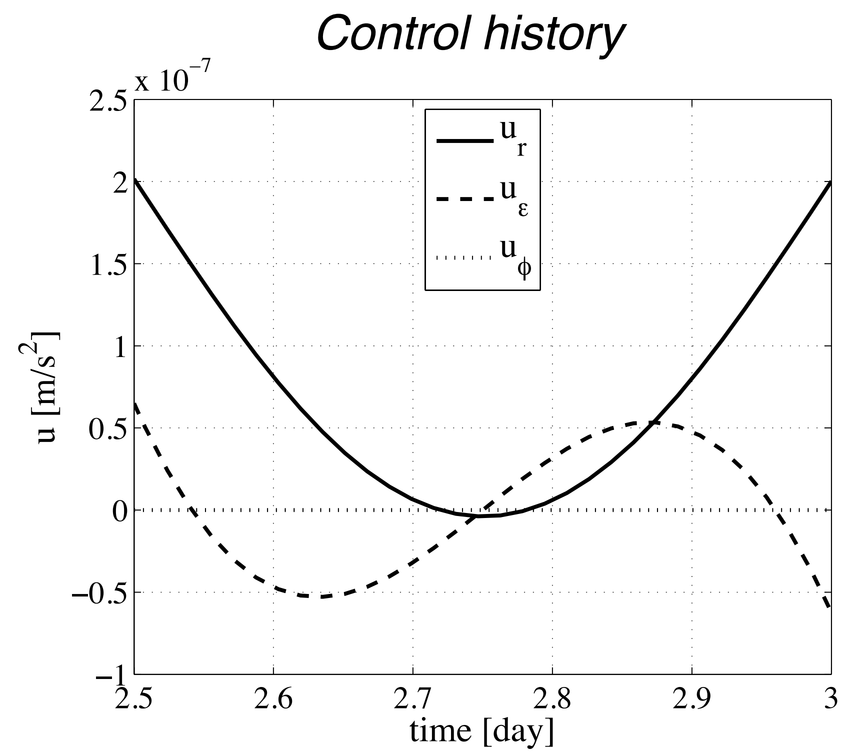
SK manoeuvre



- DA-based 4-th order expansion ⇨ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 8.4 s (Mac OS X, 2 GHz Intel Core Duo)

Application: Kepler + $\mathbf{a}_{gg}$

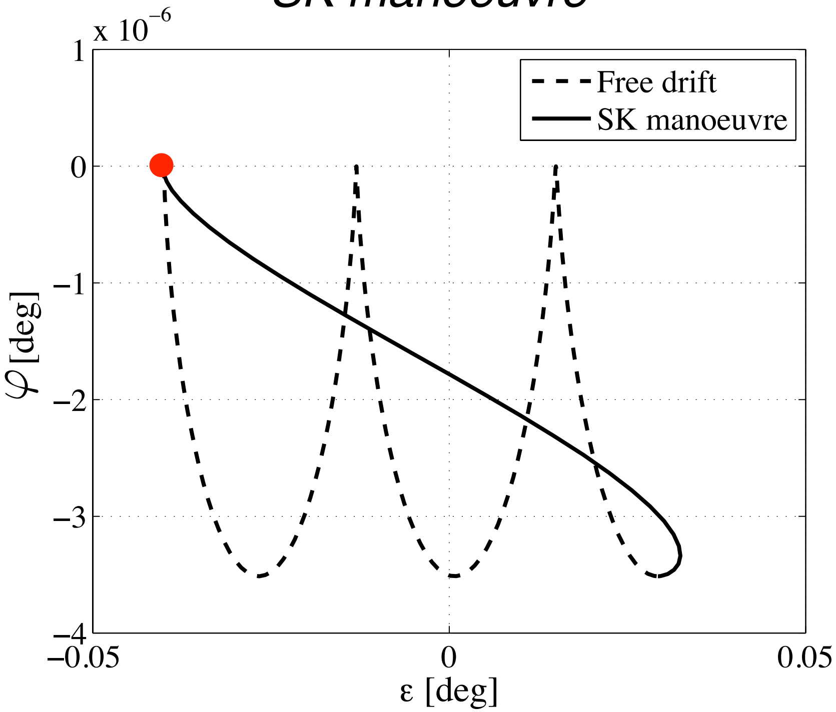




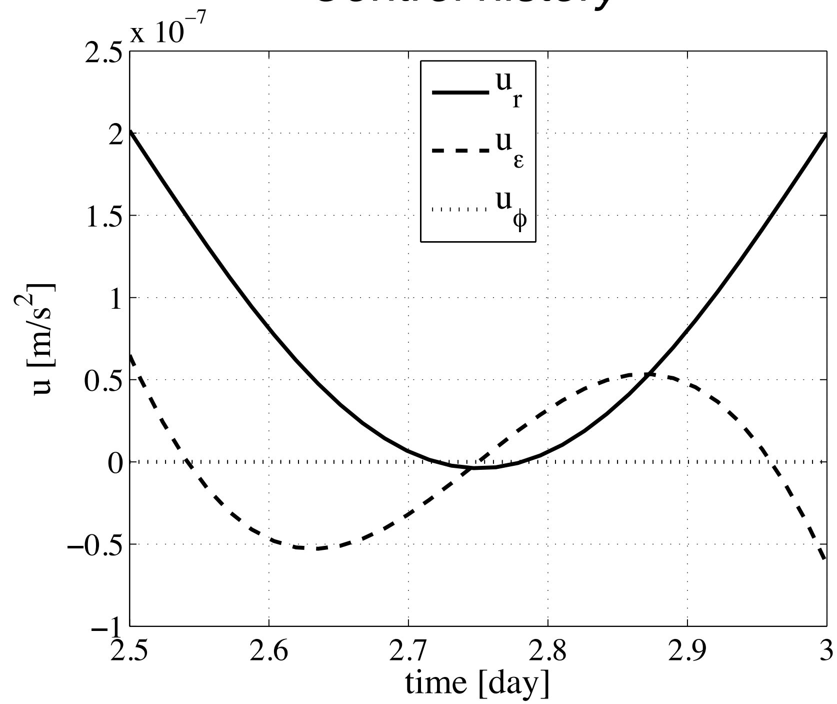
- DA-based 4-th order expansion $\Longrightarrow$ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 8.4 s (Mac OS X, 2 GHz Intel Core Duo)

Application: Kepler + $\mathbf{a}_{gg}$

SK manoeuvre



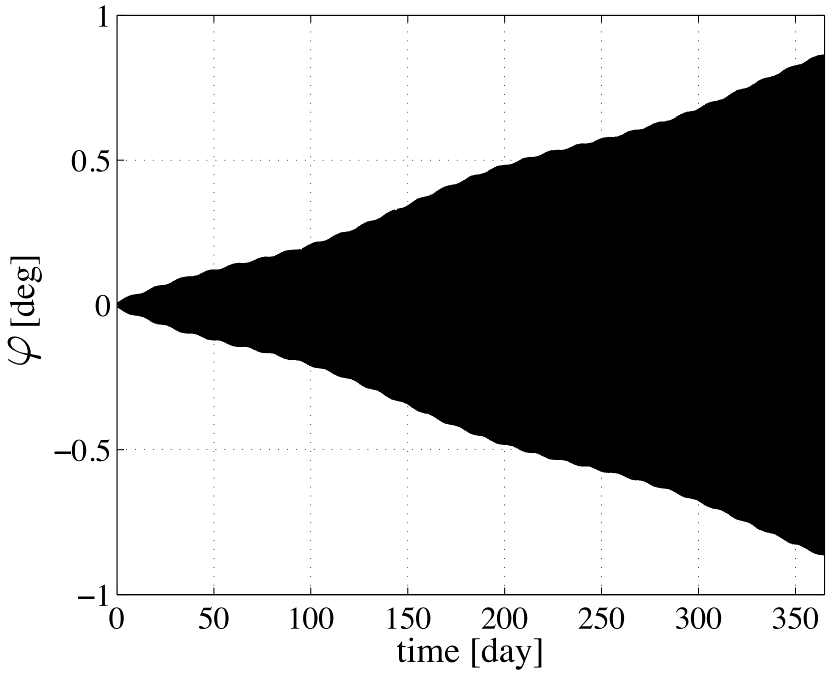
Control history



- DA-based 4-th order expansion $\Rightarrow$ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 8.4 s (Mac OS X, 2 GHz Intel Core Duo)
- Autonomous dynamics $\Rightarrow$ The same polynomial is used for **any** $\delta\mathbf{x}_0$ and t

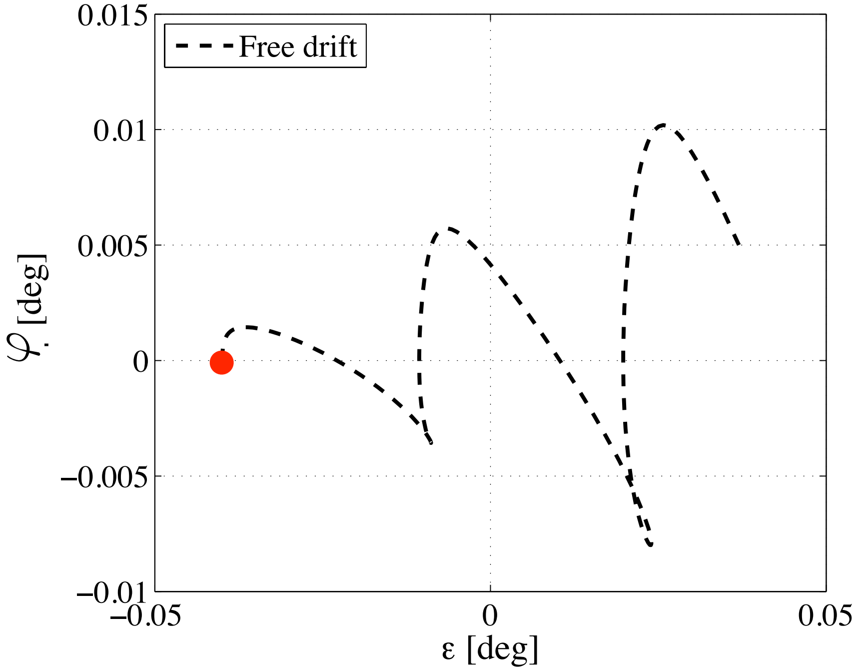
Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

Free drift: 1 year



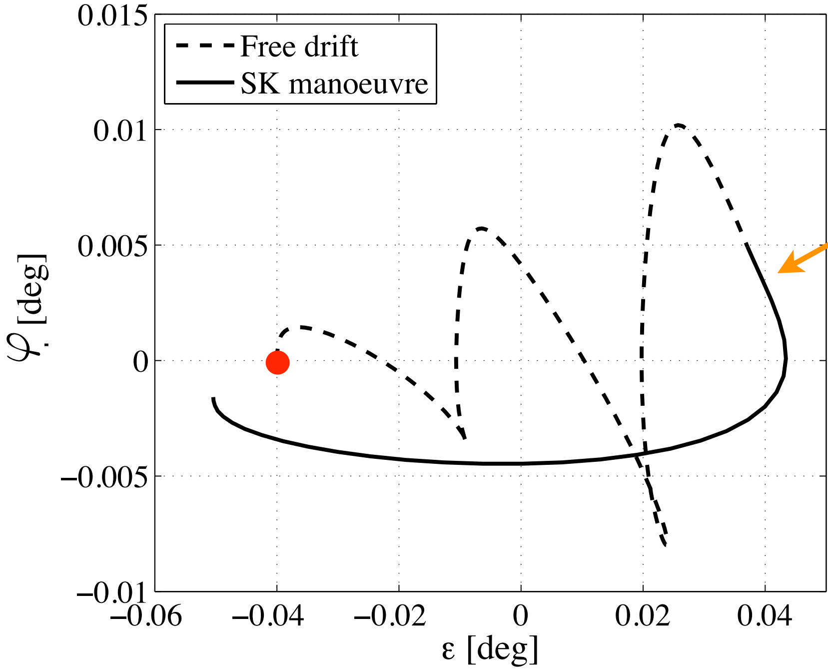
Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

Free drift: 2.5 day



Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

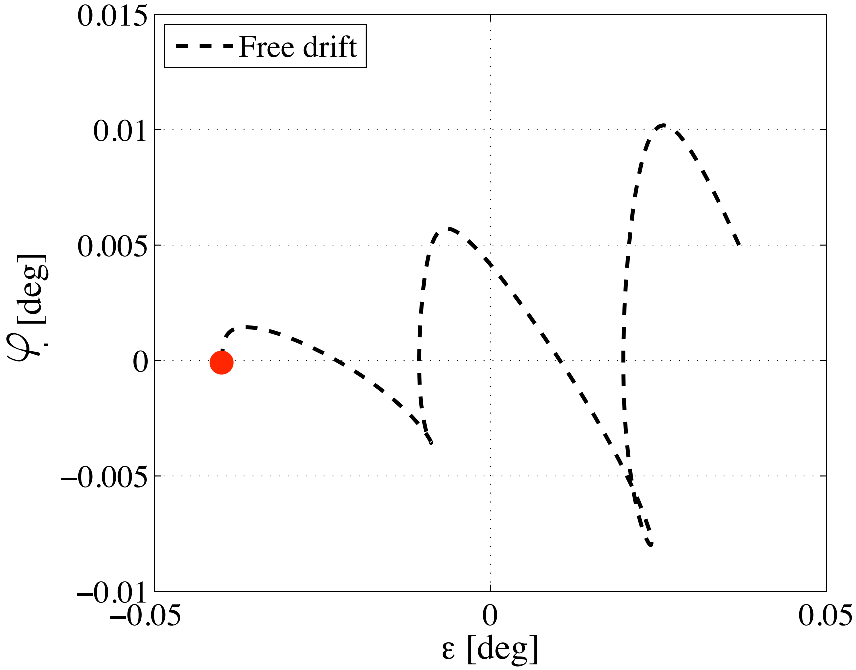
Free drift: 2.5 day



- Kepler + $\mathbf{a}_{gg}$ solution is not sufficiently accurate

Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

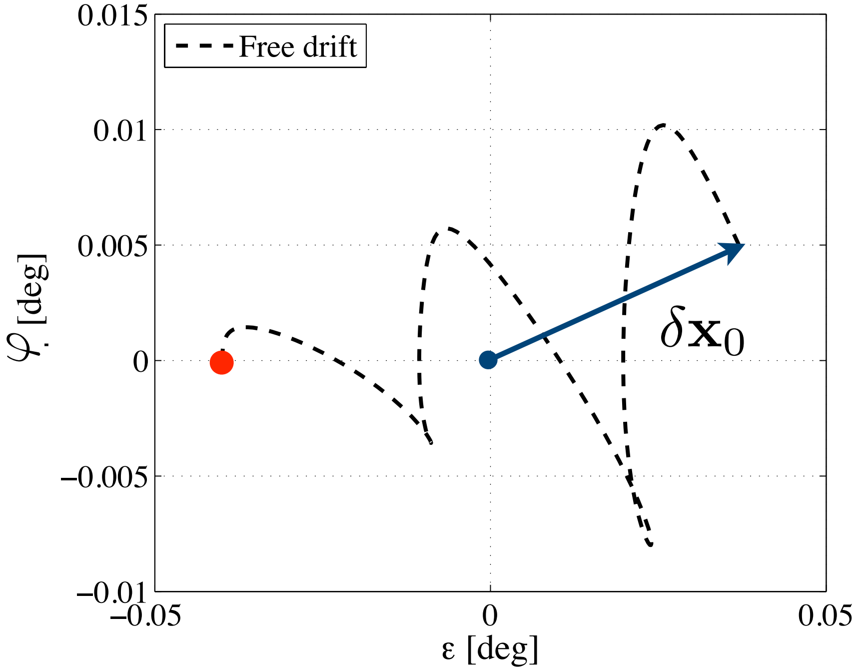
Free drift: 2.5 day



- DA-based 4-th order expansion $\Rightarrow$ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 9.2 s (Mac OS X, 2 GHz Intel Core Duo)

Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

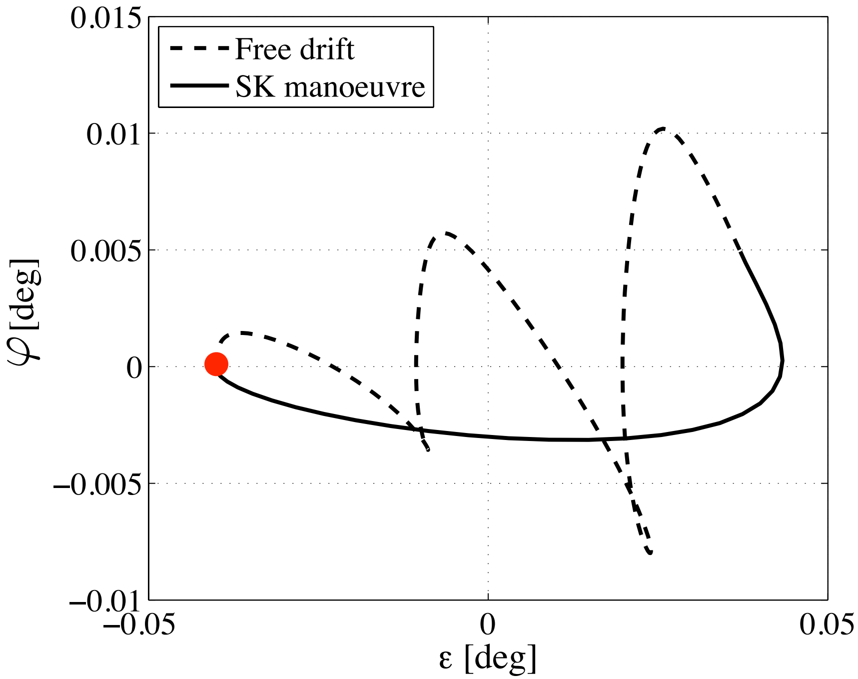
Free drift: 2.5 day



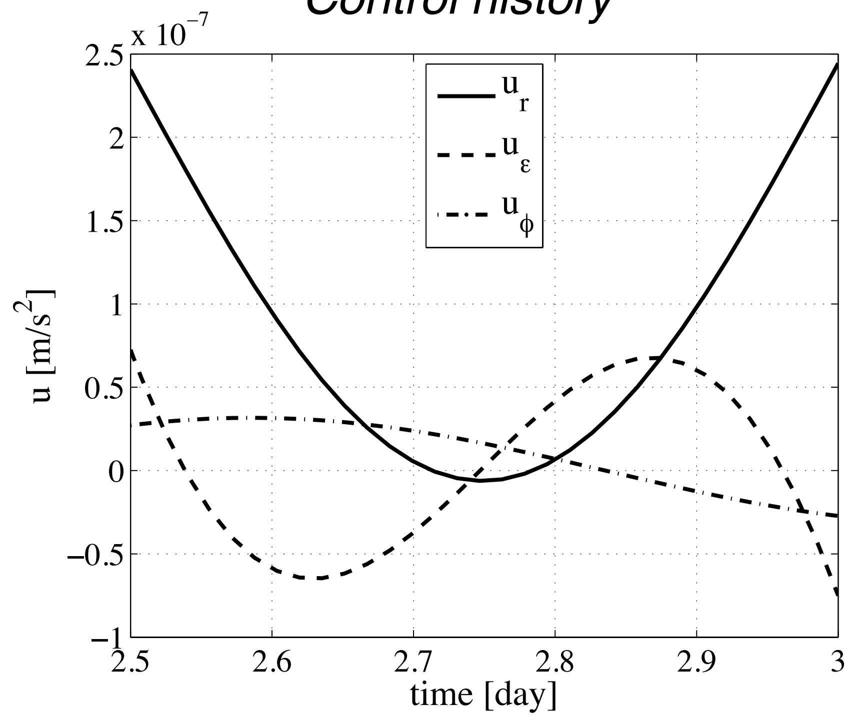
- DA-based 4-th order expansion $\Rightarrow$ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 9.2 s (Mac OS X, 2 GHz Intel Core Duo)

Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

SK manoeuvre



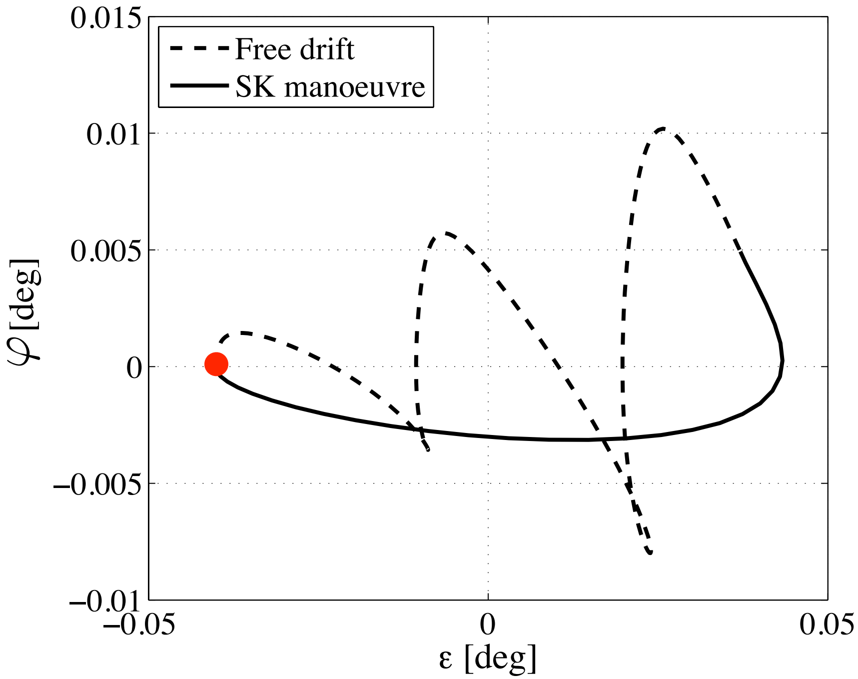
Control history



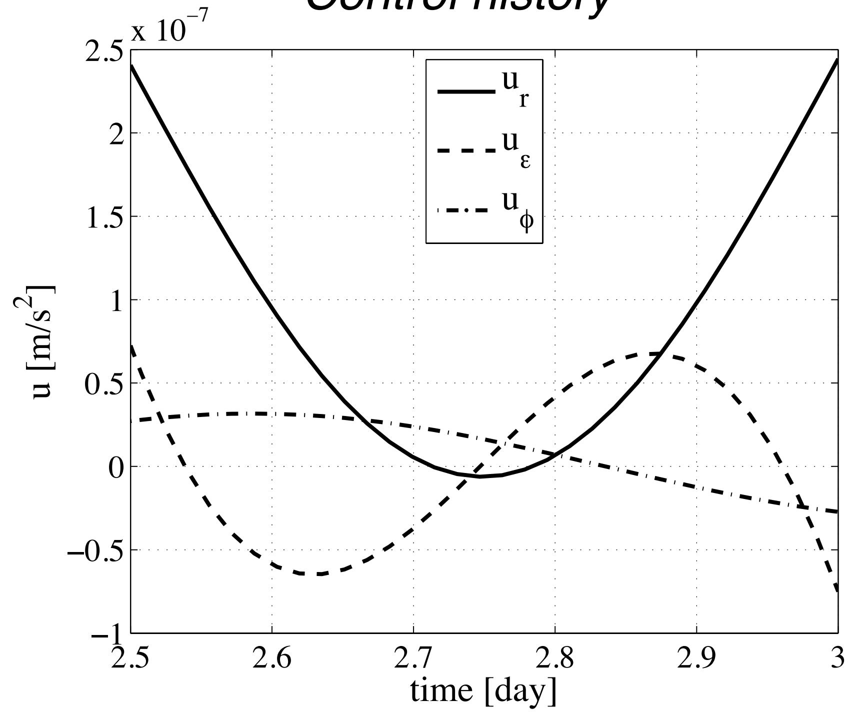
- DA-based 4-th order expansion ⇨ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 9.2 s (Mac OS X, 2 GHz Intel Core Duo)

Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$

SK manoeuvre

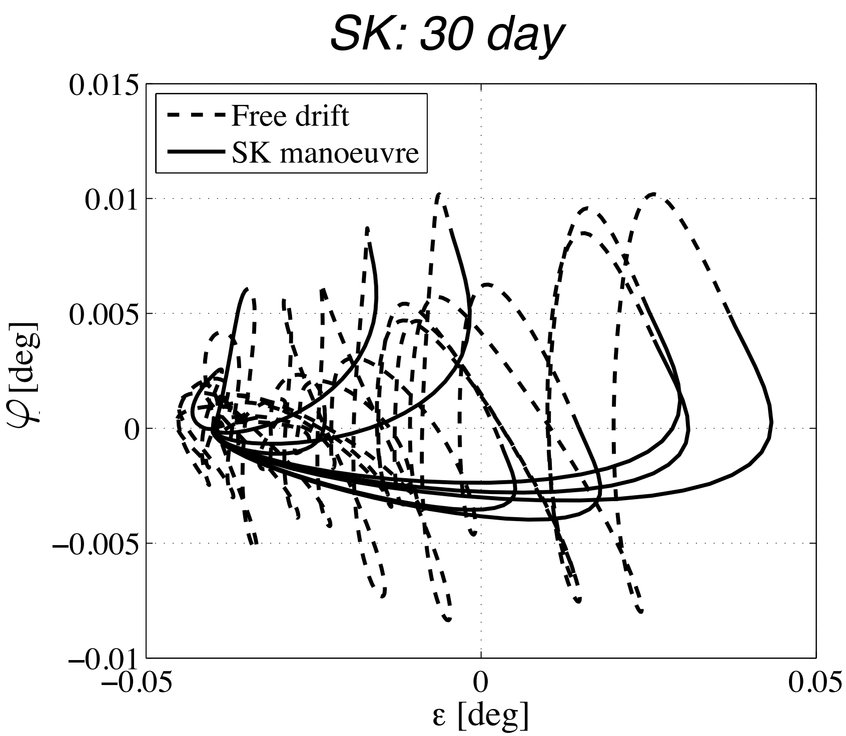


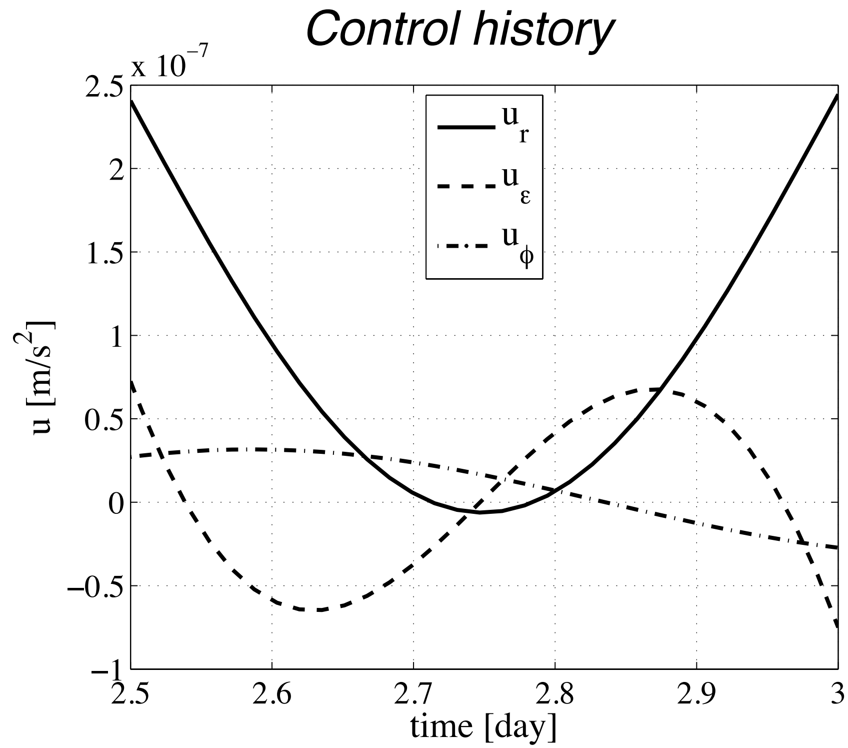
Control history



- DA-based 4-th order expansion ⇨ $\delta\boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)$
  - Computational time: 9.2 s (Mac OS X, 2 GHz Intel Core Duo)
- Non-Autonomous dynamics ⇨ Specific polynomials must be computed for each SK manoeuvre

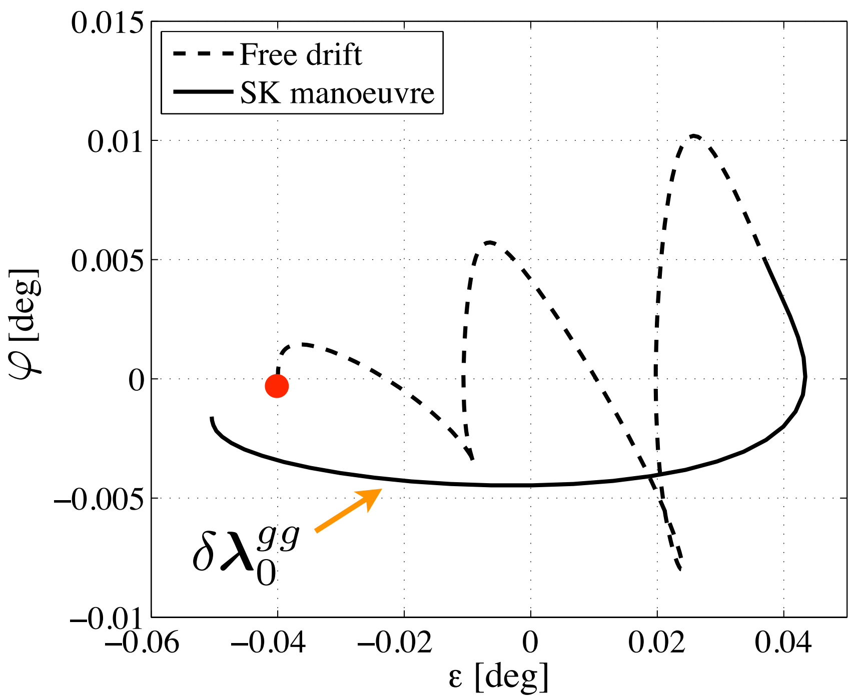
Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$





- DA-based 4-th order expansion $\Rightarrow$ $\delta \boldsymbol{\lambda}_0 = \mathcal{M}_{\mathbf{C}_f=0}(\delta \mathbf{x}_0)$
  - Computational time: 9.2 s (Mac OS X, 2 GHz Intel Core Duo)
- Non-Autonomous dynamics $\Rightarrow$ Specific polynomials must be computed for each SK manoeuvre

Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$



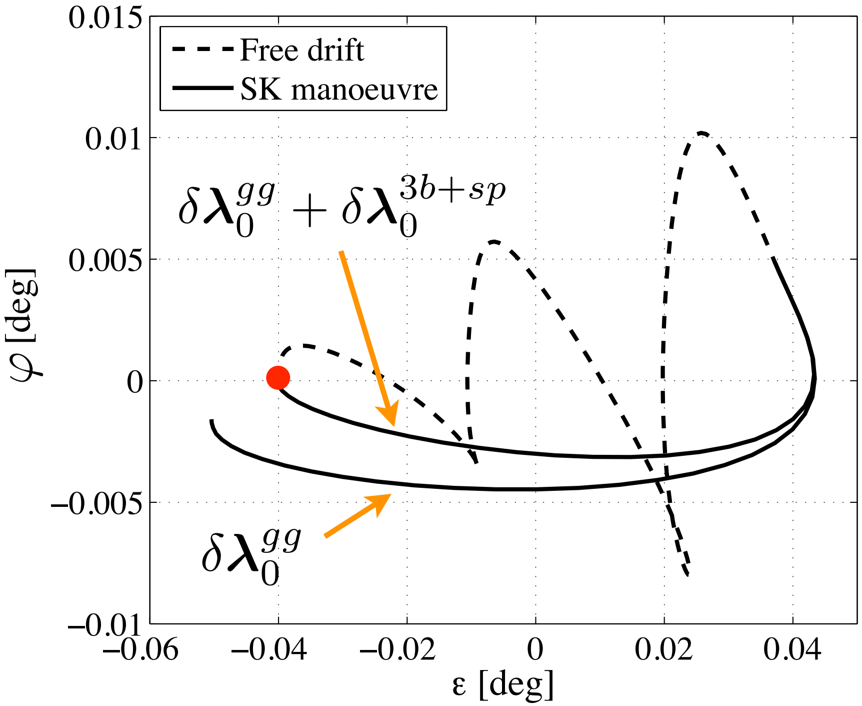
- Kepler + $\mathbf{a}_{gg}$ solution is close to the true solution

- Compute the 4-th order expansion for Kepler + $\mathbf{a}_{gg}$:

$$\boxed{\delta\boldsymbol{\lambda}_0^{gg} = \mathcal{M}^{gg}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)}$$

- For each SK manoeuvre:
  - compute the 1-th order expansion for Kepler + $\mathbf{a}_{gg}$+$\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$ around $\delta\boldsymbol{\lambda}_0^{gg}$ $\Rightarrow$ $\boxed{\delta\boldsymbol{\lambda}_0^{3b+sp} = \mathrm{M}^{3b+sp}_{\mathbf{C}_f=0}\,\delta\mathbf{x}_0}$
  - compute the complete solution: $\delta\boldsymbol{\lambda}_0 = \delta\boldsymbol{\lambda}_0^{gg} + \delta\boldsymbol{\lambda}_0^{3b+sp}$

Application: Kepler + $\mathbf{a}_{gg}$ + $\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$



- Kepler + $\mathbf{a}_{gg}$ solution is close to the true solution

- Compute the 4-th order expansion for Kepler + $\mathbf{a}_{gg}$:

$$\boxed{\delta\boldsymbol{\lambda}_0^{gg} = \mathcal{M}^{gg}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)}$$

- For each SK manoeuvre: (CPU time: **0.07** s)
  - compute the 1-th order expansion for Kepler + $\mathbf{a}_{gg}$+$\mathbf{a}_{3b}$ + $\mathbf{a}_{sp}$ around $\delta\boldsymbol{\lambda}_0^{gg}$ $\Rightarrow$ $\boxed{\delta\boldsymbol{\lambda}_0^{3b+sp} = \mathcal{M}^{3b+sp}_{\mathbf{C}_f=0}(\delta\mathbf{x}_0)}$
  - compute the complete solution: $\delta\boldsymbol{\lambda}_0 = \delta\boldsymbol{\lambda}_0^{gg} + \delta\boldsymbol{\lambda}_0^{3b+sp}$

Conclusions and Future Work

Conclusions

- An high-order nonlinear optimal control method was introduced with application to the station keeping of geostationary satellites
- The method is based on Taylor differential algebra (COSY-Infinity)
- The method ebables the accurate and fast computation of control laws thanks to the computation of proper polynomials

Future work

- Comparison between DA-based and ASRE method
- High order expansion of the solution w.r.t. uncertain parameters
  ⇨ Robustness analysis
- Development of optimal station keeping strategies for propellant mass reduction

New Trends in Astrodynamics and Applications VI
Courant Institute of Mathematical Sciences
New York, June 6-8, 2011

POLITECNICO DI MILANO
MICHIGAN STATE UNIVERSITY





High-Order Optimal Station Keeping of Geostationary Satellites

P. Di Lizia, R. Armellin, F. Topputo, M. Lavagna, F. Bernelli Zazzera
Department of Aerospace Engineering, Politecnico di Milano, Milano, Italy

M. Berz
Department of Physics and Astronomy, East Lansing, Michigan, USA

High Order Expansion of ODE Flow

- Consider the ODE initial value problem:

$$\dot{x} = f(x), \; x(0) = x_0$$

- Any integration scheme is based on algebraic operations, involving the evaluation of f at several integration points

- Initialize x_0 as a DA $[x_0]$
- Operate in the DA framework

Taylor expansion of the ODE flow

$$x_f = \mathcal{M}_{x_f}(\delta x_0)$$

- Example: explicit Euler's scheme

$$x_1 = x_0 + f(x_0) \cdot h$$

High Order Expansion of ODE Flow

- Consider the ODE initial value problem:

$$\dot{x} = f(x), \; x(0) = x_0$$

- Any integration scheme is based on algebraic operations, involving the evaluation of f at several integration points

- Initialize x_0 as a DA $[x_0]$
- Operate in the DA framework

$\Rightarrow$ Taylor expansion of the ODE flow

$$x_f = \mathcal{M}_{x_f}(\delta x_0)$$

- Example: explicit Euler's scheme

$$[x_1] = [x_0] + f([x_0]) \cdot h$$

$\Rightarrow$ $[x_1]$ is the n-th order Taylor expansion of x_1 w.r.t. x_0

High Order Expansion of TPBVPs

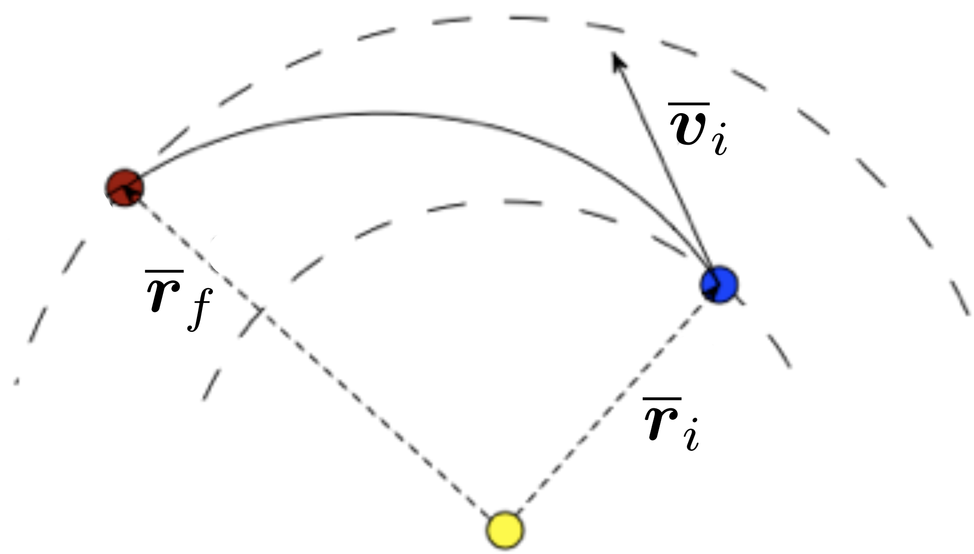
- Example: Lambert's problem
  - initial position: $\overline{\boldsymbol{r}}_i$
  - final position: $\overline{\boldsymbol{r}}_f$
  - time of flight: tof

Find the initial velocity, $\overline{\boldsymbol{v}}_i$, the spacecraft must have to reach $\overline{\boldsymbol{r}}_f$ in tof

- Various algorithms exist to identify a nominal solution of this TPBVP, based on iterative procedures

- Suppose an error $\delta \boldsymbol{r}_i$ occurs on $\overline{\boldsymbol{r}}_i$

What is the new $\boldsymbol{v}_i$?



High Order Expansion of TPBVPs

- Given a nominal solution, $\bar{\boldsymbol{v}}_{\boldsymbol{i}}$ of the Lambert's problem:
  - Expand the ODE flow w.r.t. $\delta\boldsymbol{r}_i$ and $\delta\boldsymbol{v}_i$

$$\begin{pmatrix} \delta\boldsymbol{r}_f \\ \delta\boldsymbol{v}_f \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{M}_{\boldsymbol{v}_f}] \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{r}_i \\ \delta\boldsymbol{v}_i \end{pmatrix}$$

High Order Expansion of TPBVPs

- Given a nominal solution, $\bar{\boldsymbol{v}}_i$ of the Lambert's problem:
  - Expand the ODE flow w.r.t. $\delta \boldsymbol{r}_i$ and $\delta \boldsymbol{v}_i$

$$\begin{pmatrix} \delta \boldsymbol{r}_f \\ \delta \boldsymbol{v}_f \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{M}_{\boldsymbol{v}_f}] \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{r}_i \\ \delta \boldsymbol{v}_i \end{pmatrix}$$

  - Build the following map and invert it:

$$\begin{pmatrix} \delta \boldsymbol{r}_f \\ \delta \boldsymbol{r}_i \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{I}_{\boldsymbol{r}_i}] \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{r}_i \\ \delta \boldsymbol{v}_i \end{pmatrix}$$

High Order Expansion of TPBVPs

- Given a nominal solution, $\bar{\boldsymbol{v}}_i$ of the Lambert's problem:
  - Expand the ODE flow w.r.t. $\delta \boldsymbol{r}_i$ and $\delta \boldsymbol{v}_i$

$$\begin{pmatrix} \delta \boldsymbol{r}_f \\ \delta \boldsymbol{v}_f \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{M}_{\boldsymbol{v}_f}] \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{r}_i \\ \delta \boldsymbol{v}_i \end{pmatrix}$$

  - Build the following map and invert it:

$$\begin{pmatrix} \delta \boldsymbol{r}_f \\ \delta \boldsymbol{r}_i \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{I}_{\boldsymbol{r}_i}] \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{r}_i \\ \delta \boldsymbol{v}_i \end{pmatrix} \Rightarrow \begin{pmatrix} \delta \boldsymbol{r}_i \\ \delta \boldsymbol{v}_i \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{I}_{\boldsymbol{r}_i}] \end{pmatrix}^{-1} \begin{pmatrix} \delta \boldsymbol{r}_f \\ \delta \boldsymbol{r}_i \end{pmatrix}$$

- Given a nominal solution, $\bar{\boldsymbol{v}}_i$ of the Lambert's problem:
  - Expand the ODE flow w.r.t. $\delta\boldsymbol{r}_i$ and $\delta\boldsymbol{v}_i$

$$\begin{pmatrix} \delta\boldsymbol{r}_f \\ \delta\boldsymbol{v}_f \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{M}_{\boldsymbol{v}_f}] \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{r}_i \\ \delta\boldsymbol{v}_i \end{pmatrix}$$

  - Build the following map and invert it:

$$\begin{pmatrix} \delta\boldsymbol{r}_f \\ \delta\boldsymbol{r}_i \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{I}_{\boldsymbol{r}_i}] \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{r}_i \\ \delta\boldsymbol{v}_i \end{pmatrix} \Rightarrow \begin{pmatrix} \delta\boldsymbol{r}_i \\ \delta\boldsymbol{v}_i \end{pmatrix} = \begin{pmatrix} [\mathcal{M}_{\boldsymbol{r}_f}] \\ [\mathcal{I}_{\boldsymbol{r}_i}] \end{pmatrix}^{-1} \begin{pmatrix} \delta\boldsymbol{r}_f \\ \delta\boldsymbol{r}_i \end{pmatrix}$$

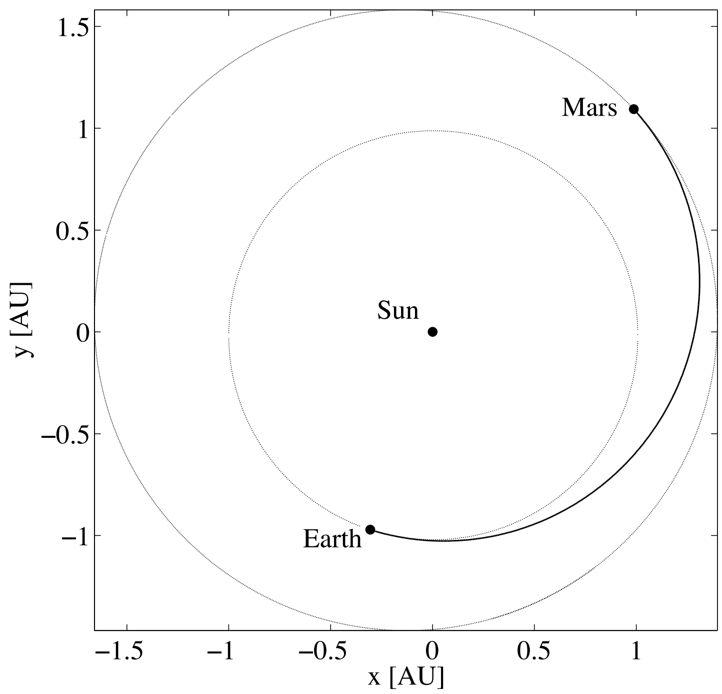
  - Impose $\delta\boldsymbol{r}_f = 0$ to obtain the Taylor expansion of the TPBVP solution w.r.t. $\delta\boldsymbol{r}_i$

$$\boxed{\delta\boldsymbol{v}_i = \mathcal{M}_{\delta\boldsymbol{r}_f=0}\left(\delta\boldsymbol{r}_i\right)}$$

  - Given any $\delta\boldsymbol{r}_i$, the previous map delivers the new solution

- Test case: Earth-Mars transfer (Mars Express)

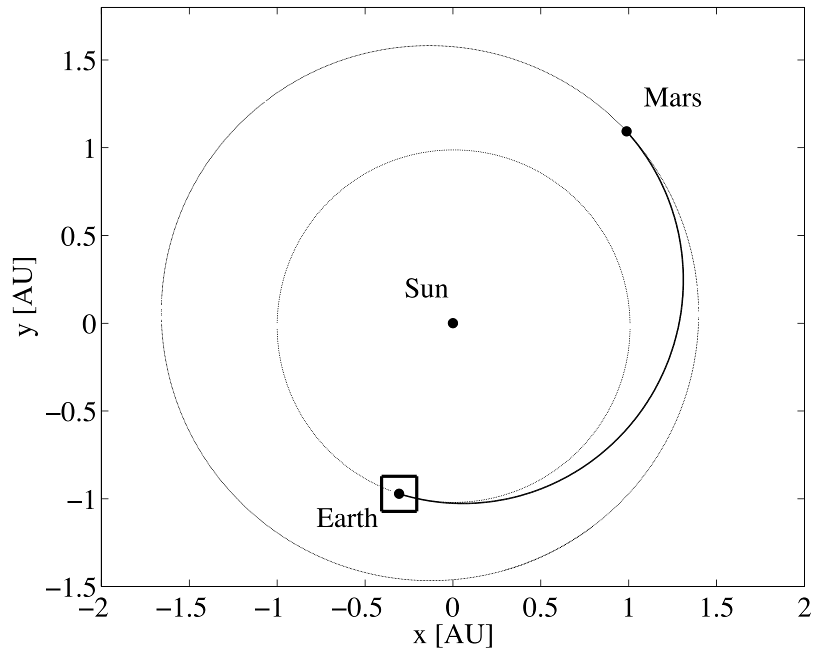
nominal



High Order Expansion of TPBVPs

- Test case: Earth-Mars transfer (Mars Express)

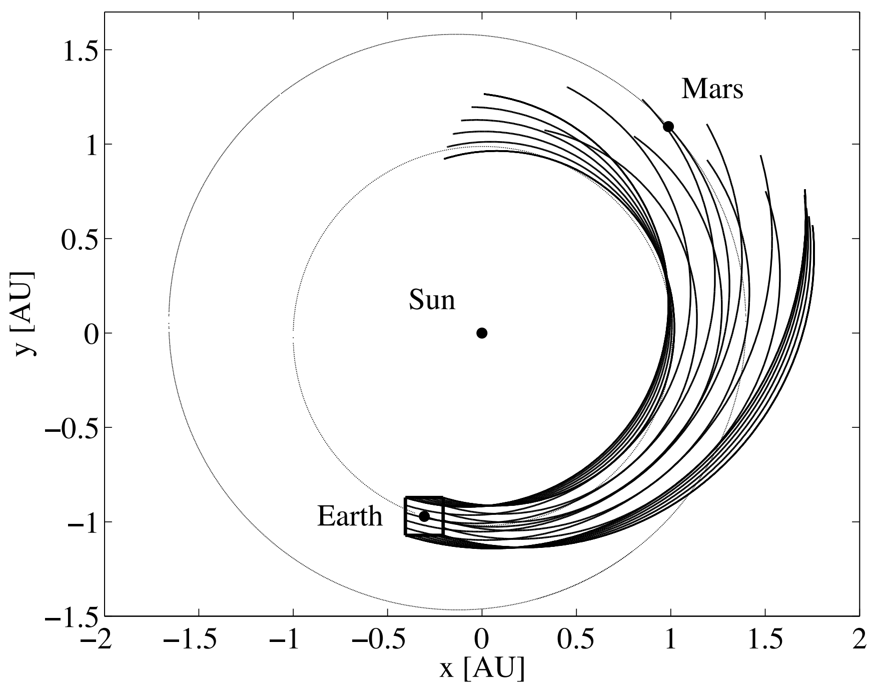
error box of 0.1 AU



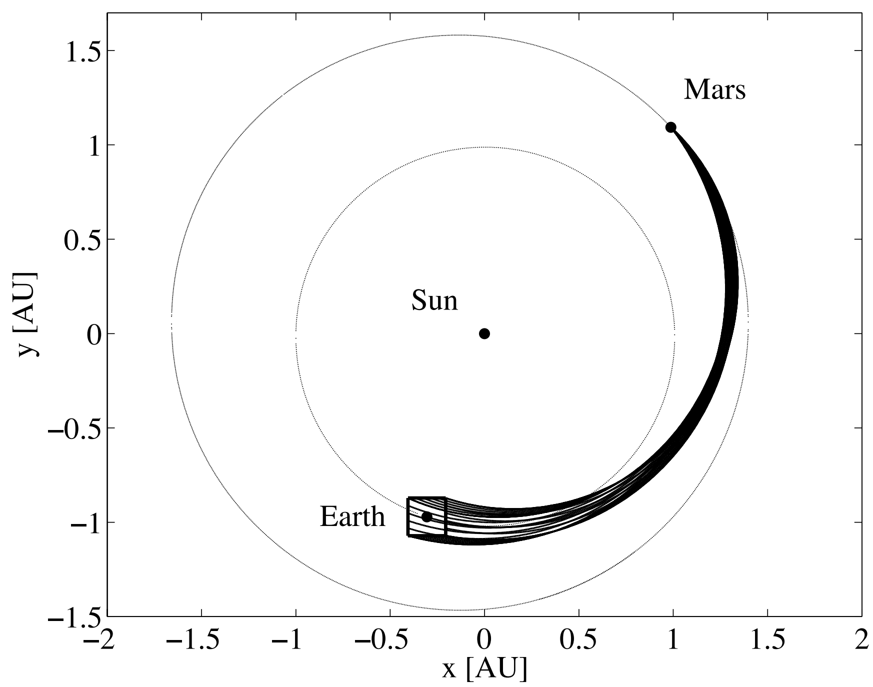
High Order Expansion of TPBVPs

- Test case: Earth-Mars transfer (Mars Express)

no corrections



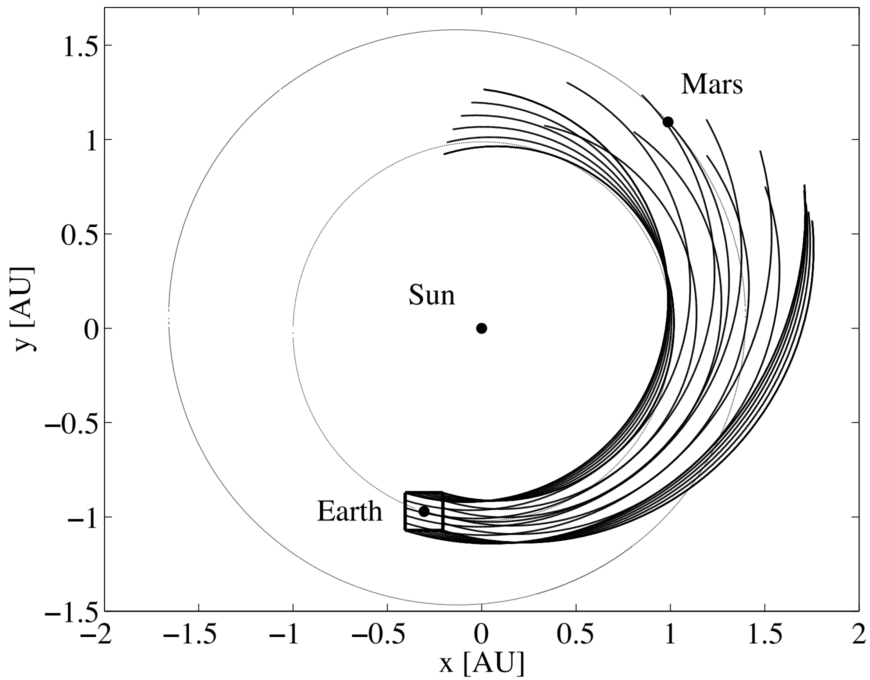
5-th order corrections



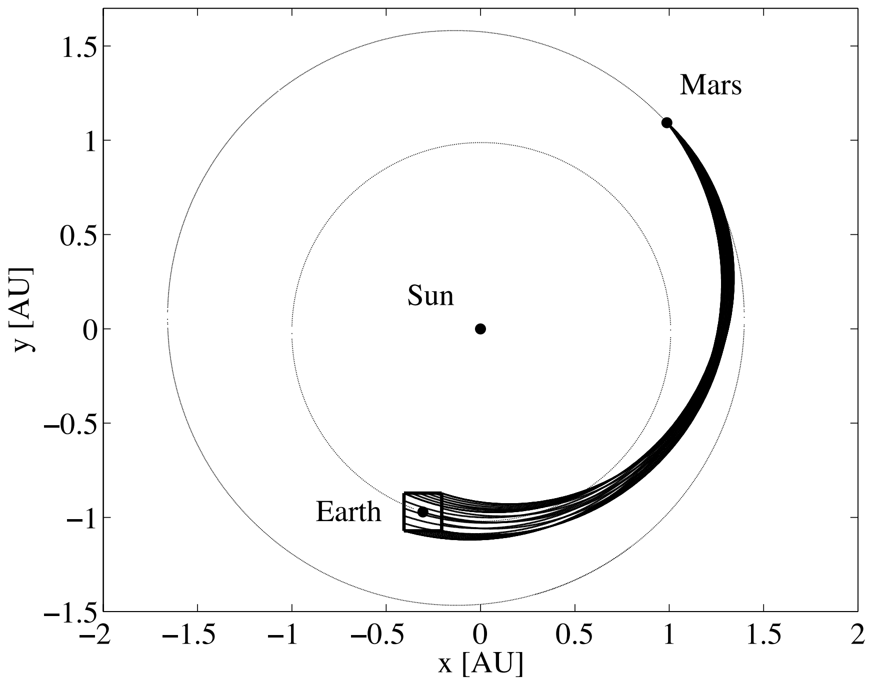
High Order Expansion of TPBVPs

- Test case: Earth-Mars transfer (Mars Express)

no corrections

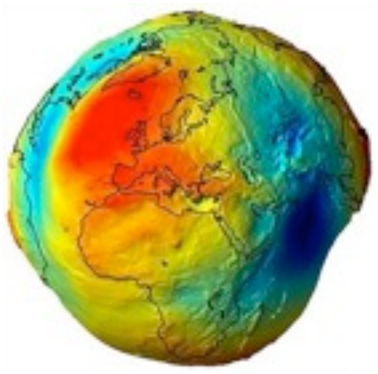


5-th order corrections



- The algorithm can be generalized to expand any TPBVP w.r.t. any parameter (e.g., final conditions $\delta \boldsymbol{r}_f$)

High Order Expansion of TPBVPs



Terra non sferica e massa non uniformemente distribuita

↓

Campo gravitazionale non sferico

Modello a Potenziale:

$$U = \frac{\mu}{r}\sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{R_T}{r}\right)^l P_{l,m}[\sin\varphi]\left\{C_{l,m}\cos m\lambda + S_{l,m}\sin m\lambda\right\}$$

$P_{l,m}$ → funzioni di Legendre

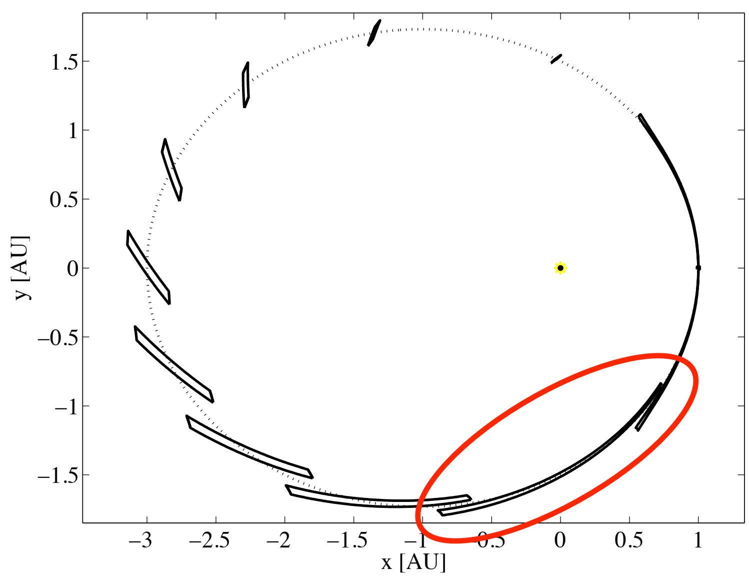
$C_{l,m}$, $S_{l,m}$ → coefficienti dell'espansione

Termini considerati → $l=m=3$

$$\mathrm{a}_g = \frac{\partial U(r,\lambda,\varphi)}{\partial r}\,\hat{r} + \frac{1}{r\cos\varphi}\frac{\partial U(r,\lambda,\varphi)}{\partial\lambda}\,\hat{\lambda} + \frac{1}{r}\frac{\partial U(r,\lambda,\varphi)}{\partial\varphi}\,\hat{\varphi}$$

High Order Sensitivity Analysis

- Example: 2-Body Problem
  - Eccentricity: 0.5 - Starting point: pericenter
  - Integration scheme: Runge-Kutta (variable step, order 8)
  - DA-based ODE flow expansion order: 5
- Uncertainty box on the initial position of 0.01 AU



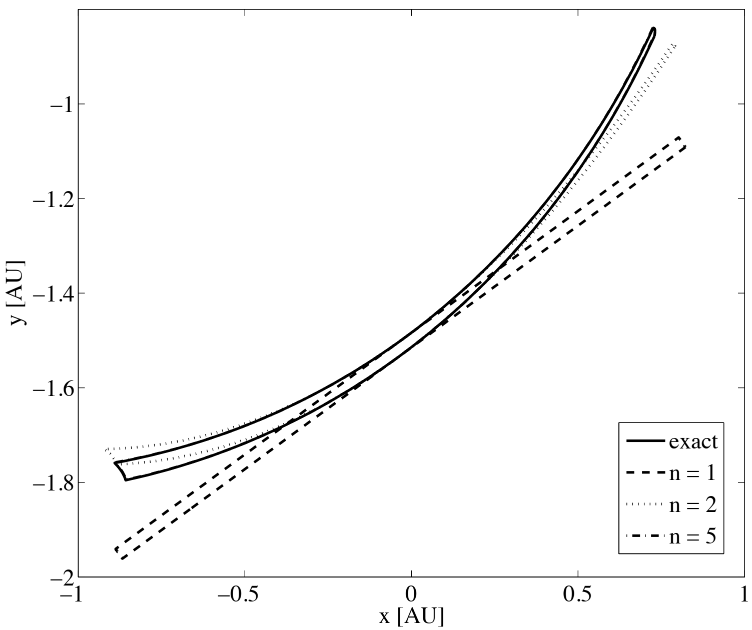
- Any sample in the uncertainty box can be propagated using the 5th order polynomial

Fast Monte Carlo simulations

High Order Sensitivity Analysis

- Example: 2-Body Problem
  - Eccentricity: 0.5 - Starting point: pericenter
  - Integration scheme: Runge-Kutta (variable step, order 8)
  - DA-based ODE flow expansion order: 5

- Uncertainty box on the initial position of 0.01 AU



- Any sample in the uncertainty box can be propagated using the 5th order polynomial

Fast Monte Carlo simulations