DINAMICA
INNOVATING TECHNOLOGY

An overview of optimal control methods with applications

F. Topputo and P. Di Lizia
Dinamica Srl

PEOPLE

MARIE CURIE ACTIONS

Registered office
Piazza della Repubblica, 10 - 20121 - Milan (Italy)
Head office
Via Morghen, 13 - 20158 - Milan (Italy)
Phone: +39-02-8342-2930
Fax: +39-02-3206-6679
e-mail: dinamica@dinamicatech.com
website: www.dinamicatech.com

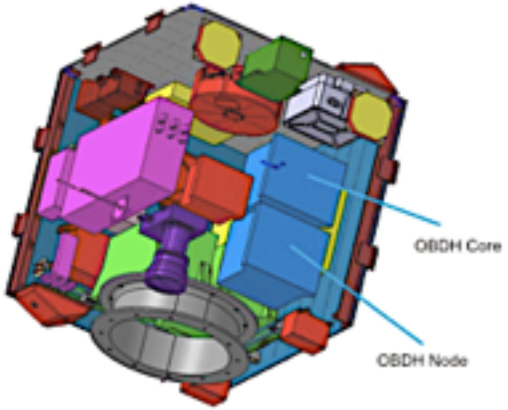
Stardust - Opening Training School
University of Strathclyde
Glasgow, UK

18-22 November 2013

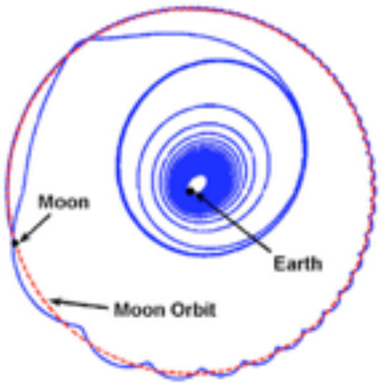
The Company

- Founded by 5 partners in January 2008
- All partners have a PhD in Aerospace Engineering
- Dinamica Srl has a strong connection with Academia
- More than 30 years of accumulated space experience
- Dinamica Srl is located in Milano

Keywords

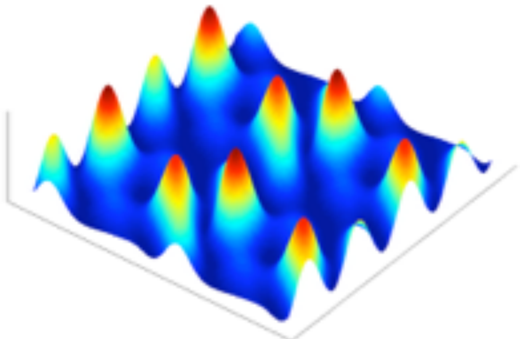


System engineering



Mission analysis

- Analysis
- Design
- Simulation
- Control
- Optimization

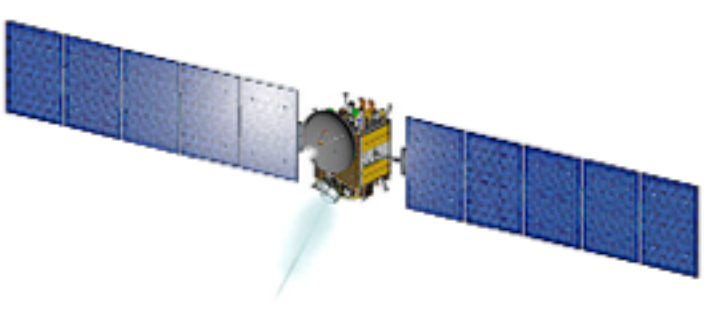


Optimization

Autonomy

The mission

- Italian SME, founded in 2008
- The mission: *"... to carry on developing methods and advanced solutions within the Space field and to transfer their use in other industrial sectors ..."*



Space field

Technology transfer

Industrial sector

A Tangible Example

Hubble

Pharmaceutical industry

System identification

Used to reconstruct unmodeled structural dynamics

Used to reconstruct a pharmaceutical process dynamics

Predictive control

Used to reduce vibrations in large telescope satellites

Used to minimize the energy supply

☑ Considerable savings (10-20%) compared to standard methods

Optimal Control Problem (1/2)

- Consider the following **dynamical system**:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$$

where: $\mathbf{x} = \{x_1, \ldots, x_n\}^T$ is the state vector and

$\mathbf{u} = \{u_1, \ldots, u_m\}^T$ is the control vector

- Determine the m control functions such that the following **performance index** is minimized:

$$J = \varphi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)dt$$

where the initial and final state vectors, $\mathbf{x}_0$ and $\mathbf{x}_f$, as well as the final time t_f , are not necessarily fixed

Optimal Control Problem (2/2)

- In addition to the previous statements suppose that the following **constraints** are imposed
  - Boundary conditions at final time t_f:

$$\psi(\mathbf{x}_f, t_f) = 0 \text{, where } \boldsymbol{\psi} = \{\psi_1, ..., \psi_p\}$$

  - Path constraints on the control variables:

$$\mathbf{C}(\mathbf{u}(t), t) \leq 0 \text{ , where } \mathbf{C} = \{C_1, ..., C_q\}$$

- Two classical solution methods:
  - **Indirect methods**: based on reducing the optimal control problem to a Boundary Value Problem (BVP)
  - **Direct methods**: based on reducing the optimal control problem to a nonlinear programming problem

Example: Low-Thrust Earth-Mars Transfer

- Given the dynamics of the controlled 2 body problem:

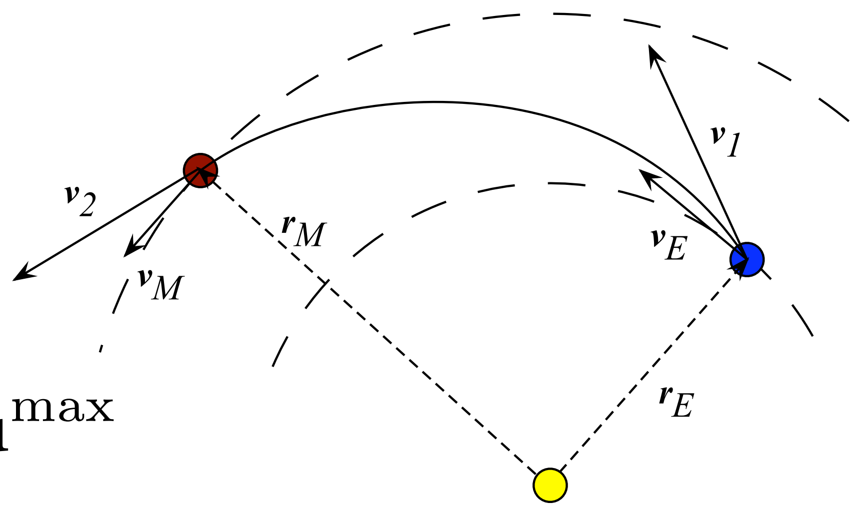
$$\ddot{\mathbf{r}} = -\frac{\mu}{\mathrm{r}^3} \cdot \mathbf{r} + \mathbf{u}$$

- Minimize: $J = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\mathrm{d}t = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u} \cdot \mathbf{u}^T \mathrm{d}t$

- Subject to equality constraints:

$$\mathbf{r}(t_0) = \mathbf{r}_E(t_0) \quad \mathbf{v}(t_0) = \mathbf{v}_E(t_0)$$
$$\mathbf{r}(t_f) = \mathbf{r}_M(t_f) \quad \mathbf{v}(t_f) = \mathbf{v}_M(t_f)$$

- and the inequality constraints $\mathbf{C}(\mathbf{u}(t), t) \leq 0$: $\|\mathbf{u}\| \leq \mathrm{u}^{\max}$



Indirect Methods (1/6)

- Reconsider the optimal control problem:

Given the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$

- Minimize: $J = \varphi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)dt$
- Subject to: $\boldsymbol{\psi}(\mathbf{x}_f, t_f) = 0$ and $\mathbf{C}(\mathbf{u}(t), t) \leq 0$

- Constraints are added to the performance index J by introducing two kinds of **Lagrange multipliers**:
  - a p-dimensional vector of constants $\boldsymbol{\nu}$ for the final constraints
  - two n- and a q-dimensional vectors of functions $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ (adjoint or costate variables) for dynamics and path constraints

- **Augmented performance index**:

$$\bar{J} = \varphi(\mathbf{x}_f, t_f) + \boldsymbol{\nu}^T \psi(\mathbf{x}_f, t_f) +$$
$$+ \int_{t_0}^{t_f} (L(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^T (\mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}}) + \boldsymbol{\mu}^T \mathbf{C}(\mathbf{u}(t), t)) dt$$

  - The dynamics is included in the augmented performance index as a constraint

- Moreover, pertaining the costate variables for the path inequality constraints $\boldsymbol{\mu}$, the generic component μ_k must satisfy the following relations:

$$\begin{cases} C_k(\mathbf{u}(t), t) < 0 & \Rightarrow \quad \mu_k(t) = 0 \\ C_k(\mathbf{u}(t), t) = 0 & \Rightarrow \quad \mu_k(t) \geq 0 \end{cases}$$

Indirect Methods (3/6)

- Integrating by parts the term $\boldsymbol{\lambda}^T \dot{\mathbf{x}}$ yields:

$$\bar{J} = \varphi(\mathbf{x}_f, t_f) + \boldsymbol{\nu}^T \psi(\mathbf{x}_f, t_f) - \boldsymbol{\lambda}_f^T \mathbf{x}_f + \boldsymbol{\lambda}_0^T \mathbf{x}_0 +$$

$$+ \int_{t_0}^{t_f} (L(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) + \dot{\boldsymbol{\lambda}}^T \mathbf{x} + \boldsymbol{\mu}^T \mathbf{C}(\mathbf{u}(t), t)) dt$$

  where $\boldsymbol{\lambda}_f = \boldsymbol{\lambda}(t_f)$ and $\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}(t_0)$

- The problem is then reduced to **identify a stationary point** of $\bar{J}$. This is achieved by imposing the gradient to be zero. The optimization variables are:
  - State vector $\mathbf{x}$ and control vector $\mathbf{u}$
  - Lagrange multipliers and costate variables $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$
  - Unknown components of the initial state $\mathbf{x}_0$, $i = \bar{k} + 1, \ldots, n$
  - Final state and time $\mathbf{x}_f$ and t_f

Indirect Methods (4/6)

Dynamics

$$\frac{\partial \bar{J}}{\partial \boldsymbol{\lambda}} = 0 \quad \Rightarrow \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$$

$$\frac{\partial \bar{J}}{\partial \mathbf{x}} = 0 \quad \Rightarrow \quad \dot{\boldsymbol{\lambda}} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} - \left(\frac{\partial L}{\partial \mathbf{x}}\right)^T$$

$$\frac{\partial \bar{J}}{\partial \mathbf{u}} = 0 \quad \Rightarrow \quad \left(\frac{\partial L}{\partial \mathbf{u}}\right)^T + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \mathbf{C}}{\partial \mathbf{u}}\right)^T \boldsymbol{\mu} = 0$$

Constraints

$$\frac{\partial \bar{J}}{\partial x_{i,0}} = 0 \quad \Rightarrow \quad \lambda_{i,0} = 0 \qquad (i = \bar{k} + 1, \ldots, n)$$

$$\frac{\partial \bar{J}}{\partial \mathbf{x}_f} = 0 \quad \Rightarrow \quad \boldsymbol{\lambda}_f = \left(\frac{\partial \varphi}{\partial \mathbf{x}_f}\right)^T + \left(\frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}_f}\right)^T \boldsymbol{\nu}$$

$$\frac{\partial \bar{J}}{\partial \boldsymbol{\nu}} = 0 \quad \Rightarrow \quad \boldsymbol{\psi}(\mathbf{x}_f, t_f) = 0$$

$$\frac{\partial \bar{J}}{\partial \boldsymbol{\mu}} \leq 0 \quad \Rightarrow \quad \mathbf{C}(\mathbf{u}(t), t) \leq 0$$

$$\frac{\partial \bar{J}}{\partial t_f} = 0 \quad \Rightarrow \quad \left(\frac{\partial \varphi}{\partial \mathbf{x}_f}\right)\mathbf{f}(\mathbf{x}_f, \mathbf{u}_f, t_f) + \frac{\partial \varphi}{\partial t_f} + \boldsymbol{\nu}^T \left(\left(\frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}_f}\right)\mathbf{f}(\mathbf{x}_f, \mathbf{u}_f, t_f) + \left(\frac{\partial \boldsymbol{\psi}}{\partial t_f}\right) \right)$$
$$+ L(\mathbf{x}_f, \mathbf{u}_f, t_f) + \boldsymbol{\mu}(t_f)^T \mathbf{C}(\mathbf{u}_f, t_f) = 0$$

Indirect Methods (5/6)

- The problem consists on finding the functions $\mathbf{x}(t)$, $\boldsymbol{\lambda}(t)$ and $\mathbf{u}(t)$ by solving the **differential-algebraic system**:

differential
$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \dot{\boldsymbol{\lambda}} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} - \left(\frac{\partial L}{\partial \mathbf{x}}\right)^T \end{cases}$$

algebraic
$$\left\{ \left(\frac{\partial L}{\partial \mathbf{u}}\right)^T + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \mathbf{C}}{\partial \mathbf{u}}\right)^T \boldsymbol{\mu} = 0 \right.$$

Euler-Lagrange equations

Note: For the sake of a more compact notation, define the Hamiltonian

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = L(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}(t)^T \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\mu}(t)^T \mathbf{C}(\mathbf{u}(t), t)$$

The previous equations read: $\dot{\mathbf{x}} = H_\lambda, \quad \dot{\boldsymbol{\lambda}} = -H_x, \quad H_u = 0$

Indirect Methods (6/6)

- The previous differential-algebraic system must be coupled to the $2n$ **boundary conditions**

$$\begin{cases} x_{i,0} \ \text{given or} \ \lambda_{i,0} = 0 \qquad i = 1, ..., n \\ \boldsymbol{\lambda}_f = \left(\frac{\partial \varphi}{\partial \mathbf{x}_f}\right)^T + \left(\frac{\partial \psi}{\partial \mathbf{x}_f}\right)^T \boldsymbol{\nu} \end{cases}$$

and to the $p+q+1$ **additional constraints**

$$\begin{cases} \psi(\mathbf{x}_f, t_f) = 0 \\ \mathbf{C}(\mathbf{u}(t), t) \leq 0 \\ \left(\frac{\partial \varphi}{\partial \mathbf{x}_f}\right)\mathbf{f}_f + \frac{\partial \varphi}{\partial t_f} + \boldsymbol{\nu}^T \left(\left(\frac{\partial \psi}{\partial \mathbf{x}_f}\right)\mathbf{f}_f + \left(\frac{\partial \psi}{\partial t_f}\right) \right) + L_f + \boldsymbol{\mu}_f^T \mathbf{C}_f = 0 \end{cases}$$

➡ The optimal control problem is reduced to a **boundary value problem on a differential-algebraic system** of equations (DAE)

Assignment #1

- Given the simple optimal control problem (Problem #1)

$$\dot{x}_1 = 0.5x_1 + u$$
$$\dot{x}_2 = u^2 + x_1 u + \frac{5}{4}x_1^2$$

Dynamics

$$J = x_2(1)$$

Obj. fcn.

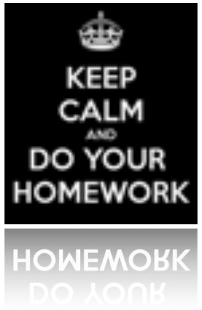
$$x_1(0) = 1$$
$$x_2(0) = 0$$

b. c.

$$t_i = 0$$
$$t_f = 1$$

init., final time

- Write the necessary conditions for optimality and show that the optimal solution is



$$x_1(t) = \frac{\cosh(1-t)}{\cosh(1)}$$
$$u(t) = \frac{-(\tanh(1-t) + 0.5)\cosh(1-t)}{\cosh(1)}$$

Optimal solution

Low-Thrust Transfer to Halo Orbit (1/4)

- Transfer the s/c from a given orbit (GTO raising) to a Halo orbit around L1 of the Earth-Moon system

- Dynamics:

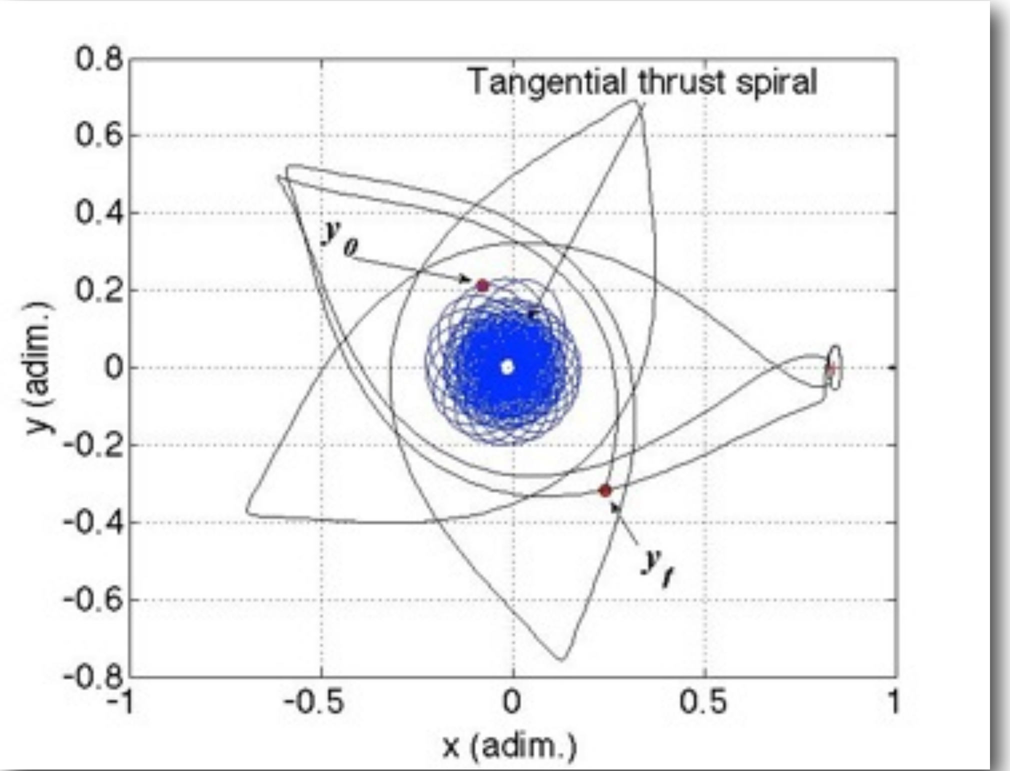
$$\ddot{x} - 2\dot{y} = \frac{\partial \Omega_3}{\partial x} + u_1$$

$$\ddot{y} + 2\dot{x} = \frac{\partial \Omega_3}{\partial y} + u_2$$

$$\ddot{z} = \frac{\partial \Omega_3}{\partial z} + u_3$$

$$\mathbf{x} = \{x, y, z, \dot{x}, \dot{y}, \dot{z}\}^T$$

$$\mathbf{u} = \{u_1, u_2, u_3\}^T$$



Low-Thrust Transfer to Halo Orbit (2/4)

- In canonical form, the dynamics reads: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$
- Performance index: minimize the quadratic functional

$$J = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\mathrm{d}t = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u} \cdot \mathbf{u}^T \mathrm{d}t$$

- Constraints: fixed $\mathbf{x}_0$ and $\mathbf{x}_f$, fixed final time t_f
- Euler-Lagrange equations:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \dot{\boldsymbol{\lambda}} &= -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} - \left(\frac{\partial L}{\partial \mathbf{x}}\right)^T \\ \left(\frac{\partial L}{\partial \mathbf{u}}\right)^T &+ \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \mathbf{C}}{\partial \mathbf{u}}\right)^T \boldsymbol{\mu} = 0 \end{aligned}$$

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \dot{\boldsymbol{\lambda}} &= -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\lambda} \\ 0 &= \frac{\partial H}{\partial \mathbf{u}} \end{aligned}$$

Low-Thrust Transfer to Halo Orbit (3/4)

- Processing the last algebraic equation leads to:

$$\left(\frac{\partial L}{\partial \mathbf{u}}\right)^T + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \mathbf{C}}{\partial \mathbf{u}}\right)^T \boldsymbol{\mu} = 0 \;\blacktriangleright\; u_i = -\lambda_{3+i}, \;\; i = 1, \ldots, 3$$

which can be inserted in the differential equations

The **DAE** system is reduced to a **ODE** system

- All constraints simply reduce to:

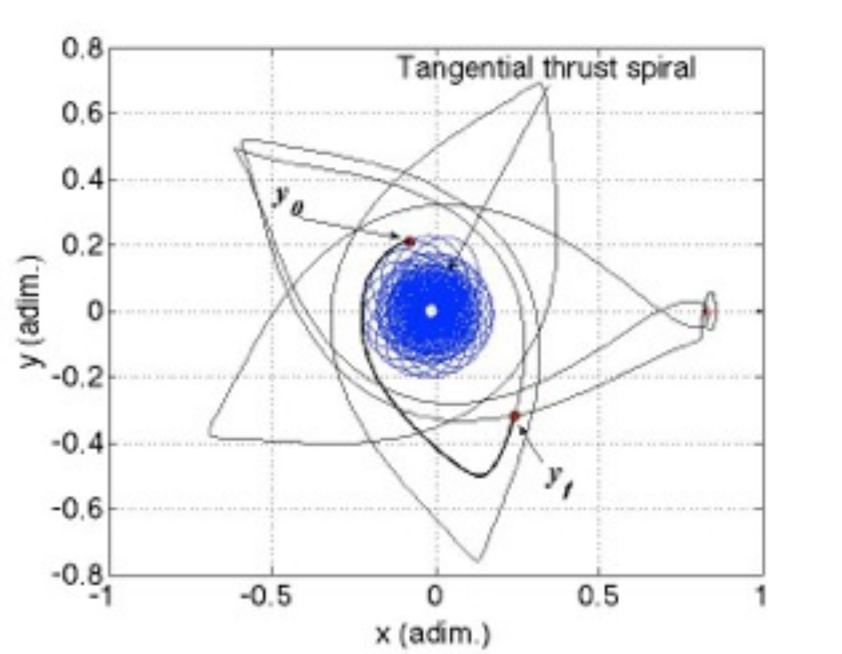
$$\begin{aligned} \mathbf{x}(t_0) - \mathbf{x}_0 &= 0 \\ \mathbf{x}(t_f) - \mathbf{x}_f &= 0 \end{aligned}$$

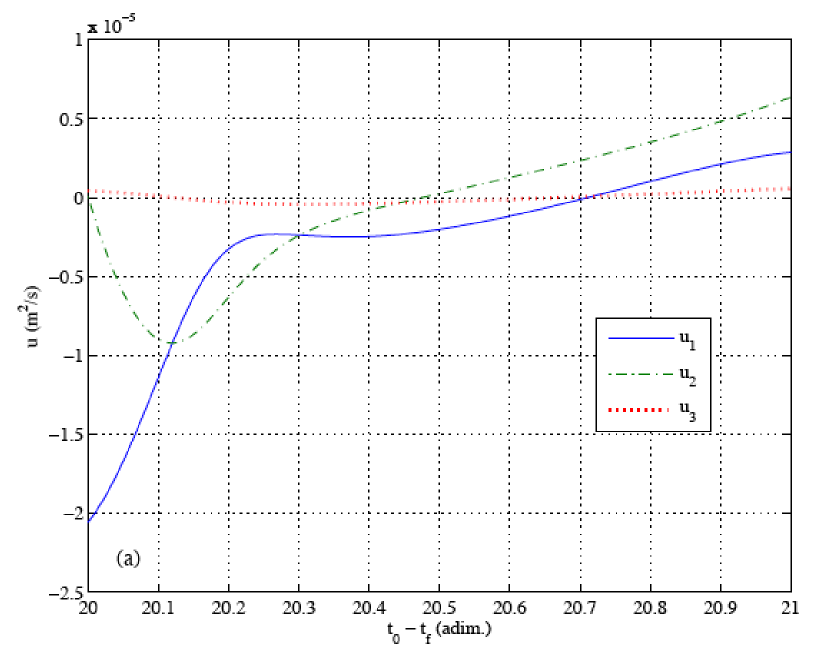
The original problem is reduced to a “simple” **Two Point Boundary Value Problem** (TPBVP)

DINAMICA INNOVATING TECHNOLOGY

Low-Thrust Transfer to Halo Orbit (4/4)

- Solution of the TPBVP:
  - Transcribe the dynamics $(\mathbf{x}, \lambda)$ ▶ $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \ldots, \mathbf{x}_N, \boldsymbol{\lambda}_N)$
  - Couple the transcribed dynamics with the constraints on $\mathbf{x}_0$ and $\mathbf{x}_f$
  - Solve the resulting system with a Newton method **starting from a suitable initial condition**
  - Evaluate the control parameters $(\mathbf{u}_0, \ldots, \mathbf{u}_N)$





DINAMICA
INNOVATING TECHNOLOGY

End-to-end optimization w/ finite thrust

- GTO-to-halo fully optimized
  - very difficult problem
  - tens of spirals
  - thrust saturation

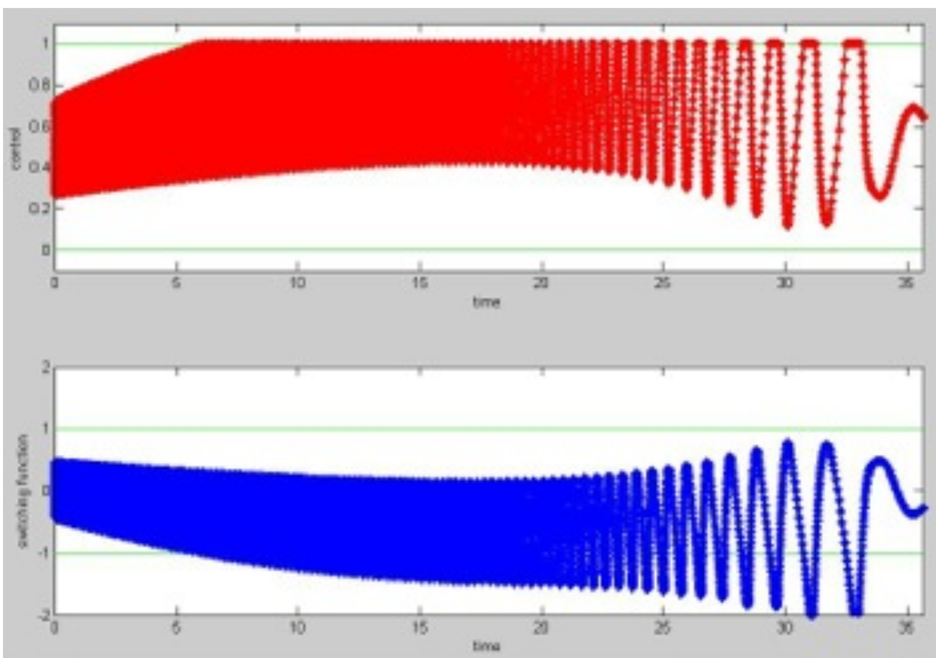
$$\ddot{x} - 2\dot{y} = \frac{\partial \Omega_3}{\partial x} + \frac{T_x}{m}, \qquad \ddot{y} + 2\dot{x} = \frac{\partial \Omega_3}{\partial y} + \frac{T_y}{m}$$

$$\ddot{z} = \frac{\partial \Omega_3}{\partial z} + \frac{T_z}{m}, \qquad \dot{m} = -\frac{T}{I_{sp} g_0}$$

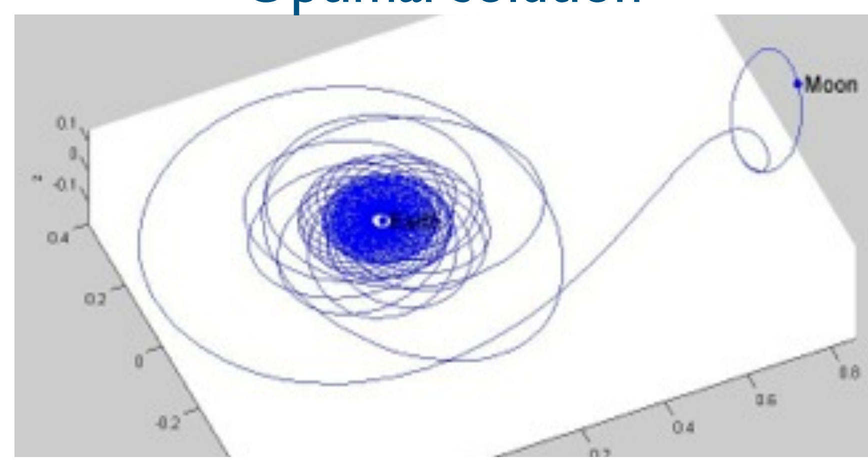
$$J = \int_{t_0}^{t_f} \frac{T(\tau)}{I_{sp} g_0} \, d\tau$$

Control

Switching fcn



Optimal solution



Assignment #2

- Given Problem #1

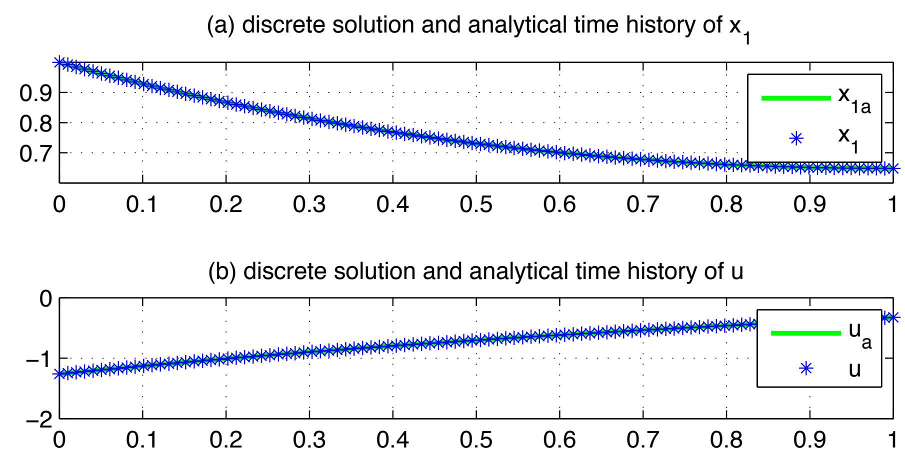
$$\dot{x}_1 = 0.5x_1 + u$$
$$\dot{x}_2 = u^2 + x_1 u + \frac{5}{4}x_1^2$$

$$J = x_2(1)$$

$$x_1(0) = 1 \qquad x_2(0) = 0$$

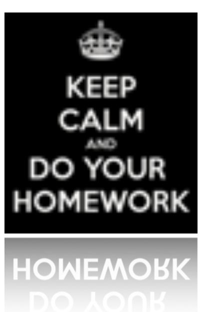
$$t_i = 0 \qquad t_f = 1$$

Dynamics — Obj. fcn. — b. c. — init., final time



- Solve the TPBVP associated
  - Matlab built-in bvp4c, bvp5c
  - Sixth-order method bvp_h6

  available at http://www.astrodynamics.eu/Astrodynamics.eu/Software.html



Indirect Methods: Remarks

- **Main Difficulties**:
  - Deriving Euler-Lagrange equations and transversality conditions for the problem at hand
  - Nonlinearity of the dynamics
  - Solution of the DAE system itself
  - Solution of the boundary value problem on the DAE system
  - Lack of a plain physical meaning of Lagrange multipliers
    - ▶ difficulty at identifying good first guesses for Lagrange multipliers (primer vector theory)

Optimal Control Problem

- Given a dynamical system: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$
- Determine $\mathbf{u}(t)$ which minimize the performance index:

$$J = \varphi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)dt$$

- and satisfy the constraints: $\psi(\mathbf{x}_f, t_f) = 0 \quad \mathbf{C}(\mathbf{u}(t), t) \leq 0$
- Two classical solution methods:
  - **Indirect methods**: based on reducing the optimal control problem to a Boundary Value Problem (BVP)
  - **Direct methods**: based on reducing the optimal control problem to a nonlinear programming problem

Nonlinear Programming Problem

- Generally constrained optimization problem

Given a function $f(\boldsymbol{x}) = f(x_1, x_2, ..., x_v)$

- **Minimize**:

$$f(\boldsymbol{x})$$

- **Subject to** K **equality** constraints:

$$c_k(\boldsymbol{x}) = 0, \quad k = 1, ..., K \qquad (K \leq v)$$

and J **inequality** constraints:

$$g_j(\boldsymbol{x}) \geq 0, \quad j = 1, ..., J$$

where J can exceed v

Unconstrained Optimization (1/6)

- Minimize: $f(\boldsymbol{x})$
- The **necessary condition** for the identification of the optimum is:

$$\nabla_{\boldsymbol{x}} f = 0$$

The optimization problem in the v variables $\boldsymbol{x}$ is reduced to the solution of a system of v nonlinear equations

Note: given the Hessian of f , $\mathbf{H}_f$, a sufficient condition is:

$$\boldsymbol{x}\,\mathbf{H}_f\,\boldsymbol{x} > 0, \quad \forall \boldsymbol{x}$$

- The solution can be found using the **Newton method**

Newton Method (1/3)

- Consider the problem:

$$F(x) = 0$$

- The Newton method is an iterative method based on a **linearization** of F around the current iterate

1. Select an initial guess $\hat{x}$
2. Consider the first order approximation of F

$$F(x) \approx F(\hat{x}) + F'(\hat{x}) \cdot (x - \hat{x}) = 0$$

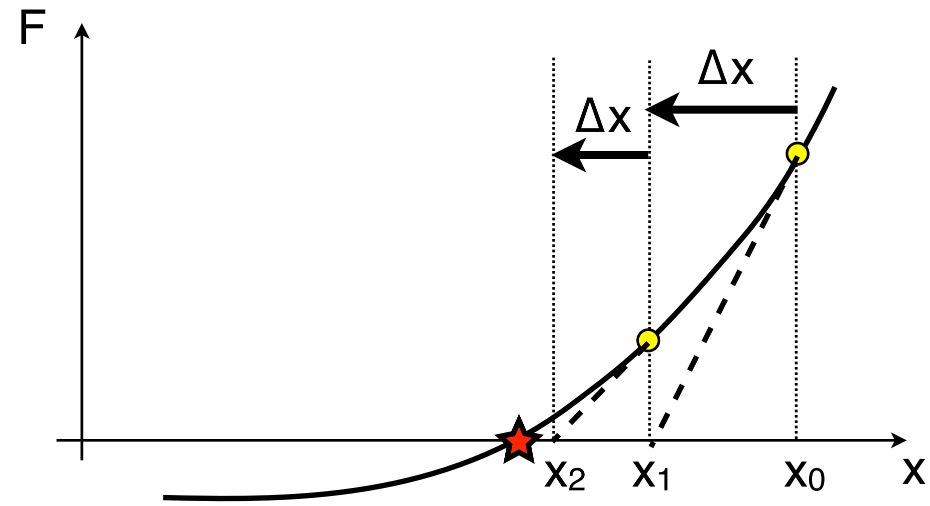
3. Find the correction:

$$\Delta x = (x - \hat{x}) = -[F'(\hat{x})]^{-1} \cdot F(\hat{x})$$

4. Update current iterate and repeat from 2 until convergence

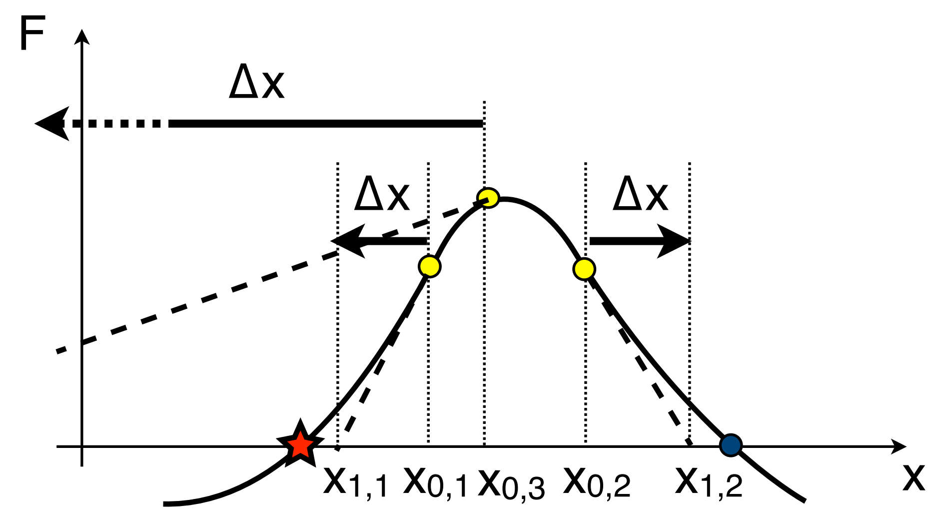
Newton Method (2/3)

- Graphical interpretation:



- Since it is based on a first order approximation of F the method is “local”

Different first guesses might lead to different solutions



Newton Method (3/3)

- Classical methods to stabilize the iteration
  - **Line Search**:

    Instead of updating the current iterate using

    $$\hat{x}_{\text{new}} = \hat{x} + \Delta x$$

    Reduce the step size using a parameter α :

    $$\hat{x}_{\text{new}} = \hat{x} + \alpha \, \Delta x$$

    where α is chosen such that

    $$||F(\hat{x}_{\text{new}})|| \; \leq \; ||F(\hat{x})||$$

  - **Trust region**:

    The direction of the computed $\Delta \boldsymbol{x}$ is slightly modified

Unconstrained Optimization (2/5)

- Solve: $\nabla_x f = 0$

Newton algorithm:

- Select an initial guess $\boldsymbol{x}$
- While stopping criterion is not satisfied
  - Find the corrections $\Delta\boldsymbol{x}$ to the current solution by solving the linear system

$$\mathbf{H}_f \, \Delta\boldsymbol{x} = -\nabla_x f$$

where $\mathbf{H}_f$ is the Hessian of f : $\mathbf{H}_f = \nabla_x^2 f$

  - Update the current solution: $\boldsymbol{x}$ ► $\boldsymbol{x} + \Delta\boldsymbol{x}$

Unconstrained Optimization (3/5)

Important note:

- Consider the following optimization problem
  - Minimize the quadratic form:

$$\frac{1}{2}\Delta \boldsymbol{x}^T \mathbf{H}_f \Delta \boldsymbol{x} + \nabla_x f^T \Delta \boldsymbol{x}$$

  - Necessary optimality conditions:

$$\mathbf{H}_f \Delta \boldsymbol{x} + \nabla_x f = 0$$

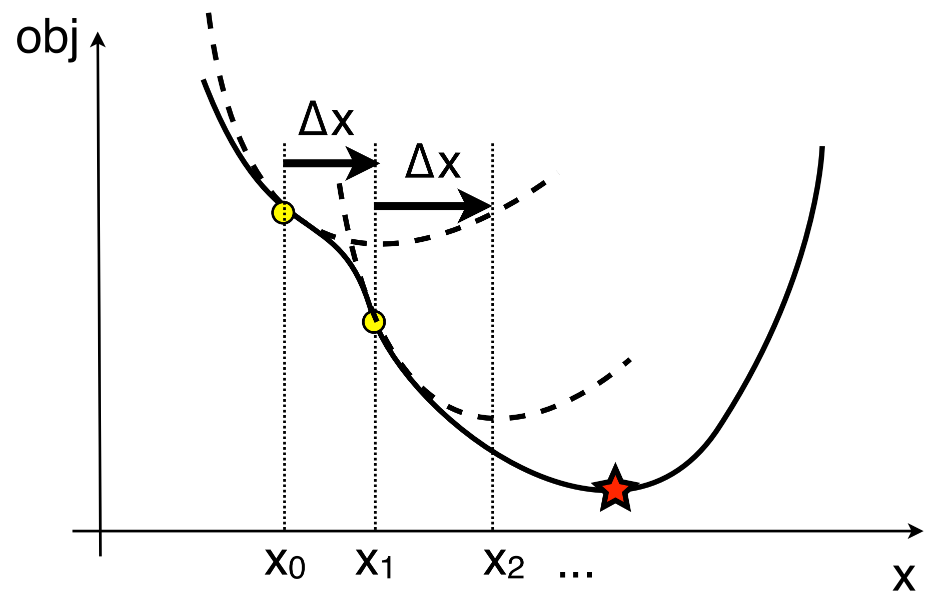
which can be written as:

$$\mathbf{H}_f \Delta \boldsymbol{x} = -\nabla_x f$$

Finding the corrections $\Delta \boldsymbol{x}$, i.e. the search direction, in the original optimization problem is equivalent to minimizing the previous quadratic form

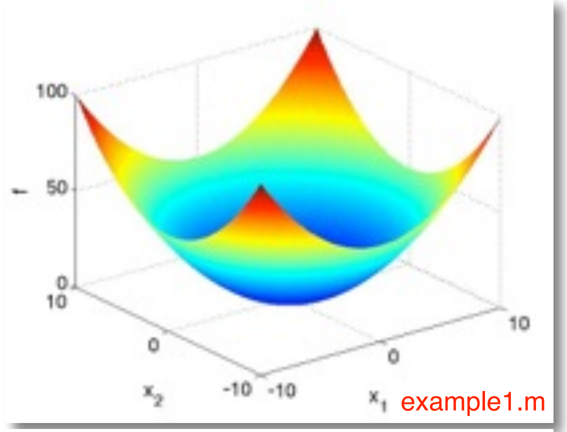
Unconstrained Optimization (4/5)

- Given the function to be minimized, $f(x)$, each iteration of the Newton method is equivalent to:
  - Approximate f around the current solution x with a quadratic form
  - Find the offset, Δx, to the zero-gradient point of the quadratic form
  - Use Δx as a correction in the original optimization problem

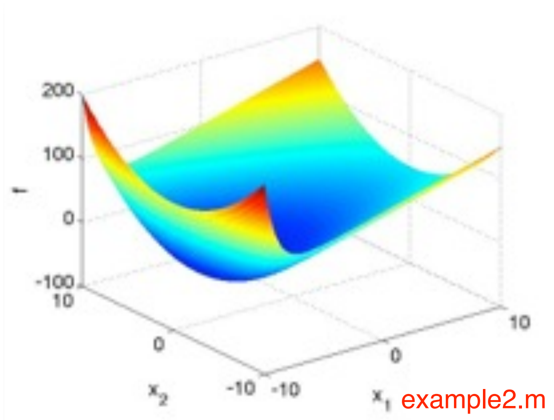


Unconstrained Optimization (5/5)

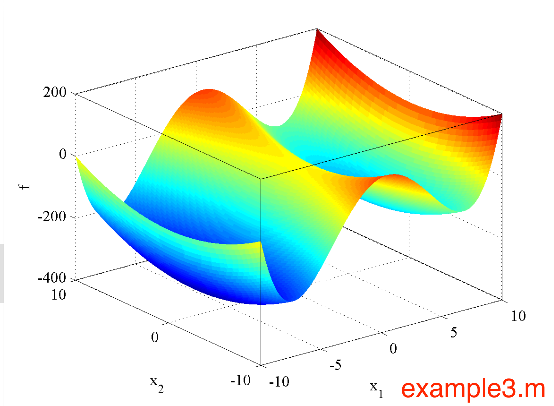
$$f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$$



$$f(x_1, x_2) = 5\,x_1 + e^{-(x_1+5)} + x_2^2$$



$$f(x_1, x_2) = \frac{x_1^4}{10} - 10x_1^2 + 10x_1 + x_2^2$$



Assignment #3

- Numerically re-compute the three unconstrained optimizations in the previous slide; i.e.,

$$f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$$

$$f(x_1, x_2) = 5\,x_1 + e^{-(x_1+5)} + x_2^2$$

$$f(x_1, x_2) = \frac{x_1^4}{10} - 10x_1^2 + 10x_1 + x_2^2$$

- Advice
  - Use Matlab built-in fminunc
  - Code a SQP solver

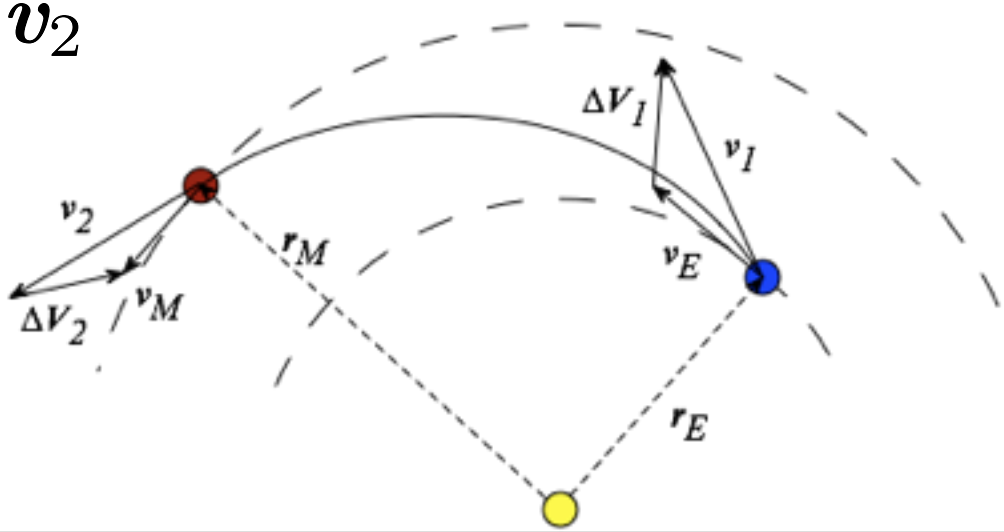
Earth-Mars 2-impulse Transfer (1/3)

- Optimization variables: departure date t_0 and time of flight t_{tof}
- Compute the positions of the starting and arrival planets through the ephemerides evaluation:

$$(\boldsymbol{r}_E, \boldsymbol{v}_E) = eph(t_0, Earth), \; (\boldsymbol{r}_M, \boldsymbol{v}_M) = eph(t_0 + t_{tof}, Mars)$$

- Solve the Lambert's problem to evaluate the escape velocity $\boldsymbol{v}_1$ and the arrival one $\boldsymbol{v}_2$
- Objective function:

$$\Delta V = \Delta V_1 + \Delta V_2$$



Earth-Mars 2-impulse Transfer (2/3)

- Minimize: $f(\boldsymbol{x}) = \Delta V(\boldsymbol{x}) = \Delta V(t_0, t_{tof})$
- Necessary conditions for optimality: $\nabla_{\boldsymbol{x}} f = 0$

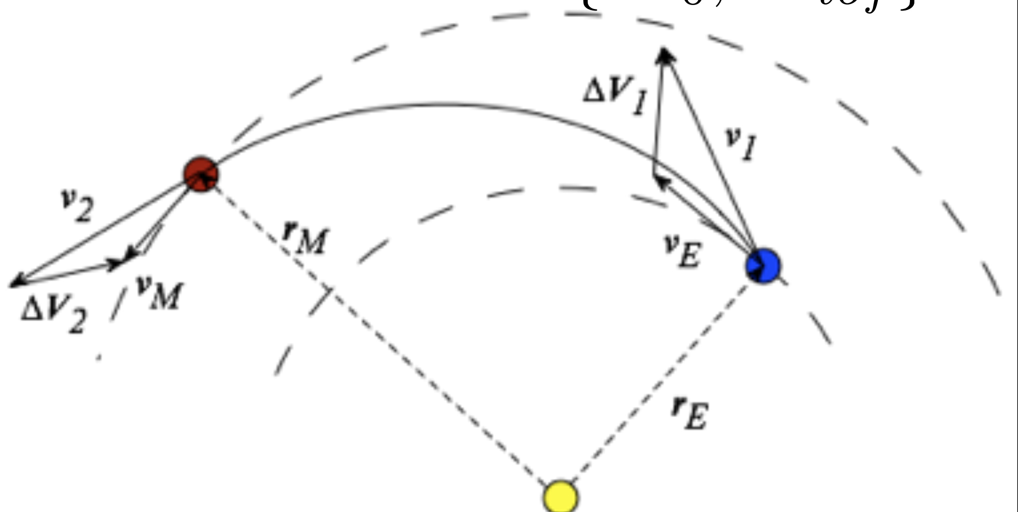
$$\frac{\partial \Delta V}{\partial t_0} = 0 \qquad \frac{\partial \Delta V}{\partial t_{tof}} = 0$$

- In a generic iteration, given the current estimate $\boldsymbol{x} = \{t_0, t_{tof}\}$, evaluate the corrections $\Delta \boldsymbol{x} = \{\Delta t_0, \Delta t_{tof}\}$:

$$\mathbf{H}_f \, \Delta \boldsymbol{x} = -\nabla_{\boldsymbol{x}} f$$

where $\mathbf{H}_f$ is:

$$\begin{bmatrix} \frac{\partial^2 \Delta V}{\partial^2 t_0} & \frac{\partial^2 \Delta V}{\partial t_0 \partial t_{tof}} \\ \frac{\partial^2 \Delta V}{\partial t_{tof} \partial t_0} & \frac{\partial^2 \Delta V}{\partial^2 t_{tof}} \end{bmatrix}$$



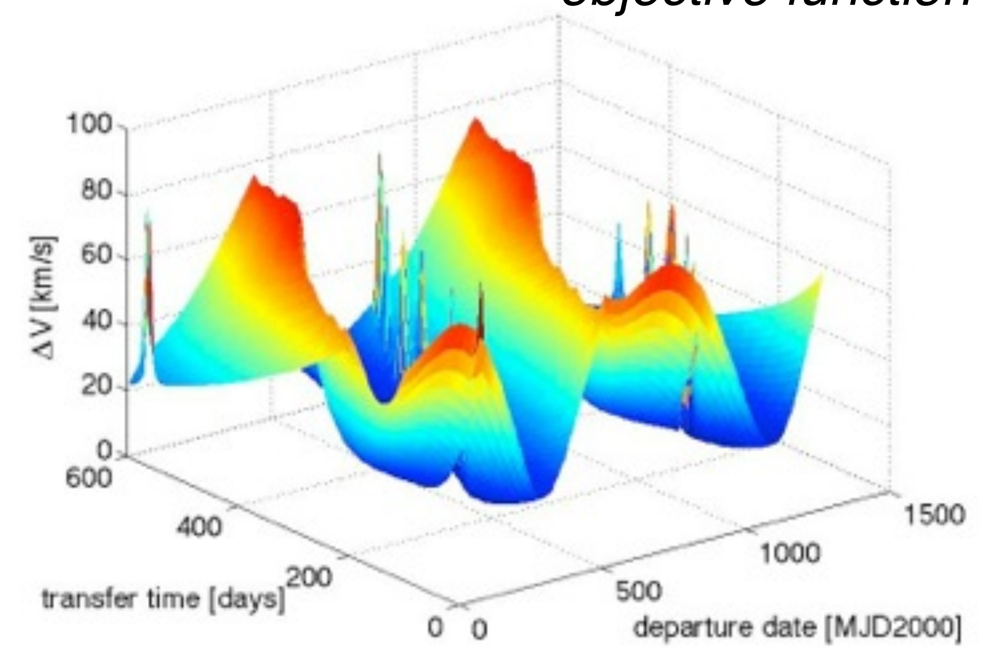
Earth-Mars 2-impulse Transfer (3/3)

- Search space:

$$t_0 \in [0, 1460]\ MJD2000 \ \cong\ \text{4 years}$$

$$t_{tof} \in [100, 600]\ day$$

objective function



exampleEM.m

Equality Constrained Optimization (1/5)

- Minimize: $f(\boldsymbol{x})$

  Subject to: $c_k(\boldsymbol{x}) = 0, \quad k = 1, ..., K \qquad (K \leq \upsilon)$

- The classical approach to the solution of the previous problem is based on the method of Lagrange multipliers

Method of Lagrange multipliers:

- Introduce the Lagrange function:

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda^T \cdot \boldsymbol{c}(\boldsymbol{x})$$

where L is a function of the υ variables $\boldsymbol{x}$ and the K Lagrange multipliers λ

Equality Constrained Optimization (2/5)

- The necessary conditions for the identification of the optimum are:

$$\nabla_x L(\boldsymbol{x}, \lambda) = \nabla_x f(\boldsymbol{x}) - \mathbf{C}^T(\boldsymbol{x}) \cdot \lambda = 0$$
$$\nabla_\lambda L(\boldsymbol{x}, \lambda) = \boldsymbol{c}(\boldsymbol{x}) = 0$$

where $\mathbf{C}(\boldsymbol{x})$ is the Jacobian of $\boldsymbol{c}(\boldsymbol{x})$

The constrained optimization problem in the v variables $\boldsymbol{x}$ has been reduced to the solution of a system of $v + K$ equations in the $v + K$ variables $(\boldsymbol{x}, \lambda)$

- Solution by Newton method

Equality Constrained Optimization (3/5)

Algorithm:

- Select an initial guess $(\boldsymbol{x}, \lambda)$
- While stopping criterion is not satisfied
  - Find the corrections $(\Delta \boldsymbol{x}, \Delta \lambda)$ to the current solution by solving the linear system

$$\begin{bmatrix} \mathbf{H}_L & -\mathbf{C}^T \\ \mathbf{C} & 0 \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{x} \\ \Delta \lambda \end{Bmatrix} = \begin{Bmatrix} -\nabla_x f \\ -\boldsymbol{c} \end{Bmatrix}$$

Karush-Kuhn-Tucker (KKT)

where $\mathbf{H}_L = \nabla^2_{\boldsymbol{xx}} f - \sum_{k=1}^{K} \lambda_k \nabla^2_{\boldsymbol{xx}} c_k$

  - Update the current solution: $(\boldsymbol{x}, \lambda)$ ► $(\boldsymbol{x} + \Delta \boldsymbol{x}, \lambda + \Delta \lambda)$

Equality Constrained Optimization (4/5)

Important note:

- Consider the following optimization problem:
  - Minimize the quadratic form:

$$\frac{1}{2}\Delta\boldsymbol{x}^T\,\mathbf{H}_L\Delta\boldsymbol{x}+(\nabla_{\boldsymbol{x}}f)^T\Delta\boldsymbol{x}$$

  - Subject to the linear constraints:

$$\mathbf{C}\Delta\boldsymbol{x}=-\boldsymbol{c}$$

- Use the approach of Lagrange multipliers
  - Lagrange function:

$$\frac{1}{2}\Delta\boldsymbol{x}^T\,\mathbf{H}_L\Delta\boldsymbol{x}+(\nabla_{\boldsymbol{x}}f)^T\Delta\boldsymbol{x}-\lambda^T\cdot(\mathbf{C}\Delta\boldsymbol{x}+\boldsymbol{c})$$

Equality Constrained Optimization (5/5)

- Necessary optimality conditions:

$$\mathbf{H}_L \Delta \boldsymbol{x} + \nabla_x f - \mathbf{C}^T \cdot \lambda = 0$$

$$\mathbf{C} \Delta \boldsymbol{x} + \boldsymbol{c} = 0$$

which can be written as:

$$\begin{bmatrix} \mathbf{H}_L & -\mathbf{C}^T \\ \mathbf{C} & 0 \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{x} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} -\nabla_x f \\ -\boldsymbol{c} \end{Bmatrix}$$

Karush-Kuhn-Tucker (KKT)

Finding the corrections $(\Delta \boldsymbol{x}, \Delta\lambda)$, i.e. the search direction, in the original optimization problem using the KKT system is equivalent to minimizing the previous quadratic form

Inequality Constrained Optimization

- Minimize: $f(\boldsymbol{x})$

  Subject to: $g_j(\boldsymbol{x}) \geq 0, \quad j = 1, ..., J$

A possible approach

- **Interior Point Method**:

  The inequality constraints are added to the objective function as a **penalty term**

$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{j=1}^{J} \mu_j \, e^{-g_j(\boldsymbol{x})}$$

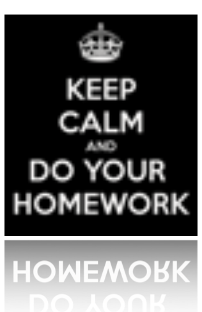
Solution is forced to move into the set of feasible solutions by means of the **barrier function** e^{-g_j}

Assignment #4

- Solve the NLP problem(*)

$$\begin{aligned} \min_{\mathbf{x}\in\mathbf{R}^3} \ & f(\mathbf{x}) := x_1^2 + x_1x_2 + 2x_2^2 - 6x_1 - 2x_2 - 12x_3 \\ \text{s.t.} \ & g_1(\mathbf{x}) := 2x_1^2 + x_2^2 \leq 15 \\ & g_2(\mathbf{x}) := x_1 - 2x_2 - x_3 \geq -3 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

- Using Matlab built-in fmincon
  - Use SQP algorithm, quasi-Newton update, and line search
  - Set solution tol, function tol, constraint tol to 1e-7
  - Specify initial guess to x_0 = (1, 1, 1)
  - Make sure g_2 is treated as linear constraint by fmincon
  - Solve w/o providing gradient of obj fcn and constraints; then re-do by providing analytic gradients
  - Repeat optimization from a different x_0; do we find the same optimal solution found previously? Why?



(*) B. Chachuat, Nonlinear and Dynamic Optimization, EPFL

Direct Methods

- Direct methods are based on reducing the optimal control problem to a nonlinear programming problem
- The core of the reduction of the optimal control problem to a nonlinear programming problem is:
  - The **parameterization** of all continuous variables
  - The **transcription** of the differential equations describing the dynamics, into a finite set of equality constraints

    Classical transcription methods:

    - **Collocation**
    - **Multiple Shooting**

The original optimal control problem is solved within the accuracy of the parameterization and the transcription method used

Parameterization

- The parameterization is based on the discretization of the continuous variables on a mesh, typically settled up on the time domain
  - Discretize the time domain as:

$$t_0 = t_1 < t_2 < \ldots < t_N = t_f$$

  - Discretize the states and the controls over the previous mesh by defining $\mathbf{x}_k = \mathbf{x}(t_k)$ and $\mathbf{u}_k = \mathbf{u}(t_k)$

$$\mathbf{x}(t) \;\blacktriangleright\; \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$$

$$\mathbf{u}(t) \;\blacktriangleright\; \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$$

  - Consequently, a new vector of variables can be defined:

$$\mathbf{X} = \{t_f, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_N, \mathbf{u}_N\}$$

Transcription: Collocation (1/2)

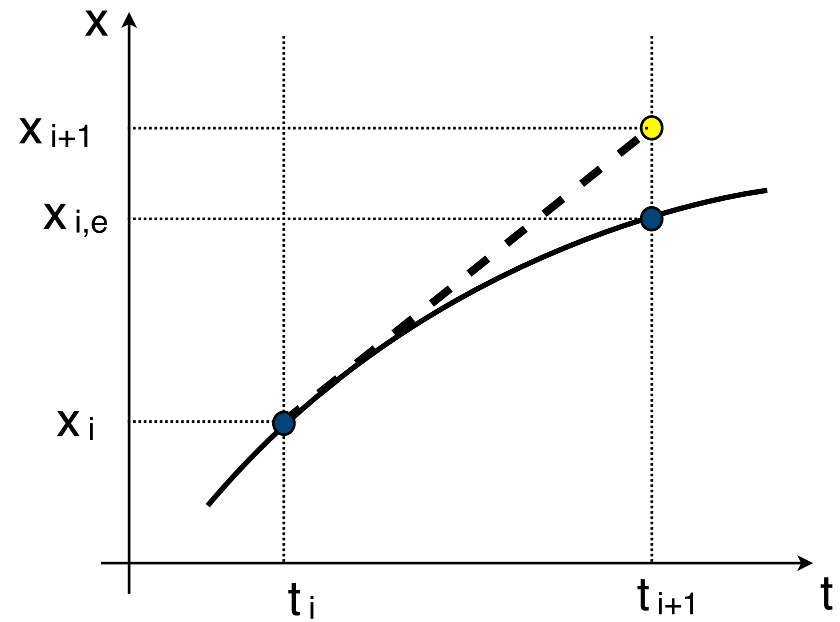
- Collocation methods are based on the **transcription** of the differential equations into a finite set of defects constraints **using a numerical integration scheme**

- Simplest case: Euler's scheme
  - Solution is approximated using a linear expansion

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \dot{\mathbf{x}}(t_i) \cdot (t_{i+1} - t_i)$$
$$= \mathbf{x}_i + \dot{\mathbf{x}}(t_i) \cdot h$$

  - But $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$, then:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i)$$



  - which can be written in terms of **defects constraints**:

$$\boldsymbol{h}_i = \mathbf{x}_{i+1} - \mathbf{x}_i - h \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i) = 0$$

Transcription: Collocation (2/2)

- Other numerical integration schemes can be applied
  - Runge-Kutta schemes:

$$\boldsymbol{h}_i = \mathbf{x}_{i+1} - \mathbf{x}_i - h_i \sum_{j=1}^{k} \beta_j \mathbf{f}_{ij} = 0$$

- The optimal control problem has been **parameterized**:
  - $\mathbf{x}(t)$ and $\mathbf{u}(t)$ $\Longrightarrow$ $\mathbf{X} = \{t_f, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_N, \mathbf{u}_N\}$
  - Minimize:

$$J = \varphi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)dt$$

$\Longrightarrow$

  - Minimize: $J(\mathbf{X})$
  - Dynamics: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ $\Longrightarrow$
  - Subject to: $\boldsymbol{h}(\mathbf{X}) = 0$

Nonlinear Programming Problem

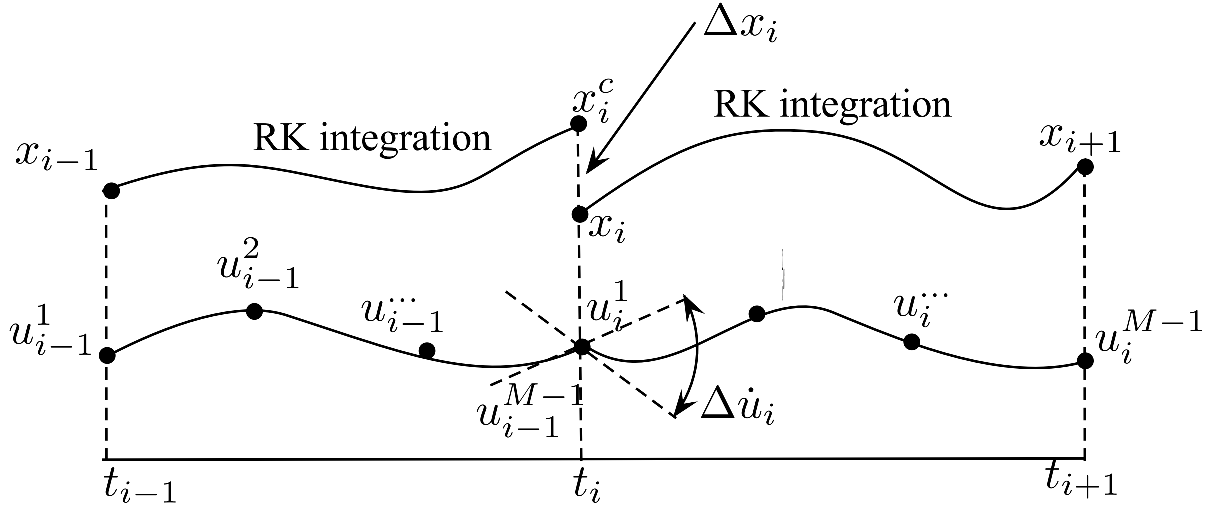
Transcription: Multiple Shooting (1/2)

- Time domain in discretized:

$$t_0 = t_1 < t_2 < ... < t_N = t_f$$

- Within each time interval, splines are used to model the control profile $\mathbf{u}(t)$ ► each time interval contains $M-1$ subintervals, where M is the number of points defining the splines
- On a generic node, the vector of variables will be:

$$\mathbf{X}_i = \{\mathbf{x}_i, \mathbf{u}_i^1, ..., \mathbf{u}_i^{M-1}\}$$



Transcription: Multiple Shooting (2/2)

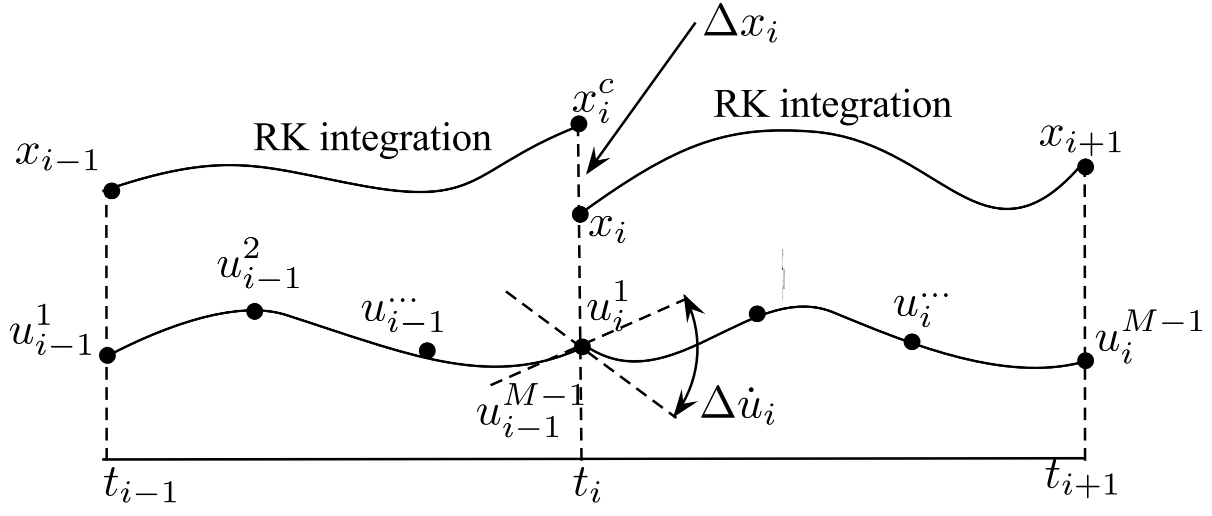
- Within a generic time interval, the splines are used to map the discrete values $\{\mathbf{u}_i^1, ..., \mathbf{u}_i^{M-1}\}$ into continuous functions $\mathbf{u}(t)$

  Numerical integration can be used to compute $\mathbf{x}_{i+1}^c$
- The dynamics is transcribed into a set of defects constraints:

$$\boldsymbol{h}_i = \mathbf{x}_i^c - \mathbf{x}_i = 0$$

- The vector of variables for the nonlinear programming problem is:

$$\mathbf{X} = \{t_f, \mathbf{X}_1, ..., \mathbf{X}_N\}$$



Assignment #5

- Solve Problem #1 with direct transcription and collocation

$$\dot{x}_1 = 0.5x_1 + u$$
$$\dot{x}_2 = u^2 + x_1 u + \frac{5}{4}x_1^2$$

Dynamics

$$J = x_2(1)$$

Obj. fcn.

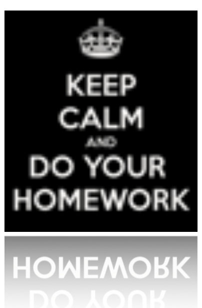
$$x_1(0) = 1$$
$$x_2(0) = 0$$

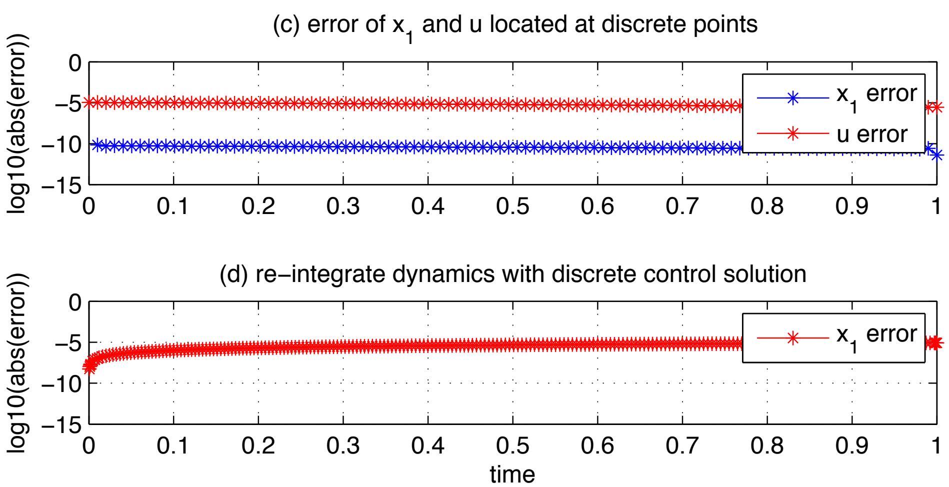
b. c.

$$t_i = 0$$
$$t_f = 1$$

init., final time

- Use Euler method for direct transcription
- Provide analytic gradients and zero initial guess
- Compare numerical vs analytical solution
- Make trade-off between CPU time and solution accuracy





Low-Thrust Earth-Mars Transfer (1/2)

- Optimal control problem:
  - Given the dynamics of the controlled 2 body problem:

$$\ddot{\mathbf{r}} = -\frac{\mu}{\mathrm{r}^3} \cdot \mathbf{r} + \mathbf{u}$$

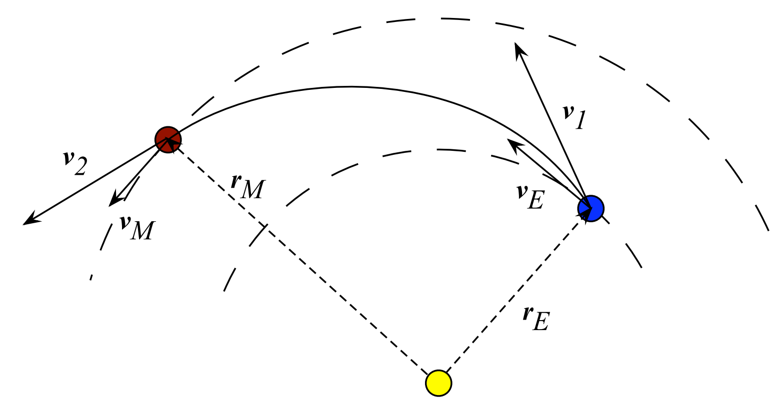
  - Minimize: $J = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\mathrm{d}t = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u} \cdot \mathbf{u}^T \mathrm{d}t$

  - Subject to: $\mathbf{r}(t_0) = \mathbf{r}_E(t_0) \quad \mathbf{v}(t_0) = \mathbf{v}_E(t_0)$

$$\mathbf{r}(t_f) = \mathbf{r}_M(t_f) \quad \mathbf{v}(t_f) = \mathbf{v}_M(t_f)$$

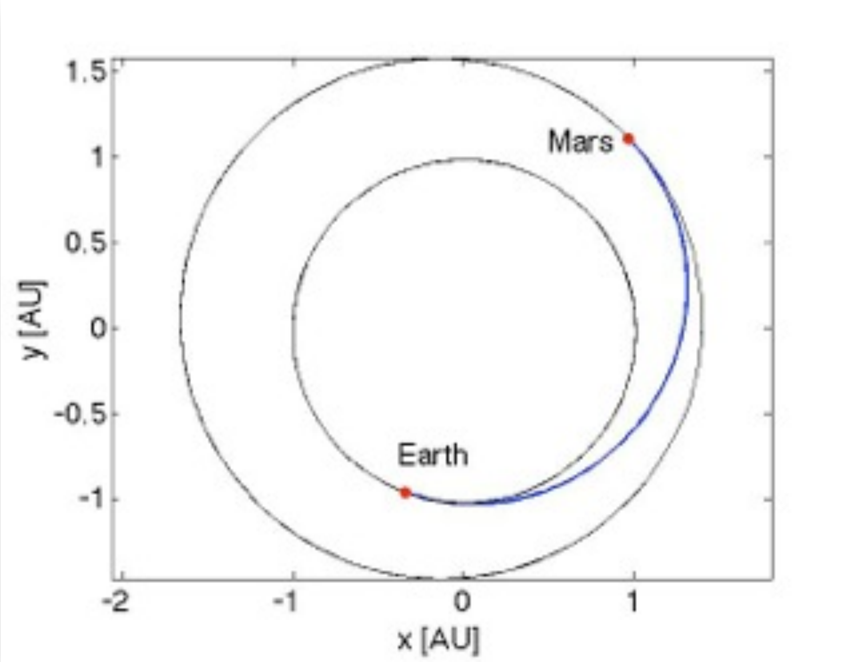
- Transcription technique: **Simple shooting**

Note: Simple shooting is multiple shooting when $N = 1$

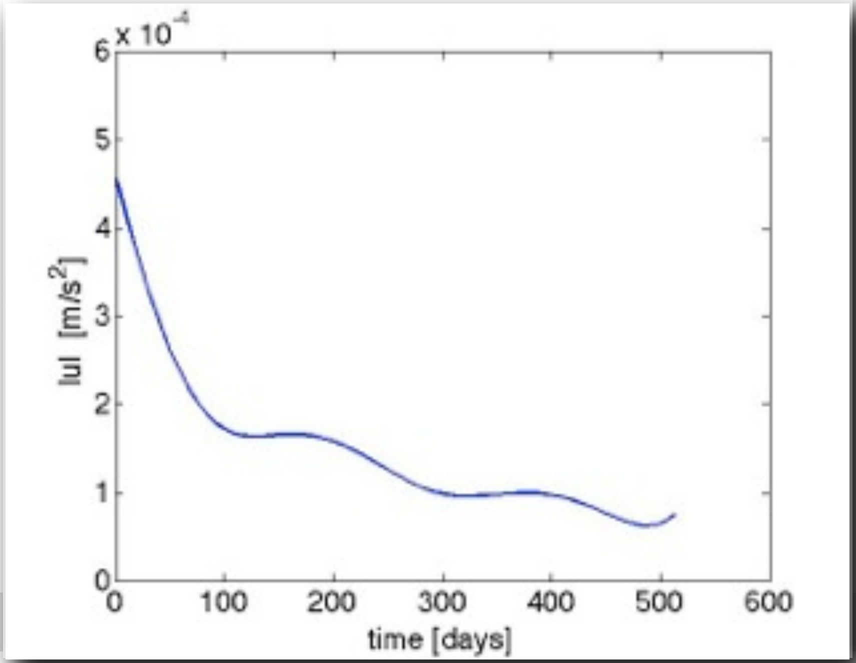


Low-Thrust Earth-Mars Transfer (2/2)

- Cubic splines for $\mathbf{u}(t)$ built on four points ► $M = 4$
- Earth's ephemerides are used to set i.c. for the integration of the shooting method ► constraints on $\mathbf{x}_0$ automatically satisfied
- Optimization variables: $t_0, t_f, \mathbf{u}^1, ..., \mathbf{u}^4$ ($\dim(\mathbf{X}) = 14$)
- First guess: ballistic Lambert's arc



first guess



solution

Controlled Traj. in Relative Dynamics

- Given the equations of the relative dynamics:

$$\begin{cases} \ddot{x} - 2n\dot{y} - 3n^2 x &= 0 \\ \ddot{y} + 2n\dot{x} &= 0 \\ \ddot{z} + n^2 z &= 0 \end{cases}$$

- Minimize: $J = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\mathrm{d}t = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u} \cdot \mathbf{u}^T \mathrm{d}t$

- Subject to: $\mathbf{r}(t_0) = \mathbf{r}_0 \qquad \mathbf{r}(t_f) = \mathbf{r}_f$

  $\mathbf{v}(t_0) = \mathbf{v}_0 \qquad \mathbf{v}(t_f) = \mathbf{v}_f$

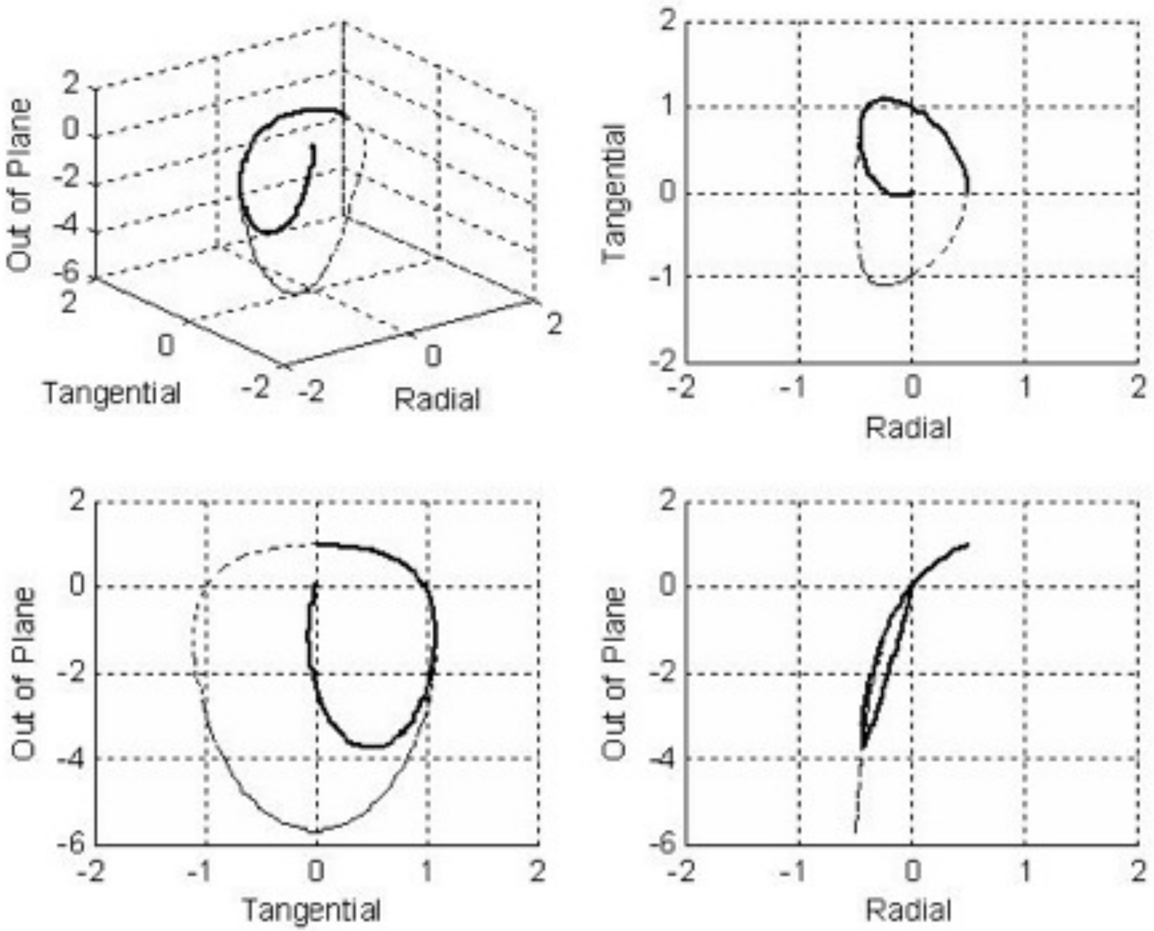
- **Note**: $\mathbf{r}(t_0), \mathbf{v}(t_0) = \mathbf{0},\ \mathbf{r}(t_f), \mathbf{v}(t_f) \neq \mathbf{0}$ ► Formation depl.

  $\mathbf{r}(t_0), \mathbf{v}(t_0) \neq \mathbf{0},\ \mathbf{r}(t_f), \mathbf{v}(t_f) \neq \mathbf{0}$ ► Formation reconf.

  $\mathbf{r}(t_0), \mathbf{v}(t_0) \neq \mathbf{0},\ \mathbf{r}(t_f), \mathbf{v}(t_f) = \mathbf{0}$ ► Docking

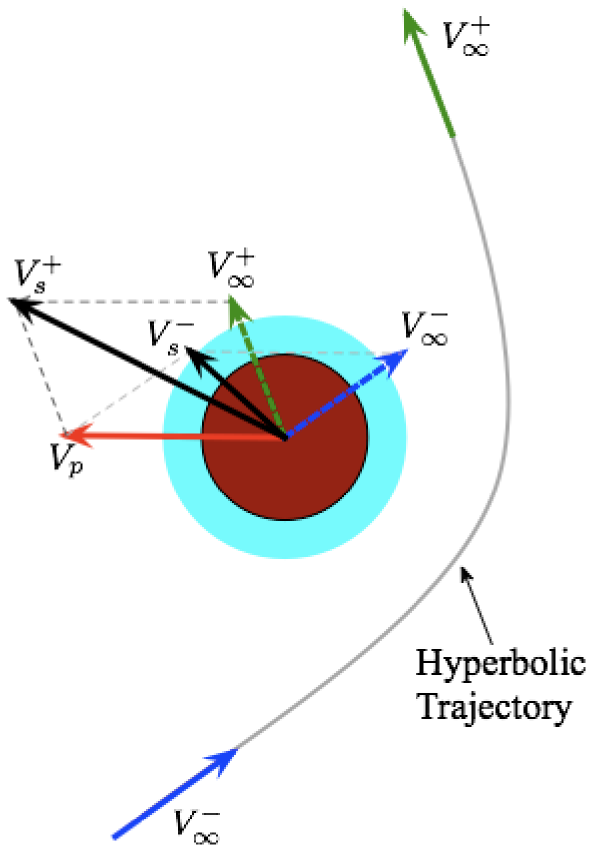
Formation Flying Deployment

- Transcription technique: **Simple shooting**
- Cubic splines for $\mathbf{u}(t)$ built on six points ► $M = 6$
- First guess: $\mathbf{u}(t) = \mathbf{0}$

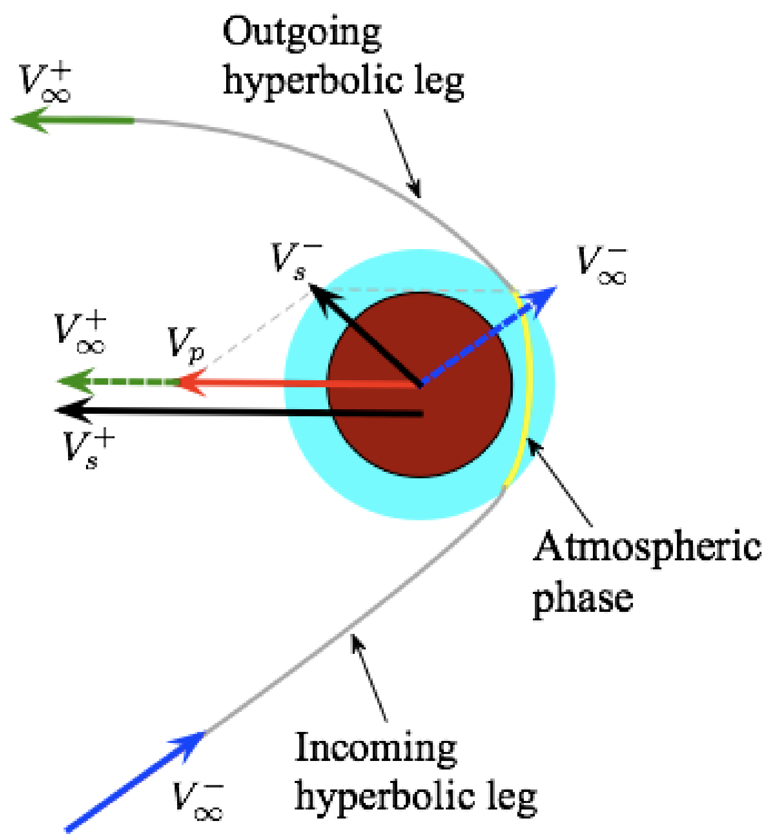


- Reference orbit:
  - $a = 26570\ km$

Mars Aero-Gravity Assist (1/4)



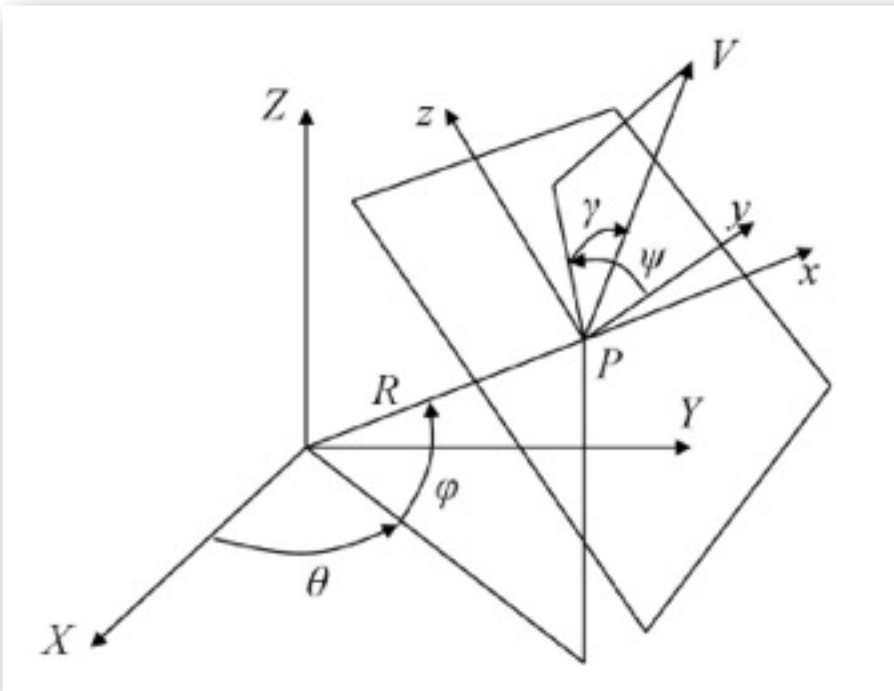
Gravity assist



Aero-Gravity Assist

Mars Aero-Gravity Assist (2/4)

- Dynamics:



$$\begin{cases} \dot{R} = V \sin\gamma \\ \dot{\theta} = \dfrac{V\cos\gamma\cos\psi}{R\cos\phi} \\ \dot{\phi} = \dfrac{V\cos\gamma\sin\psi}{R} \\ \dot{V} = \dfrac{D}{m} - G\sin\gamma \\ V\dot{\gamma} = \dfrac{L\cos\sigma}{m} - G\cos\gamma + \dfrac{V^2\cos\gamma}{R} \\ V\dot{\psi} = \dfrac{L\sin\sigma}{m\cos\gamma} - \dfrac{V^2\tan\phi\cos\gamma\cos\psi}{R} \end{cases}$$

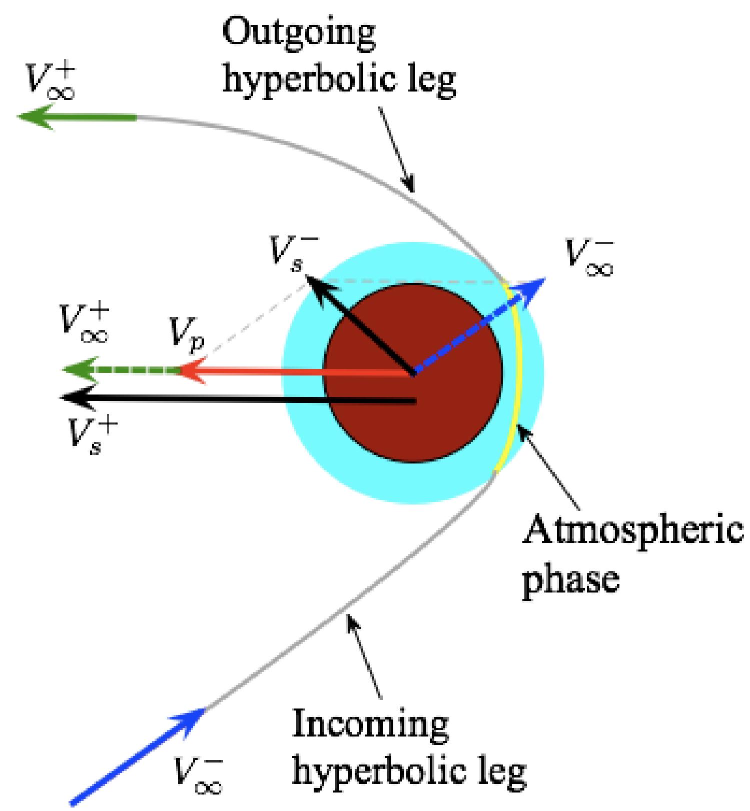
- Control parameters:
  - ~~Bank angle σ~~ ▶ *Planar maneuver*
  - $\lambda = C_L / C_L((L/D)_{max})$
  - Atmospheric entry conditions

Mars Aero-Gravity Assist (3/4)

- Optimal control problem

  Find the optimal control law, $\lambda(t)$, the free atmospheric entry conditions and the final time t_f to

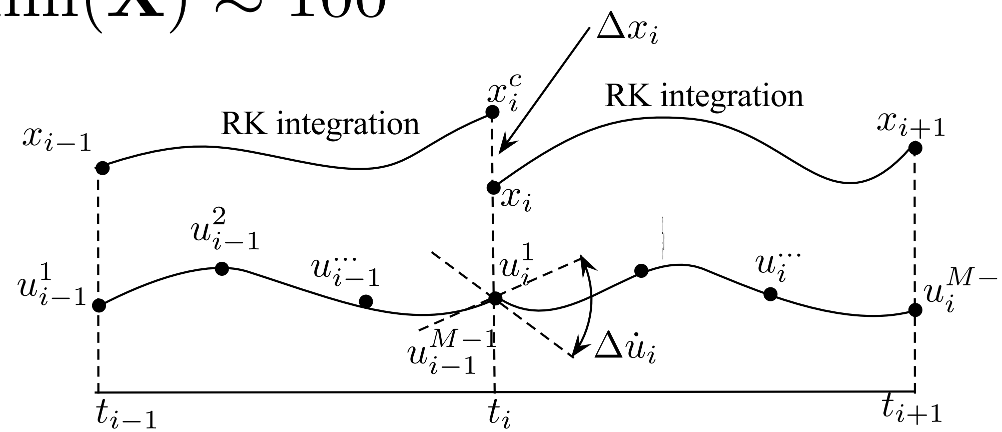
  - Maximize:
    - Final heliocentric velocity V_s^+
  - Subject to:
    - atmospheric entry conditions must be consistent with entry conditions in planetary sphere of influence ($\boldsymbol{V}_\infty^-$ assigned)
    - Convective heating at stagnation point

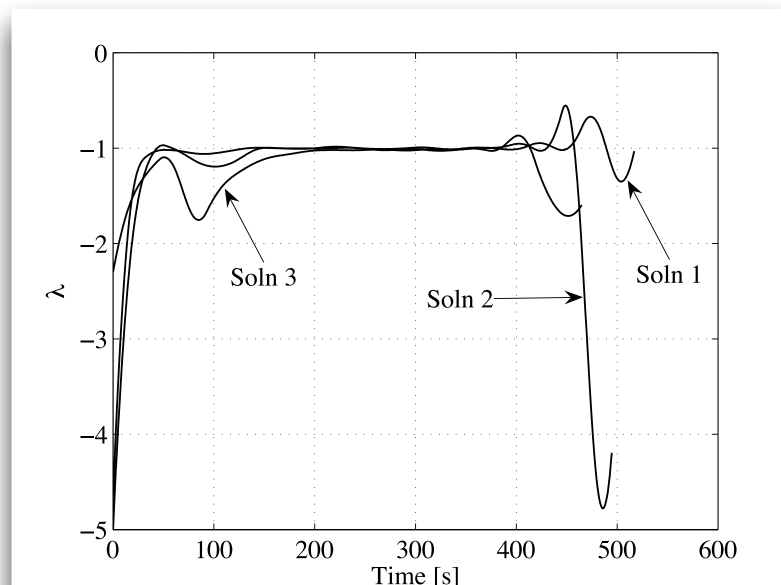


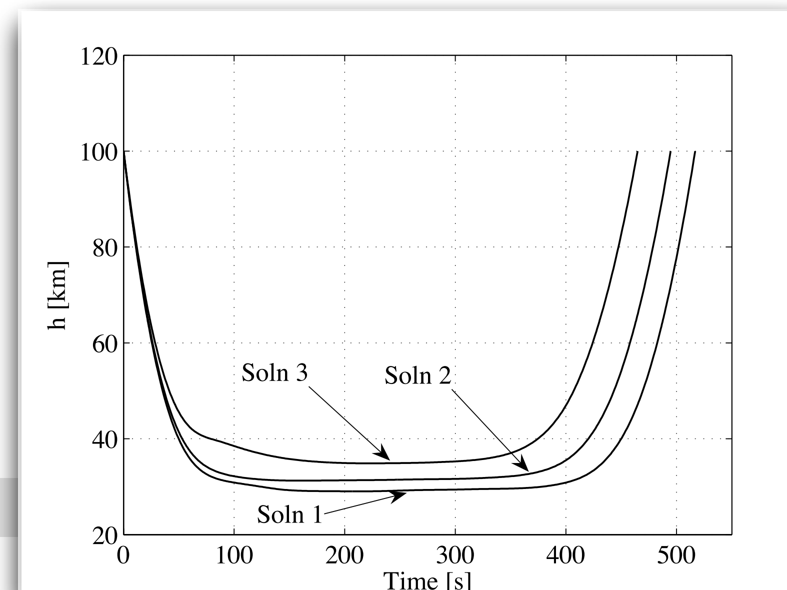
$$\dot{q}_{w_0} = 1.35e-8 \left(\frac{\rho}{r_n}\right)^{1/2} V^{3.04} \left(1 - \frac{h_w}{H}\right) < \dot{q}_{w_0}^{\max}$$

Mars Aero-Gravity Assist (4/4)

- Transcription technique: **Multiple Shooting**
  - $N = 11, \; M = 4$ ► $\dim(\mathbf{X}) \approx 100$
  - Cubic splines
  - First guess using simple shooting and evolutionary algorithms







GTOC II (1/3)

- Optimal control problem:
  - Given the dynamics of the controlled 2 body problem:

$$\ddot{\mathbf{r}} = -\frac{\mu}{\mathrm{r}^3}\cdot\mathbf{r}+\mathbf{u}$$

    Visit **four given asteroids**
  - Maximize: $J = m_f/(t_f - t_0),\, m_f = m_0 \cdot e^{-\frac{1}{I_{sp} g_0}\int_{t_0}^{t_f} |\mathbf{u}| d\tau}$
  - Subject to:
    - $\mathbf{r}(t_{dep,P}) = \mathbf{r}_P(t_{dep,P}) \quad \mathbf{v}(t_{dep,P}) = \mathbf{v}_P(t_{dep,P})$

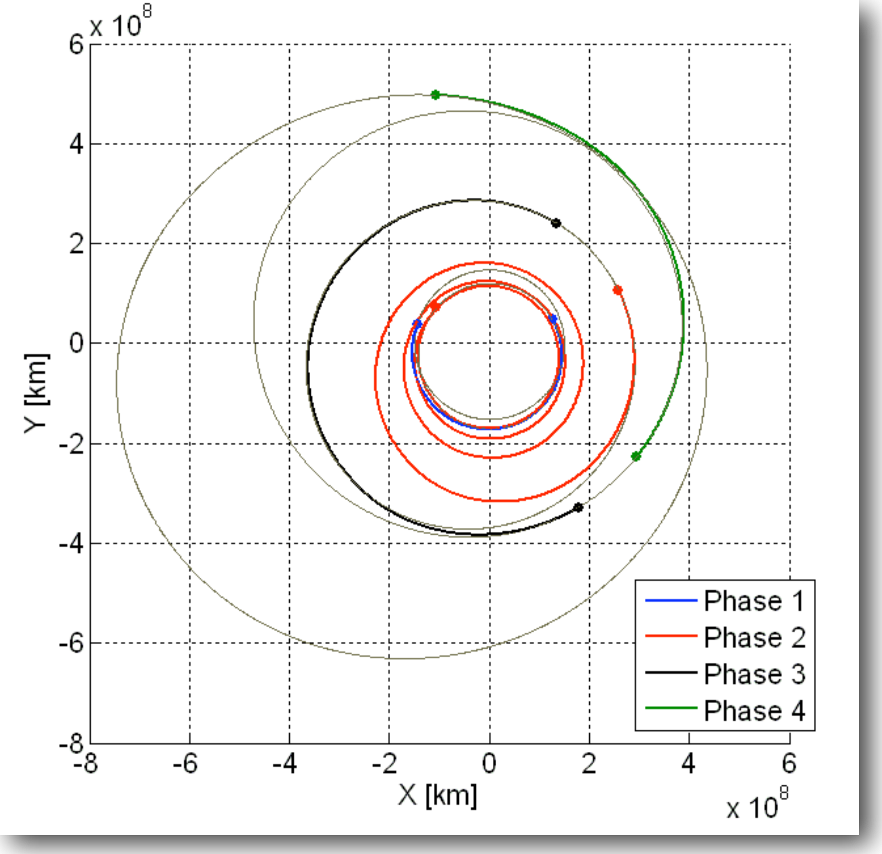
      $\mathbf{r}(t_{arr,P}) = \mathbf{r}_P(t_{arr,P}) \quad \mathbf{v}(t_{arr,P}) = \mathbf{v}_P(t_{arr,P})$
    - $\|\mathbf{u}\| \leq \mathrm{u}^{\max}$
    - $t_{dep,P_i} - t_{arr,P_{i-1}} \leq 90\ days$

GTOC II (2/3)

- Transcription technique: **Collocation**

- Optimization variables:
  - Four departure epochs (Earth and three asteroids)
  - Four transfer times
  - Control parameters deriving from transcription
  - State parameters deriving from transcription

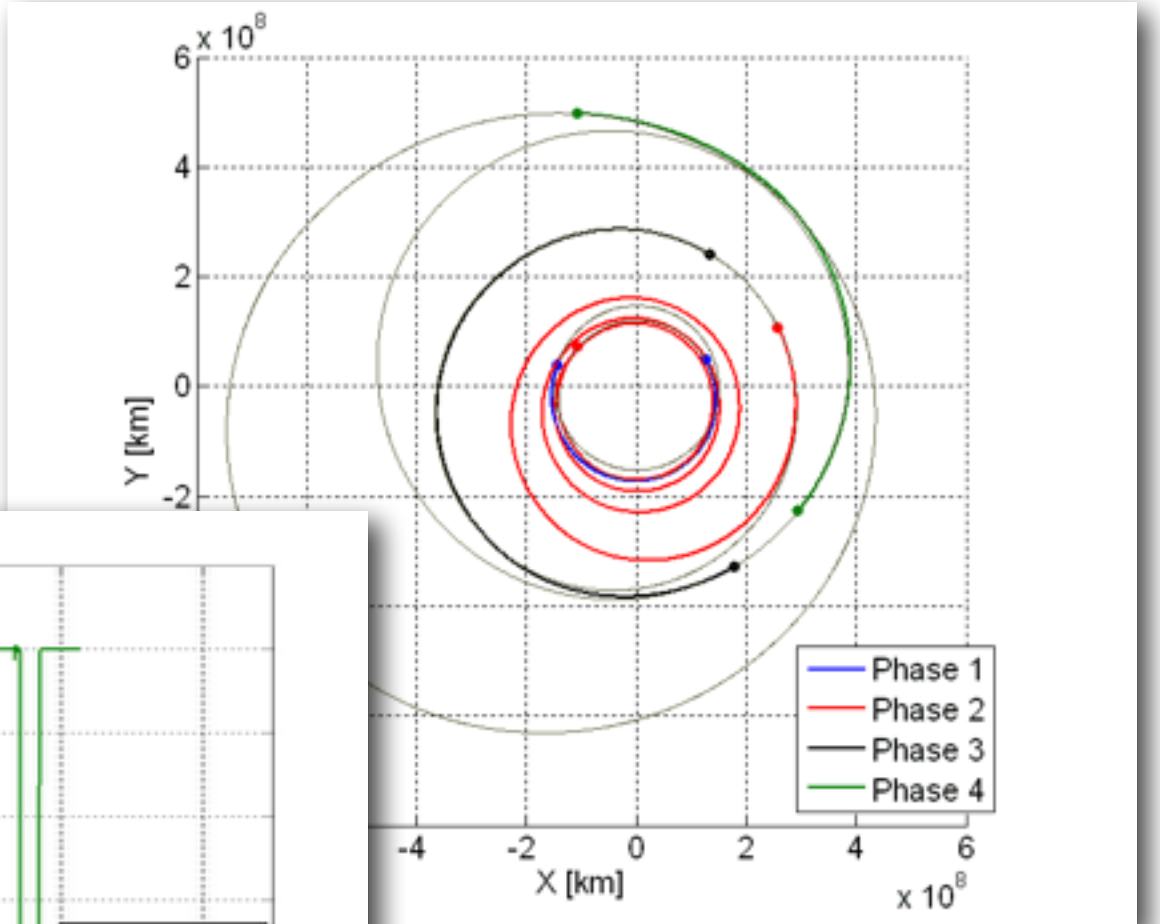
$$\dim(\mathbf{X}) \approx 1000$$

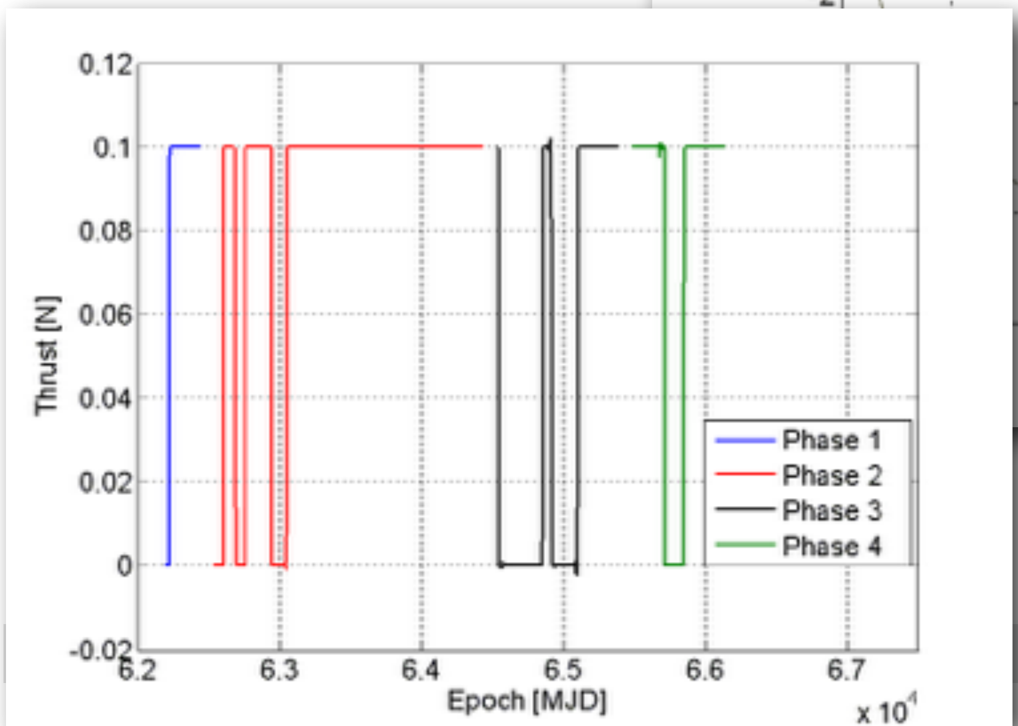


GTOC II (3/3)

- Identified solution:

Collocation methods can better describe discontinuities





Multiple Shooting vs Collocation

- Both Multiple Shooting and Collocation can be considered **robust methods**, even if highly nonlinear dynamics must be dealt with
- **Advantage of Collocation w.r.t. Multiple Shooting**:
  - Better management of discontinuities of the control functions
- **Disadvantage of Collocation w.r.t. Multiple Shooting**:
  - Higher number of variables

Direct Methods vs Indirect Methods

- **Main Advantages of Direct Methods**:
  - No need of deriving the equations related to the necessary conditions for optimality
  - More versatility and easier implementation in black-box tools

- **Main Disadvantage of Direct Methods**:
  - Need of numerical techniques to effectively estimate Hessians and Jacobians

- **Approximate methods**
  - Avoid both indirect and direct
  - Suboptimal solutions

Definition of the *original* problem

Find $\boldsymbol{u}(t), \quad t \in [t_i,\, t_f], \quad \boldsymbol{u} = (u_1, u_2, \ldots, u_m)$

minimizing $$J = \varphi(\boldsymbol{x}(t_f), t_f) + \int_{t_i}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, t)\,\mathrm{d}t$$

with dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t), \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_n)$

and boundary conditions $\boldsymbol{x}(t_i) = \boldsymbol{x}_i \qquad \psi(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), t_f) = 0$

Note: control saturation, path constraints, variable final time, etc., not considered for simplicity

DINAMICA INNOVATING TECHNOLOGY

Solution of the original problem

Hamiltonian of the problem $$H(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{u},t) = L(\boldsymbol{x},\boldsymbol{u},t) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)$$

$\boldsymbol{x}(t),\ \boldsymbol{\lambda}(t),\ \boldsymbol{u}(t),\ \boldsymbol{\nu}$ that satisfy the necessary conditions

$$\dot{\boldsymbol{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} \qquad \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}} \qquad \frac{\partial H}{\partial \boldsymbol{u}} = 0 \qquad (1)$$

under $\boldsymbol{x}(t_i) = \boldsymbol{x}_i$ $\qquad \boldsymbol{\lambda}(t_f) = \left[\frac{\partial \varphi}{\partial \boldsymbol{x}} + \left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\nu}\right]_{t=t_f}$ $\qquad \boldsymbol{\psi}(\boldsymbol{x}(t_f),\boldsymbol{u}(t_f),t_f) = 0$

Iterative methods used to solve (1)

- Convergence depends on initial guess
- Guessing $\boldsymbol{\lambda}_i$ is not trivial (no physical meaning)
- Difficult to treat (algebraic-differential system)
- Deep knowledge of the problem required

Why approximate methods

- Avoid solving problem (1)
- Transform problem (1) into a simpler problem
- Ease the computation of solutions
- Deliver sub-optimal solutions
- Examples
  - Direct transcription [Hargraves&Paris 1987, Enright&Conway, Betts 1998]
  - Generating function [Park&Scheeres, 2006]
  - SDRE [Pearson 1962, Wernli&Cook 1975, Mracek&Cloutier 1998]
  - ASRE [Cimen&Banks 2004]
  - ...

(Time Varying) LQR

Dynamics: $\dot{\boldsymbol{x}} = A(t)\boldsymbol{x} + B(t)\boldsymbol{u},$ Initial condition: $\boldsymbol{x}(t_i) = \boldsymbol{x}_i$

Objective function: $J = \frac{1}{2}\boldsymbol{x}^T(t_f)S(t_f)\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_i}^{t_f}\left[\boldsymbol{x}^T Q(t)\boldsymbol{x} + \boldsymbol{u}^T R(t)\boldsymbol{u}\right]\mathrm{d}t,$

Necessary conditions of optimality

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= A(t)\boldsymbol{x} + B(t)\boldsymbol{u}, \\
\dot{\boldsymbol{\lambda}} &= -Q(t)\boldsymbol{x} - A^T(t)\boldsymbol{\lambda}, \\
0 &= R(t)\boldsymbol{u} + B^T(t)\boldsymbol{\lambda},
\end{aligned}
\quad \Rightarrow \quad \boldsymbol{u} = -R^{-1}(t)B^T(t)\boldsymbol{\lambda}
$$

$$
\begin{pmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{pmatrix} = \begin{bmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{pmatrix} \quad (2)
$$

Linear system of *2n* differential equations

Solution of TVLQR by the STM

Exact solution of system (2):

($\boldsymbol{x}_i$, $\boldsymbol{\lambda}_i$ initial state, costate)

$$\boxed{\begin{aligned}\boldsymbol{x}(t) &= \phi_{xx}(t_i,t)\boldsymbol{x}_i + \phi_{x\lambda}(t_i,t)\boldsymbol{\lambda}_i,\\ \boldsymbol{\lambda}(t) &= \phi_{\lambda x}(t_i,t)\boldsymbol{x}_i + \phi_{\lambda\lambda}(t_i,t)\boldsymbol{\lambda}_i,\end{aligned}} \quad (3)$$

$\phi_{xx},\ \phi_{x\lambda},\ \phi_{\lambda x},\ \phi_{\lambda\lambda}$ are the components of the state transition matrix (STM)

$$\Phi(t_i,t) = \begin{bmatrix} \phi_{xx}(t_i,t) & \phi_{x\lambda}(t_i,t) \\ \phi_{\lambda x}(t_i,t) & \phi_{\lambda\lambda}(t_i,t) \end{bmatrix}$$

STM subject to
$$\begin{bmatrix} \dot\phi_{xx} & \dot\phi_{x\lambda} \\ \dot\phi_{\lambda x} & \dot\phi_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{bmatrix} \phi_{xx} & \phi_{x\lambda} \\ \phi_{\lambda x} & \phi_{\lambda\lambda} \end{bmatrix}$$

with $\phi_{xx}(t_i,t_i) = I_{n\times n},\ \phi_{x\lambda}(t_i,t_i) = 0_{n\times n},\ \phi_{\lambda x}(t_i,t_i) = 0_{n\times n},\ \phi_{\lambda\lambda}(t_i,t_i) = I_{n\times n}$

⇨

- If $\boldsymbol{\lambda}_i$ was known, it would be possible to compute $\boldsymbol{x}(t),\ \boldsymbol{\lambda}(t)$ through (3), and $\boldsymbol{u}(t)$ with $\boldsymbol{u} = -R^{-1}(t)B^T(t)\boldsymbol{\lambda}$
- $\boldsymbol{\lambda}_i$ computed by using (3) and the final condition (3 types)

SolveLQR.m

Hard constrained problem (HCP)

- Final state given

$$\dot{\boldsymbol{x}} = A(t)\boldsymbol{x} + B(t)\boldsymbol{u}, \qquad \boldsymbol{x}(t_i) = \boldsymbol{x}_i$$

$$J = \frac{1}{2}\int_{t_i}^{t_f} \left[\boldsymbol{x}^T Q(t)\boldsymbol{x} + \boldsymbol{u}^T R(t)\boldsymbol{u}\right] \mathrm{d}t, \qquad \boldsymbol{x}(t_f) = \boldsymbol{x}_f$$

Statement of HCP

Write the first of (3) at $t = t_f$,

$$\boldsymbol{x}_f = \phi_{xx}(t_i, t_f)\boldsymbol{x}_i + \phi_{x\lambda}(t_i, t_f)\boldsymbol{\lambda}_i,$$

and solve for $\boldsymbol{\lambda}_i$; i.e.,

$$\boldsymbol{\lambda}_i(\boldsymbol{x}_i, \boldsymbol{x}_f, t_i, t_f) = \phi_{x\lambda}^{-1}(t_i, t_f)\left[\boldsymbol{x}_f - \phi_{xx}(t_i, t_f)\boldsymbol{x}_i\right]$$

⇨ Solving a HCP requires inverting the $n \times n$ matrix $\phi_{x\lambda}$

INNOVATING TECHNOLOGY

Soft constrained problem (SCP)

- Final state not given

$$\dot{\boldsymbol{x}} = A(t)\boldsymbol{x} + B(t)\boldsymbol{u}, \qquad \boldsymbol{x}(t_i) = \boldsymbol{x}_i, \qquad \boldsymbol{\lambda}(t_f) = S(t_f)\boldsymbol{x}(t_f),$$

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f)S(t_f)\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[\boldsymbol{x}^T Q(t)\boldsymbol{x} + \boldsymbol{u}^T R(t)\boldsymbol{u}\right] \mathrm{d}t$$

Statement of SCP

Write (3) at $t = t_f$,

$$\begin{aligned} \boldsymbol{x}(t_f) &= \phi_{xx}(t_i,t_f)\boldsymbol{x}_i + \phi_{x\lambda}(t_i,t_f)\boldsymbol{\lambda}_i, \\ S(t_f)\boldsymbol{x}(t_f) &= \phi_{\lambda x}(t_i,t_f)\boldsymbol{x}_i + \phi_{\lambda\lambda}(t_i,t_f)\boldsymbol{\lambda}_i \end{aligned}$$

and solve for $\boldsymbol{\lambda}_i$; i.e.,

$$\boldsymbol{\lambda}_i(\boldsymbol{x}_i, t_i, t_f) = \left[\phi_{\lambda\lambda}(t_i,t_f) - S(t_f)\phi_{x\lambda}(t_i,t_f)\right]^{-1}\left[S(t_f)\phi_{xx}(t_i,t_f) - \phi_{\lambda x}(t_i,t_f)\right]\boldsymbol{x}_i$$

⇒ Solving a SCP requires inverting the $n \times n$ matrix $[\phi_{\lambda\lambda}(t_i,t_f) - S(t_f)\phi_{x\lambda}(t_i,t_f)]$

Mixed constrained problem (MCP)

- Some components of final state given (and some not)

$$\dot{\boldsymbol{x}} = A(t)\boldsymbol{x} + B(t)\boldsymbol{u}, \quad \boldsymbol{x}(t_i) = \boldsymbol{x}_i, \quad x_i(t_f) = x_{i,f}, \quad \lambda_j(t_f) = S(t_f)x_j(t_f)$$

$$J = \frac{1}{2}\boldsymbol{x}_j^T(t_f)S(t_f)\boldsymbol{x}_j(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[\boldsymbol{x}^T Q(t)\boldsymbol{x} + \boldsymbol{u}^T R(t)\boldsymbol{u}\right] \mathrm{d}t$$

Statement of MCP (x_i given, x_j free)

Write (3) at $t = t_f$, write $\boldsymbol{\lambda}_i = (\lambda_{i,i}, \lambda_{i,j})^T$, and solve for $\lambda_{i,i}$ using $x_{i,f}$ (HCP) and for $\lambda_{i,j}$ using $\lambda_j(t_f) = S(t_f)x_j(t_f)$ (SCP); i.e.,

$$\lambda_{i,i}(x_{i,i}, x_{f,i}, t_i, t_f) = \phi_{x\lambda,i}^{-1}(t_i, t_f)\left[x_{f,i} - \phi_{xx,i}(t_i, t_f)x_{i,i}\right],$$

$$\lambda_{i,j}(x_{i,j}, t_i, t_f) = \left[\phi_{\lambda\lambda,j}(t_i, t_f) - S(t_f)\phi_{x\lambda,j}(t_i, t_f)\right]^{-1}\left[S(t_f)\phi_{xx,j}(t_i, t_f) - \phi_{\lambda x,j}(t_i, t_f)\right]x_{i,j}$$

⇒ Solving a SCP requires inverting the matrices $\phi_{x\lambda,i}$ and $\left[\phi_{\lambda\lambda,j}(t_i, t_f) - S(t_f)\phi_{x\lambda,j}(t_i, t_f)\right]^{-1}$

Idea of the method

Re-write the nonlinear problem as

original dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t),$$

factorized dynamics

$$\dot{\boldsymbol{x}} = A(\boldsymbol{x}, t)\boldsymbol{x} + B(\boldsymbol{x}, \boldsymbol{u}, t)\boldsymbol{u},$$

$$J = \varphi(\boldsymbol{x}(t_f), t_f) + \int_{t_i}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, t)\,\mathrm{d}t$$

original objective function

factorized objective function

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f)S(\boldsymbol{x}(t_f), t_f)\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[\boldsymbol{x}^T Q(\boldsymbol{x}, t)\boldsymbol{x} + \boldsymbol{u}^T R(\boldsymbol{x}, t)\boldsymbol{u}\right]\mathrm{d}t$$

Idea: to use state-dependent matrices $A(\boldsymbol{x}, t),\ B(\boldsymbol{x}, \boldsymbol{u}, t),\ Q(\boldsymbol{x}, t),\ R(\boldsymbol{x}, t)$

such that for given arguments $\overline{\boldsymbol{x}}(t),\ \overline{\boldsymbol{u}}(t)$ they depend on time only; i.e.,

$$A(\overline{\boldsymbol{x}}(t), t),\ B(\overline{\boldsymbol{x}}(t), \overline{\boldsymbol{u}}(t), t),\ Q(\overline{\boldsymbol{x}}(t), t),\ R(\overline{\boldsymbol{x}}(t), t) \Rightarrow \underline{A(t),\ B(t),\ Q(t),\ R(t)}$$

The algorithm: iterations

Iteration 0 - Find $\boldsymbol{x}^{(0)}(t),\ \boldsymbol{u}^{(0)}(t)$ solving "***Problem 0***" $\left[\overline{\boldsymbol{x}} = \boldsymbol{x}_i,\ \overline{\boldsymbol{u}} = \boldsymbol{0}\right]$

$$\dot{\boldsymbol{x}}^{(0)} = A(\boldsymbol{x}_i, t)\boldsymbol{x}^{(0)} + B(\boldsymbol{x}_i, \boldsymbol{0}, t)\boldsymbol{u}^{(0)},$$

$$J = \frac{1}{2}\boldsymbol{x}^{(0)T}(t_f)S(\boldsymbol{x}_i, t_f)\boldsymbol{x}^{(0)}(t_f) + \frac{1}{2}\int_{t_i}^{t_f}\left[\boldsymbol{x}^{(0)T}Q(\boldsymbol{x}_i, t)\boldsymbol{x}^{(0)} + \boldsymbol{u}^{(0)T}R(\boldsymbol{x}_i, t)\boldsymbol{u}^{(0)}\right]\mathrm{d}t$$

Iteration i - Find $\boldsymbol{x}^{(i)}(t),\ \boldsymbol{u}^{(i)}(t)$ satisfying "***Problem i***" $\left[\overline{\boldsymbol{x}} = \boldsymbol{x}^{(i-1)},\ \overline{\boldsymbol{u}} = \boldsymbol{u}^{(i-1)}\right]$

$$\dot{\boldsymbol{x}}^{(i)} = A(\boldsymbol{x}^{(i-1)}(t), t)\boldsymbol{x}^{(i)} + B(\boldsymbol{x}^{(i-1)}(t), \boldsymbol{u}^{(i-1)}(t), t)\boldsymbol{u}^{(i)},$$

$$J=\frac{1}{2}\boldsymbol{x}^{(i)T}(t_f)S(\boldsymbol{x}^{(i-1)}(t_f), t_f)\boldsymbol{x}^{(i)}(t_f)+\frac{1}{2}\int_{t_i}^{t_f}\left[\boldsymbol{x}^{(i)T}Q(\boldsymbol{x}^{(i-1)}(t), t)\boldsymbol{x}^{(i)}+\boldsymbol{u}^{(i)T}R(\boldsymbol{x}^{(i-1)}(t), t)\boldsymbol{u}^{(i)}\right]\mathrm{d}t$$

The algorithm: convergence

Problem i = TVLQR

- Each problem corresponds to a time-varying linear quadratic regulator (TVLQR)
- The method requires solving a series of TVLQR
- Iterations terminate when, for given ε

$$||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i-1)}||_\infty = \max_{t \in [t_i, t_f]} \{|x_j^{(i)}(t) - x_j^{(i-1)}(t)|, j = 1, \ldots, n\} \leq \varepsilon$$

the difference between each component of the state, evaluated for all times, changes by less than ε between two successive iterations

Numerical examples

- Low-thrust dynamics in central vector field
  - Low-thrust rendez-vous
  - Low-thrust orbital transfer
  - Low-thrust stationkeeping of GEO satellites

Rendez-vous: statement and factorization

Dynamics

$$
\begin{aligned}
\dot{x}_1 &= x_3, \\
\dot{x}_2 &= x_4, \\
\dot{x}_3 &= 2x_4 - (1 + x_1)(1/r^3 - 1) + u_1, \\
\dot{x}_4 &= -2x_3 - x_2(1/r^3 - 1) + u_2,
\end{aligned}
$$

with $r = \sqrt{(x_1 + 1)^2 + x_2^2}$

- Rotating frame
  - x_1 radial displacement
  - x_2 transversal displacement
- Normalized units
  - length unit = orbit radius
  - time unit = $1/\omega$

State: $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$

Control: $\boldsymbol{u} = (u_1, u_2)$

Initial condition: $\boldsymbol{x}_i = (0.2, 0.2, 0.1, 0.1)$

[Park&Scheeres, 2006]

Factorization (dynamics)

$$
\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ f(x_1, x_2)(1 + 1/x_1) & 0 & 0 & 2 \\ 0 & f(x_1, x_2) & -2 & 0 \end{bmatrix}}_{A(\boldsymbol{x})} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{B} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}
$$

with $f(x_1, x_2) = -1/[(x_1 + 1)^2 + x_2^2]^{3/2} + 1$

Rendez-vous: HCP and SCP definition

HCP

$$J = \frac{1}{2}\int_{t_i}^{t_f} \boldsymbol{u}^T \boldsymbol{u}\,\mathrm{d}t$$

$$\boldsymbol{x}_f = (0,0,0,0),\ \ t_i = 0,\ \ t_f = 1$$

SCP

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f)S\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \boldsymbol{u}^T \boldsymbol{u}\,\mathrm{d}t$$

$$S = \mathrm{diag}(25, 15, 10, 10),\ \ t_i = 0,\ \ t_f = 1$$

$\boldsymbol{x}(t_f)$ free

Factorization (objective function)

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f)S(t_f)\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[\boldsymbol{x}^T Q(t)\boldsymbol{x} + \boldsymbol{u}^T R(t)\boldsymbol{u}\right]\mathrm{d}t$$

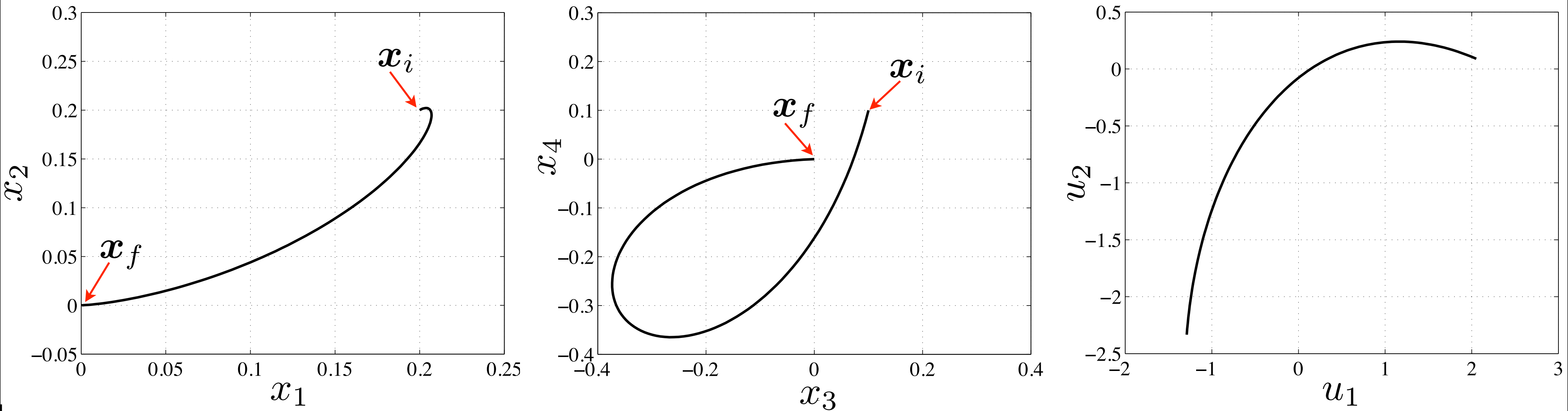
with $Q = 0_{4\times 4},\ \ R = I_{2\times 2}$

(S not defined in HCP)

- Termination tolerance $\varepsilon = 10^{-9}$

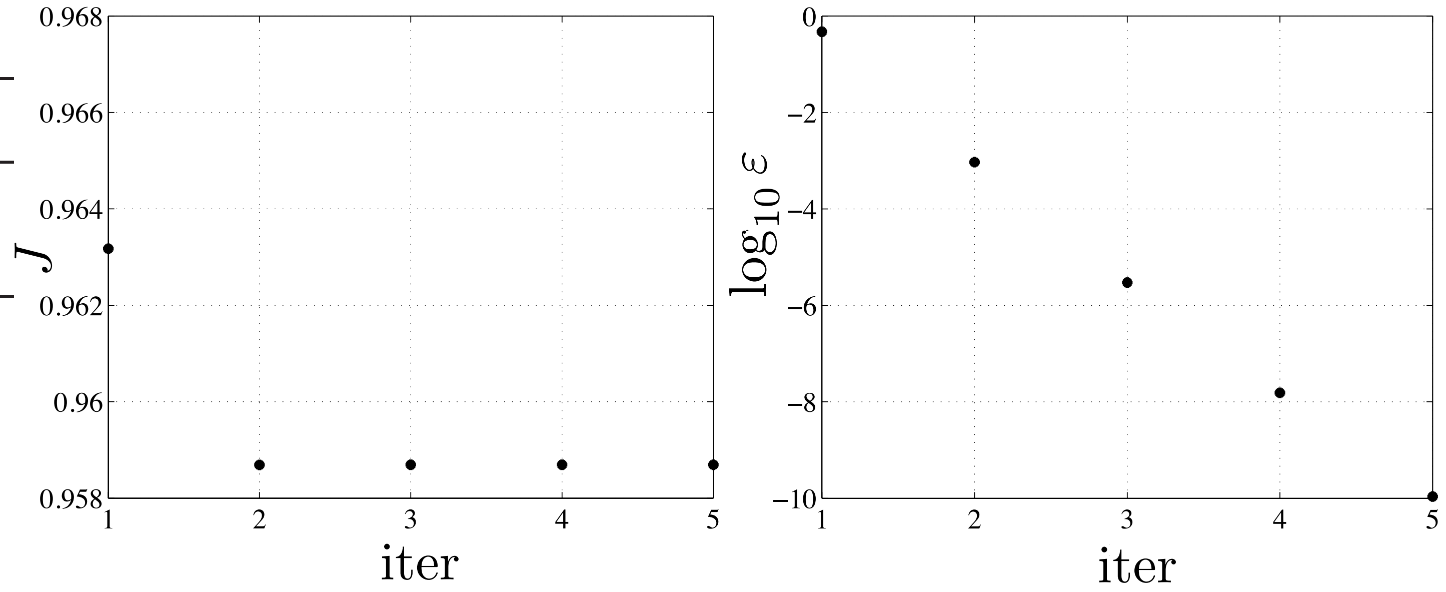
  (valid for all examples show)

Rendez-vous: results (HCP)



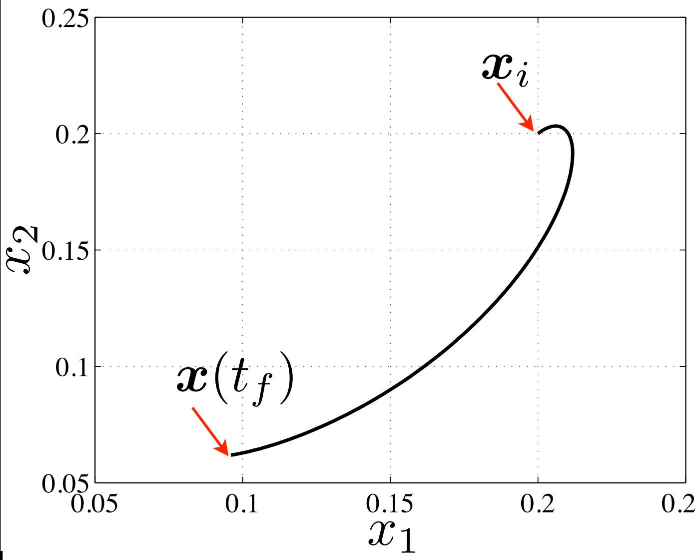
Problem	J	Iter	CPU Time (s)
▷ HCP	0.9586	5	0.447
SCP	0.5660	6	0.492

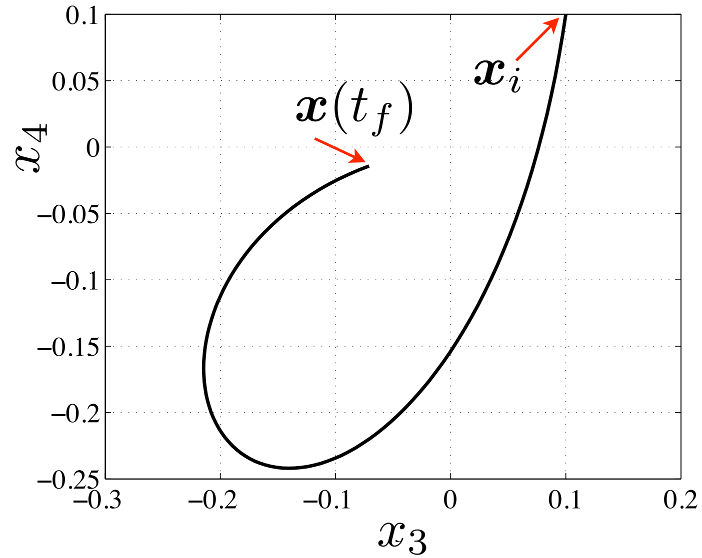
CPU Time referred to an Intel Core 2 Duo 2 GHz with 4 GB RAM running Mac OS X 10.6

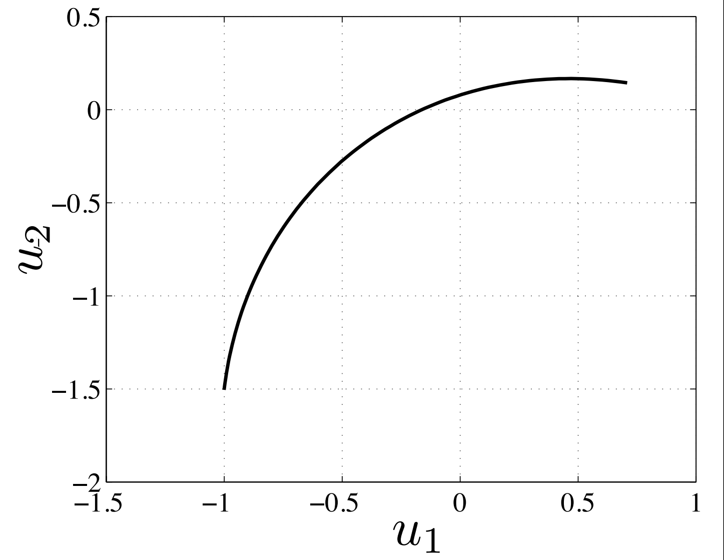


SolveNL.m

Rendez-vous: results (SCP)

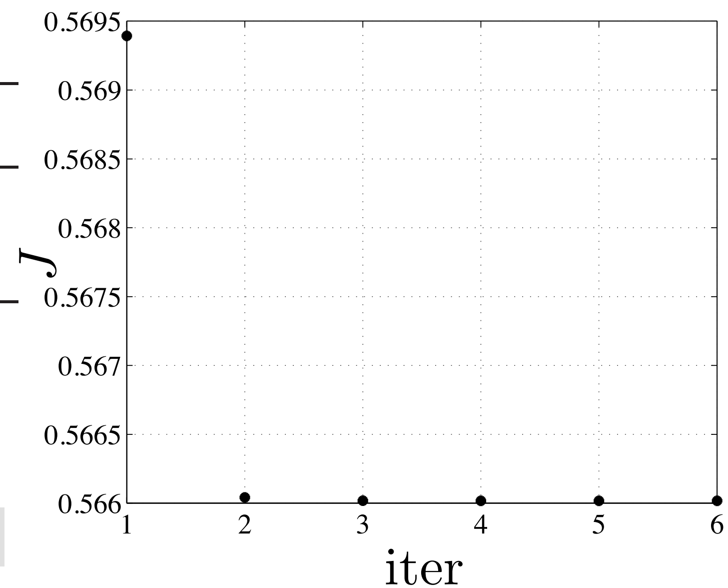


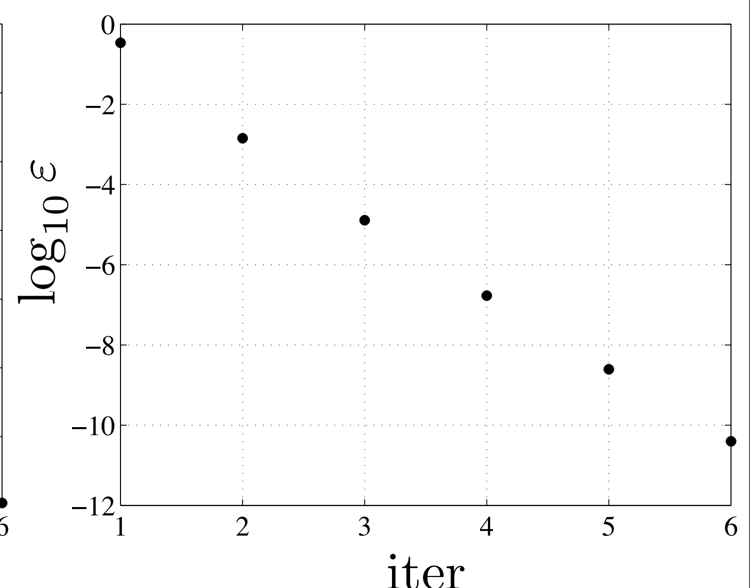




Problem	J	Iter	CPU Time (s)
HCP	0.9586	5	0.447
▷ SCP	0.5660	6	0.492

CPU Time referred to an Intel Core 2 Duo 2 GHz with 4 GB RAM running Mac OS X 10.6





SolveNL.m

Orbital transfer: statement

Dynamics

$$\begin{aligned}\dot{x}_1 &= x_3,\\ \dot{x}_2 &= x_4,\\ \dot{x}_3 &= x_1 x_4^2 - 1/x_1^2 + u_1,\\ \dot{x}_4 &= -2x_3 x_4/x_1 + u_2/x_1.\end{aligned}$$

State

$$\boldsymbol{x} = (x_1, x_2, x_3, x_4)$$

Control

$$\boldsymbol{u} = (u_1, u_2)$$

- Rotating frame (polar coordinates)
  - x_1 radial distance
  - x_2 angular phase
- Normalized units
  - length unit = initial orbit radius
  - time unit = $1/\omega$ (initial orbit)

Objective function

$$J = \frac{1}{2}\int_{t_i}^{t_f} \boldsymbol{u}^T \boldsymbol{u}\,\mathrm{d}t$$

$$t_i = 0 \qquad t_f = \pi$$

Initial condition

$$\boldsymbol{x}_i = (1, 0, 0, 1)$$

Final conditions

Problem A $\boldsymbol{x}_f = (1.52, \underline{\pi}, 0, \sqrt{1/1.52})$

Problem B $\boldsymbol{x}_f = (1.52, \underline{1.5\pi}, 0, \sqrt{1/1.52})$

Orbital transfer: results

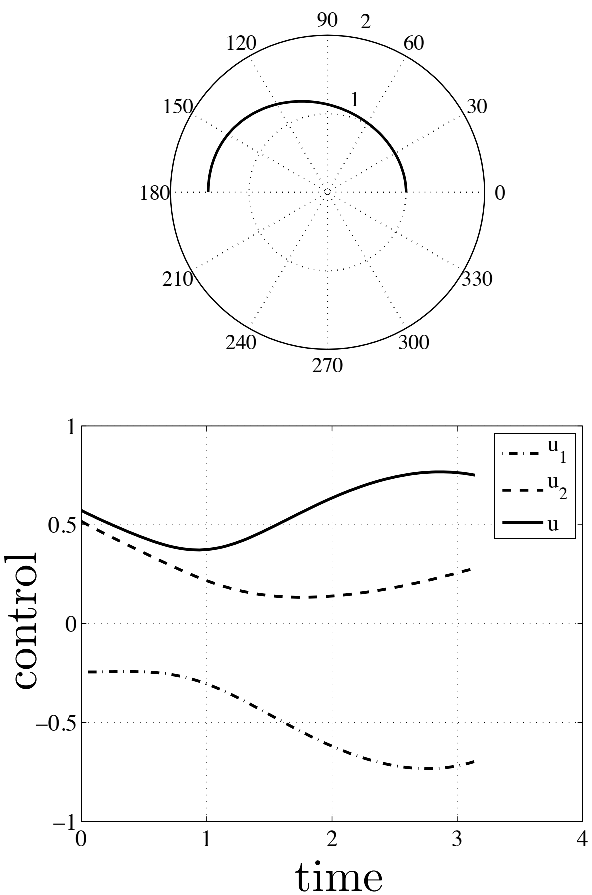
Factorization

$$A(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1/x_1^3 & 0 & 0x_1x_4 & \\ 0 & 0 & -2x_4/x_1 0 & \end{bmatrix}, \quad B(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1/x_1 \end{bmatrix}$$

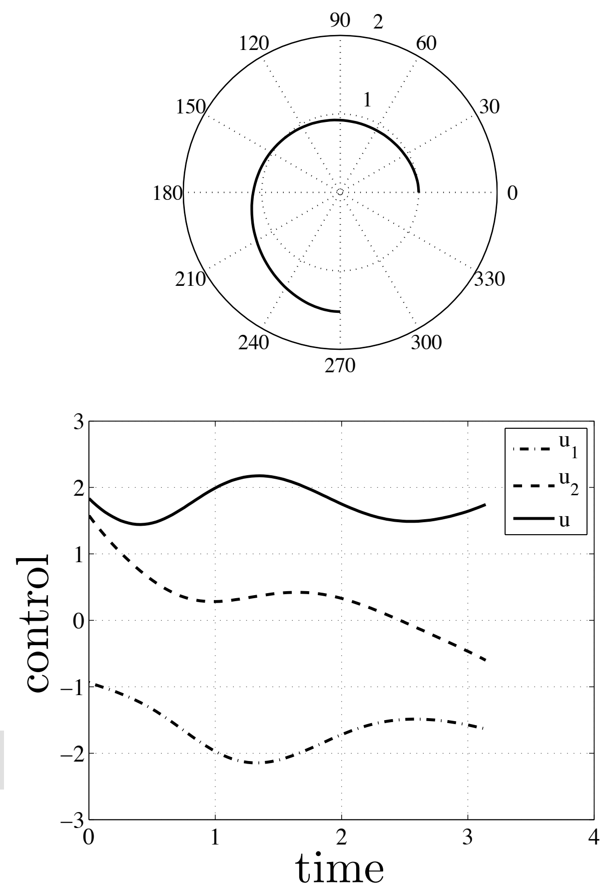
	Problem	J	Iter	CPU Time (s)
A	$x_{2,f} = \pi$	0.5298	22	6.262
B	$x_{2,f} = 1.5\pi$	4.8665	123	47.865

Results

Problem A



Problem B



Stationkeeping: statement

Dynamics

$$
\begin{aligned}
\dot{x}_1 &= x_4,\\
\dot{x}_2 &= x_5,\\
\dot{x}_3 &= x_6,\\
\dot{x}_4 &= -\frac{1}{x_1^2} + x_1 x_6^2 + x_1(x_5+1)\cos^2 x_3 + a_1(x_1,x_2,x_3) + u_1,\\
\dot{x}_5 &= 2x_6(x_5+1)\tan x_3 - 2\frac{x_4}{x_1}(x_5+1) + \frac{a_2(x_1,x_2,x_3)}{x_1\cos x_3} + \frac{u_2}{x_1\cos x_3},\\
\dot{x}_6 &= -2\frac{x_4}{x_1}x_6 - (x_5+1)^2\sin x_3\cos x_3 + \frac{a_3(x_1,x_2,x_3)}{x_1} + \frac{u_3}{x_1},
\end{aligned}
$$

State

$$\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$$

Control

$$\boldsymbol{u} = (u_1, u_2, u_3)$$

Initial condition

$$\boldsymbol{x}_i = (1, 0.05 \times 180/\pi, 0.05 \times 180/\pi, 0, 0, 0) \quad t_i = 0$$

Final condition

$$x_{f,1} = 1 \quad x_{f,j} \text{ free} \quad j = 2, \ldots, 6 \quad t_f = \pi$$

Objective function

$$
\begin{aligned}
Q &= \mathrm{diag}(0,1,1,1,1,1),\\
R &= \mathrm{diag}(1,1,1),\\
S &= 100\,\mathrm{diag}(1,1,1,1,1)
\end{aligned}
$$

- Rotating frame (spherical coordinates)
  - x_1 radial distance
  - x_2 longitude deviation
  - x_3 latitude
- Normalized units
  - length unit = GEO radius
  - time unit = $1/\omega$ (initial orbit)
- Reference longitude = 60 E
- Perturbations $a_1,\ a_2,\ a_3$

DINAMICA
INNOVATING TECHNOLOGY

Stationkeeping: factorization

Factorization

$$A(\boldsymbol{x}) = \begin{bmatrix} & 0_{33} & & & I_{33} & \\ a_{41} & 0 & 0 & 0 & a_{45} & a_{46} \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\ a_{61} & 0 & 0 & a_{64} & a_{65} & a_{66} \end{bmatrix}, \quad B(\boldsymbol{x}) = \begin{bmatrix} & 0_{33} & \\ 1 & 0 & 0 \\ 0 & \frac{1}{r^2 \cos^2 \varphi} & 0 \\ 0 & 0 & \frac{1}{r^2} \end{bmatrix}$$

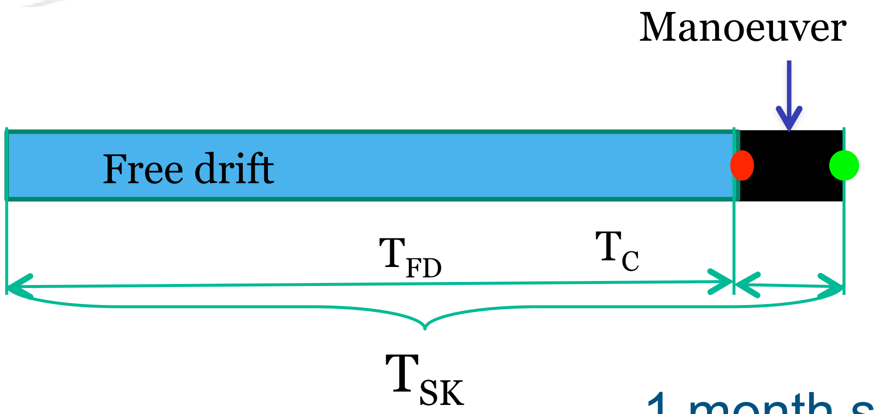
with

$$\begin{aligned} a_{41} &= -\frac{1}{x_1^3} + \alpha_1 x_6^2 + (\alpha_2 x_5^2 + 2\alpha_3 x_5 + 1)\cos^2 x_3, & a_{56} &= [2 + 2(1-\beta_1)x_5]\tan x_3, \\ a_{45} &= [(1-\alpha_2)x_1 x_5 + 2(1-\alpha_3)]\cos^2 x_3, & a_{61} &= -\frac{1}{2x_1}\sin 2x_3, \\ a_{46} &= (1-\alpha_1)x_1 x_6, & a_{64} &= -2(1-\gamma_1)\frac{1}{x_1}x_6, \\ a_{54} &= -\frac{2}{x_1} - 2(1-\beta_2)\frac{1}{x_1}x_5, & a_{65} &= [-\tfrac{1}{2}x_5 - 1]\sin 2x_3, \\ a_{55} &= 2\beta_1 x_5 \tan x_3 - 2\beta_2 \frac{x_4}{x_1}, & a_{66} &= -2\gamma_1 \frac{x_4}{x_1} \end{aligned}$$

- $\alpha_1, \ \alpha_2, \ \alpha_3, \ \beta_1, \ \beta_2, \ \gamma_1$ free parameters
- Can vary in [0, 1]

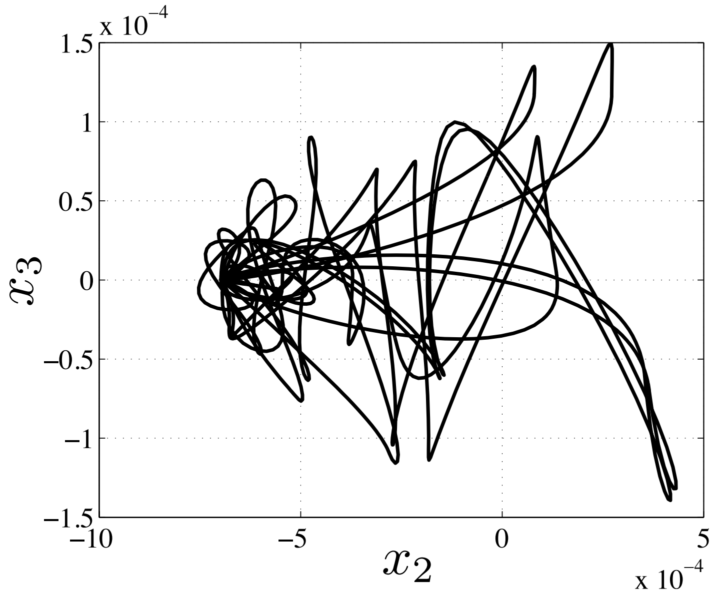
[Topputo&Bernelli-Zazzera 2011]

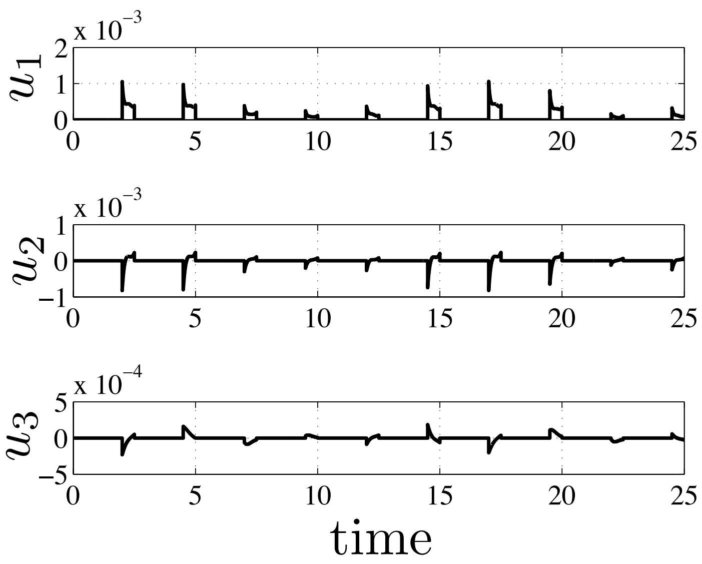
Stationkeeping: results



- Control cycle
  - Free drift (2.5 days)
  - Maneuver (0.5 day)

1 month simulation





- 1 year stationkeeping simulated in [Topputo&Bernelli-Zazzera 2011]

DINAMICA INNOVATING TECHNOLOGY

Final Remarks

- All the previous numerical techniques for optimization are eventually based on the **use of the Newton method**:
  - Direct methods ► Solution of the nonlinear system of equations related to the necessary conditions for optimality
  - Indirect methods ► Solution of the boundary value problem on the DAE system
  - Approx methods ► Solution of TVLQR, no need of first guess solution, but suboptimal

  They suffer of the same **disadvantages**:
  - **Local convergence**, i.e., they tend to converge to solutions close to the supplied first guesses
  - **Need of "good" first guesses** for the solution

➡ **Note**: global optimization is another matter!

DINAMICA
INNOVATING TECHNOLOGY

Selected references

- **Direct Methods**:
  - Enright, P.J., and Conway, B.A., Discrete Approximation to Optimal Trajectories Using Direct Transcription and Nonlinear Programming, *J. of Guid., Contr., and Dyn.*, 15, 994–1001, 1992
  - Betts, J.T., Survey of Numerical Methods for Trajectory Optimization, *J. of Guid., Contr., and Dyn.*, 21, 193–207, 1998
  - Betts, J. T., Practical Methods for Optimal Control Using Nonlinear Programming, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2001
  - Conway, B.A., Space Trajectory Optimization, Cambridge University Press, 2010
- **Indirect Methods**:
  - L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, The Mathematical Theory of Optimal Processes, John Wiley & Sons, New York, NY, USA, 1962
  - Bryson, A.E., Ho, Y.C., Applied Optimal Control, Hemisphere Publishing Co., Washigton, 1975
- **ASRE Method**:
  - Cimen, T. and Banks, S.P., Global optimal feedback control for general nonlinear systems with nonquadratic performance criteria, Systems and Control Letters, vol. 53, no. 5, pp. 327–346, 2004.