

AN AVERAGING OPTIMAL CONTROL TOOL FOR LOW-THRUST MINIMUM-TIME TRANSFERS

J. Fourcade, S. Geffroy, R.Epenoy
Centre National d'Etudes Spatiales
18 avenue Edouard Belin, 31401 Toulouse cedex 4, France
e-mail : Jean.Fourcade@cnes.fr, Sophie.Geffroy@cnes.fr,
Richard.Epenoy@cnes.fr

Abstract

Low thrust transfers, which are attractive for mass consumption, lead to long transfers with many revolutions around the Earth. Both minimum time and fuel saving strategy transfers can be easily formulated using the maximum principle of Pontryagin. However, the associated boundary value problems are numerically unstable and cannot be solved as these low-thrust transfers are modeled by so-called "rapidly rotating" problems. Moreover, the difficult problem of finding good initial estimation of the adjoint variables still remains. Two main techniques have been applied to solve these two problems. First averaging method is used to average the differential equations of the optimal problem over one orbital revolution. In eliminating rapid oscillations, the resulting problem is simpler to solve than the initial one. Secondly, the problem of adjoint variable estimation is solved using related but simpler transfer whose solution gives a good estimation of the adjoint variables. This paper describes a software, developed at CNES, where these techniques have been implemented to solve minimum time transfers. This software shows to be a powerful tool to solve most of Earth transfer problems. It is distributed by CNES as a Freeware. A low-thrust geostationary transfer is presented as a test case.

Key words: low-thrust transfer software, optimal control problem, averaging techniques.

INTRODUCTION

Low-thrust transfers become today of real interest as these transfers require less propellant and thus increase the begin of life mass of the satellite. However, these transfers require much longer time than classical ones and priority should be given to minimum-time strategy. Contrary to impulsional one, a low-thrust transfer needs continuous thrust to be controlled at each time. This leads to solve an optimal control problem so-called rapidly rotating because of the "rapid" angular revolutions compared to the "slow" orbit changes. Averaging techniques, whose principle consists in eliminating the rapid angular oscillations, show to be well adapted to solve this kind of optimal control problem (see [2],[3]).

In this paper, an operational tool based on averaging techniques is presented to compute minimum-time transfers, whatever the initial and final orbits, where the thrust is assumed bounded and spacecraft mass variable, leading to a non-constant acceleration.

The first part presents the equations of the control problem as they are implemented in the software. That way, the user should clearly understand how the software is organized and even modify or improve some part of it.

The second part can be considered as the user manual to run the software. The Man Machine Interface of the tool is described, as well as all interface files of the software. Some tricks to solve convergence problems are also given.

Finally, an application case is presented, corresponding to a GTO to GEO transfer.

LOW THRUST TRANSFER PROBLEM FORMULATION

To find the optimum control law the Pontryagin's maximum principle is applied. This leads to a Two Point Boundary Value Problem (TPBVP) which is numerical unstable and difficult to solve directly. Two ideas, taken from [1] are used to solve the problem.

First, the initial problem is replaced by an averaged problem where the right-hand sides of the differential equations are averaged over one orbital revolution. This smooths both the state and the adjoint state variables, which leads to a numerically better-conditioned problem.

Secondly, the difficult task to start the numerical resolution with a good estimate of the unknown parameters of the boundary value problem is solved using related but simpler problem. A simpler averaged problems is introduced: the so-called (P_2 -3D) problem: minimum-energy transfer where the control vector is the acceleration instead of the thrust and where the modulus of the acceleration is not constrained. The transfer time is free and the final angular variable fixed.

EQUATIONS OF MOTION

We use the classical non-singular equinoctial parameters, defined as

$$\begin{aligned} a & \\ e_x &= e \cos(\omega + \Omega) \\ e_y &= e \sin(\omega + \Omega) \\ h_x &= \tan(i/2) \cos(\Omega) \\ h_y &= \tan(i/2) \sin(\Omega) \\ L &= \omega + \Omega + \nu \end{aligned}$$

where L is the true longitude and ν the true anomaly. To take into account the true acceleration of the thrust, we must add the mass m as the seventh state variable. Its evolution is given by:

$$\frac{dm}{dt} = -\frac{\|\vec{u}\|}{g_e Isp}$$

where $\|\vec{u}\|$ is the thrust modulus, g_e the gravitational acceleration at sea-level, and Isp the specific impulse of the thruster. Let x be the vector of the five equinoctial parameters (a, e_x, e_y, h_x, h_y) and \vec{u} the thrust vector expressed in the tangential-normal orbital local frame. Then, the motion of the spacecraft is described by the Gauss's equations, which are:

$$\begin{cases} \frac{dx}{dt} = f(x, L) \frac{\vec{u}}{m} \\ \frac{dL}{dt} = g_o(x, L) + g_1(x, L) \frac{\vec{u}}{m} \end{cases}$$

with:

$$f(x, L) = \sqrt{\frac{x_1}{\mu}} \frac{A}{D} \begin{bmatrix} \frac{2x_1 BD}{A^2} & 0 & 0 \\ \frac{2(x_2 + \cos L)D}{B} & \frac{-C}{B} & -x_3(x_4 \sin L - x_5 \cos L) \\ \frac{2(x_3 + \sin L)D}{B} & \frac{E}{B} & x_2(x_4 \sin L - x_5 \cos L) \\ 0 & 0 & \frac{1}{2}(1 + x_4^2 + x_5^2) \cos L \\ 0 & 0 & \frac{1}{2}(1 + x_4^2 + x_5^2) \sin L \end{bmatrix}$$

and:

$$g_0(x, L) = \sqrt{\frac{\mu}{x_1^3} \frac{D^2}{A^3}}$$

$$g_1(x, L) = \begin{bmatrix} 0 & 0 & \sqrt{\frac{x_1}{\mu}} \frac{A}{D} (x_4 \sin L - x_5 \cos L) \end{bmatrix}$$

and where A, B, C, D and E are defined as follows:

$$A = \sqrt{1 - e_x^2 - e_y^2}$$

$$B = \sqrt{1 + 2e_x \cos L + 2e_y \sin L + e_x^2 + e_y^2}$$

$$C = 2e_x e_y \cos L - \sin L (e_x^2 - e_y^2) + 2e_y + \sin L$$

$$D = 1 + e_x \cos L + e_y \sin L$$

$$E = 2e_x e_y \sin L + \cos L (e_x^2 - e_y^2) + 2e_x + \cos L$$

Let us denote u_t, u_n, u_w the thrust coordinates (with u_t the tangential direction, u_n the normal direction and u_w the direction orthogonal to the orbit plane) and ξ and ψ the steering angles which describe the direction of the thrust vector with respect to the velocity vector and with respect to the orbital plane (see figure 1)

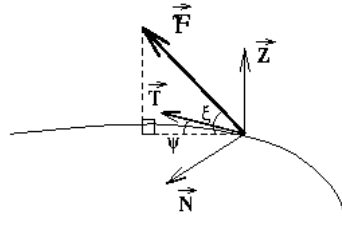


Figure 1: Direction of thrust vector in the tangential–normal orbital local frame

The components of the thrust vector are thus:

$$\begin{cases} u_t = \|\vec{u}\| \cos \xi \cos \psi \\ u_n = \|\vec{u}\| \cos \xi \sin \psi \\ u_w = \|\vec{u}\| \sin \xi \end{cases}$$

PROBLEM (P₁) FORMULATION

The spacecraft thrust is commanded continuously during the transfer in order to minimize the total transfer time. This command must satisfy a constraint on the thrust modulus, as assumed by low-thrust transfer. The set of admissible commands is thus characterized by:

$$U_{ad} = \left\{ \vec{u} \in L^2([t_0, t_1], \mathbb{R}^3) \mid \|\vec{u}(t)\| \leq u_{\max} \text{ a.e. } t \right\}$$

The minimum-time transfer problem without rendezvous in longitude can be formulated as follows:

$$(P_1) \left\{ \begin{array}{l} \min_{\vec{u} \in U_{ad}} t_1 \\ \frac{dx}{dt} = f(x, L) \frac{\vec{u}}{m} \\ \frac{dL}{dt} = g_o(x, L) + g_1(x, L) \frac{\vec{u}}{m} \\ \frac{dm}{dt} = -\frac{\|\vec{u}\|}{g_e Isp} \\ x(t_0) = x_0 \quad x(t_1) = x_1 \\ L(t_0) = 0 \quad L(t_1) \text{ free} \\ m(t_0) = m_0 \quad m(t_1) \text{ free} \\ t_1 \text{ free} \end{array} \right.$$

To solve this problem it is necessary first to scale the variables. The following scaling is used:

$$\begin{aligned} \underline{t} &= t \sqrt{\frac{\mu}{a_1^3}} \\ \underline{a} &= \frac{a}{a_1} \\ \underline{m} &= \frac{m}{m_0} \\ \underline{\vec{u}} &= \frac{\vec{u}}{u_{\max}} \end{aligned}$$

where a_1 is the semi-major axis of the final orbit and m_0 the initial mass. Introducing the small parameter:

$$\varepsilon = \frac{U_{\max} a_1^2}{m_0 \mu}$$

and considering the new definition of the five-state vector $\underline{x} = (a, e_x, e_y, h_x, h_y)$ the problem (P_1) falls into:

$$(P_1) \left\{ \begin{array}{l} \min_{\vec{u} \in U_{ad}} t_1 \\ \frac{d\underline{x}}{d\underline{t}} = \varepsilon \underline{f}(\underline{x}, L) \frac{\underline{\vec{u}}}{\underline{m}} \\ \frac{dL}{d\underline{t}} = \underline{g}_o(\underline{x}, L) + \varepsilon \underline{g}_1(\underline{x}, L) \frac{\underline{\vec{u}}}{\underline{m}} \\ \frac{d\underline{m}}{d\underline{t}} = -\delta \varepsilon \|\underline{\vec{u}}\| \\ \underline{x}(t_0) = \underline{x}_0 \quad \underline{x}(t_1) = \underline{x}_1 \\ L(t_0) = 0 \quad L(t_1) \text{ free} \\ \underline{m}(t_0) = 1 \quad \underline{m}(t_1) \text{ free} \\ t_1 \text{ free} \end{array} \right.$$

where:

$$\begin{aligned} \underline{f}(\underline{x}, L) &= \sqrt{\mu} f(\underline{x}, L) \\ \underline{g}_o(\underline{x}, L) &= \sqrt{\frac{1}{\mu}} g_o(\underline{x}, L) \\ \underline{g}_1(\underline{x}, L) &= \sqrt{\mu} g_1(\underline{x}, L) \\ \delta &= \sqrt{\frac{\mu}{a_1}} \frac{1}{g_e Isp} \end{aligned}$$

and the new set of admissible commands is:

$$\underline{U}_{ad} = \left\{ \underline{\bar{u}} \in L^2([t_0, t_1]; \mathfrak{R}^3) \quad \|\underline{\bar{u}}(t)\| \leq 1 \text{ a.e. } t \right\}$$

Let us consider now θ as the new independant variable defined as follows:

$$\theta = \varepsilon L$$

and let us introduce:

$$\tau = \varepsilon \underline{t}$$

Then, the problem P_j falls at the first order in ε :

$$(P_1) \left\{ \begin{array}{l} \min_{\underline{\bar{u}} \in \underline{U}_{ad}} \tau(\theta_1) \\ \frac{d\underline{x}}{d\theta} = \frac{f(\underline{x}, \frac{\theta}{\varepsilon})}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} \frac{\underline{\bar{u}}}{m} \\ \frac{d\tau}{d\theta} = \frac{1}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} \\ \frac{d\underline{m}}{d\theta} = - \frac{\delta \|\underline{\bar{u}}\|}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} \\ \underline{x}(0) = \underline{x}_0 \quad \underline{x}(\theta_1) = \underline{x}_1 \\ \tau(0) = \tau_0 \quad \tau(\theta_1) \text{ free} \\ \underline{m}(0) = 1 \quad \underline{m}(\theta_1) \text{ free} \\ \theta_1 \text{ free} \end{array} \right.$$

As the final scaled true latitude θ_1 is free, the numerical resolution of the two point boundary value problem requires the introduction of the independant variable $s \in [0,1]$ with:

$$\theta = s\theta_1$$

where $\theta_1(s)$ becomes a new state variable.

The (P_j) control problem can thus be finally expressed as:

$$(P_1) \left\{ \begin{array}{l} \min_{\underline{u} \in U_{ad}} \tau(1) \\ \frac{d\underline{x}}{ds} = \theta_1 \frac{f(x, \frac{s\theta_1}{\varepsilon})}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} \frac{\underline{u}}{m} \\ \frac{d\tau}{ds} = \theta_1 \frac{1}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} \\ \frac{d\underline{m}}{ds} = -\theta_1 \frac{\delta \|\underline{u}\|}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} \\ \frac{d\theta_1}{ds} = 0 \\ \underline{x}(0) = \underline{x}_0 \quad \underline{x}(1) = \underline{x}_1 \\ \tau(0) = \tau_0 \quad \tau(1) \text{ free} \\ \underline{m}(0) = 1 \quad \underline{m}(1) \text{ free} \\ \theta_1(0) \text{ free} \quad \theta_1(1) \text{ free} \end{array} \right.$$

Let us introduce the adjoint state vector $p^T = (p_{\underline{x}}^T, p_{\tau}, p_{\underline{m}}, p_{\theta_1})$ associated to the augmented state vector $z^T = (\underline{x}^T, \tau, \underline{m}, \theta_1)$. The Hamiltonian of this problem can be written as :

$$H(z, p, \underline{u}, s) = \frac{\theta_1}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} (-V^T \frac{\underline{u}}{m} - p_{\tau} + \delta p_{\underline{m}} \|\underline{u}\|)$$

where V is the following vector:

$$V = \underline{f}^T \cdot p_{\underline{x}}$$

The minimization of the Hamiltonian relatives to \underline{u} gives the optimal command:

$$\underline{u}^*(x, p_{\underline{x}}, s) = \frac{V}{\|V\|} \quad \text{with} \quad \|\underline{u}^*\| = 1$$

The adjoint state evolution is given by:

$$\frac{dp}{ds} = \frac{\partial H}{\partial z}$$

p_{τ} is constant because $\frac{\partial H}{\partial \tau} = 0$ (the Hamiltonian is time independant) and $p_{\tau}(1) = -\frac{\partial J}{\partial \tau} = -1$ where J is the cost function. Therefore $p_{\tau} = -1$

The minimized Hamiltonian becomes :

$$H^*(z, p, s) = \frac{\theta_1}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} (-\frac{\|V\|}{m} + 1 + \delta p_{\underline{m}})$$

The derivation of the Hamiltonian with respect to z leads to the following boundary value problem with consists of 14 equations :

$$\left\{ \begin{array}{l} \frac{d\underline{x}}{ds} = -\frac{\partial H^*}{\partial p_{\underline{x}}} \\ \frac{dp_{\underline{x}}}{ds} = \frac{\partial H^*}{\partial \underline{x}} \\ \frac{d\underline{m}}{ds} = -\theta_1 \frac{\delta}{\underline{g}_0} \\ \frac{dp_{\underline{m}}}{ds} = \theta_1 \frac{1}{\underline{g}_0} \frac{\|V\|}{\underline{m}^2} \\ \frac{d\theta_1}{ds} = 0 \\ \frac{dp_{\theta_1}}{ds} = \frac{\partial H^*}{\partial \theta_1} \end{array} \right.$$

The state equation of τ was dropped as it can be demonstrated that :

$$\tau = \frac{1-\underline{m}}{\delta}$$

The boundary conditions are given by the initial and final state vector \underline{x}_0 and \underline{x}_1 (10 variables) the initial mass $\underline{m}(0)$ and transversality conditions on the adjoint state:

- $p_{\underline{m}}(1) = 0$ since the final mass is free,
- $p_{\theta_1}(0) = p_{\theta_1}(1) = 0$ since $\theta_1(0)$ and $\theta_1(1)$ are free.

Thus the 7 initial unknown variables of the boundary problem are:

$$p_{\underline{x}}(0), p_{\underline{m}}(0), \theta_1(0)$$

PROBLEM (P_1) RESOLUTION

Let us apply the averaging method described in [1]. The problem (P_1) defined previously :

$$(P_1) \left\{ \begin{array}{l} \min_{\underline{u} \in \underline{U}_{ad}} \tau(1) \\ \frac{d\underline{x}}{ds} = \theta_1 \frac{f(x, \frac{s\theta_1}{\varepsilon})}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} \frac{\underline{u}}{m} \\ \frac{d\tau}{ds} = \theta_1 \frac{1}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} \\ \frac{dm}{ds} = -\theta_1 \frac{\delta \|\underline{u}\|}{\underline{g}_o(x, \frac{s\theta_1}{\varepsilon})} \\ \frac{d\theta_1}{ds} = 0 \\ \underline{x}(0) = \underline{x}_0 \quad \underline{x}(1) = \underline{x}_1 \\ \tau(0) = \tau_0 \quad \tau(1) \text{ free} \\ \underline{m}(0) = 1 \quad \underline{m}(1) \text{ free} \\ \theta_1(0) \text{ free} \quad \theta_1(1) \text{ free} \end{array} \right.$$

falls into the so-called rapidly rotating form:

$$(P_1) \left\{ \begin{array}{l} \min_{u \in U_{ad}} h(x(\tau_1)) \\ \frac{dx}{d\tau} = f(x(\tau), \frac{\tau}{\varepsilon}, u(\tau)) \\ x(\tau_0) = x_0 \quad g(\tau_1, x(\tau_1)) = 0 \\ \tau_1 \text{ free} \end{array} \right.$$

where f is 2π periodic in the fast movement τ/ε and where U_{ad} is defined as:

$$U_{ad} = \left\{ u \in L^2([t_0, t_1]; \mathfrak{R}^3) \mid \|u(\tau)\| \leq 1 \text{ a.e. } \tau \right\}$$

The associated averaged problem is:

$$(\overline{P}_1) \left\{ \begin{array}{l} \min_{v \in V_{ad}} h(X(\tau_1)) \\ \frac{dX}{d\tau} = \overline{f(x(\tau), \cdot, u(\tau, \cdot))} \\ X(\tau_0) = x_0 \quad g(\tau_1, X(\tau_1)) = 0 \\ \tau_1 \text{ free} \end{array} \right.$$

using the classical averaging notation for any function f ω -periodic in its first argument:

$$\overline{f(\cdot, \alpha)} = \frac{1}{\omega} \int_0^{\omega} f(\theta, \alpha) d\theta$$

and where V_{ad} is defined as:

$$V_{ad} = \left\{ v \in L^2([t_0, t_1], \mathfrak{R}^+; \mathfrak{R}^3) \mid \|v(\tau, \theta)\| \leq 1 \text{ a.e. } (\tau, \theta) \right\}$$

Notice that the averaged command depends explicitly on the rapid movement τ/ε contrary to the averaged state.

The (\overline{P}_1) problem provides a good approximation of the initial problem (P_1) as the error between the averaged and exact solutions decreases with the value of ε .

As fully detailed in [1], the method to obtain the Pontryagin's necessary conditions of problem (\bar{P}_1) is performed in three steps :

- Step 1: Minimization of the initial Hamiltonian which leads to the optimum control law;
- Step 2 : Derivation of the Hamiltonian with respect to x and p_x witch leads to the state and adjoint state equations;
- Step 3: Averaging of the differential equations.

Under certain assumptions (satisfied by the minimum time transfer for instance), it is possible to inverse the Hamiltonian derivation and minimization (step 2 and 3). This inversion is preferable when the Hamiltonien can be averaged analytically.

Therefore, the averaged boundary value problem which must be solved now, consists in the following equations, derived from the previous page:

$$\left\{ \begin{array}{l} \frac{d\bar{x}}{ds} = - \frac{\partial \bar{H}^*}{\partial \bar{p}_x} \\ \frac{d\bar{p}_x}{ds} = \frac{\partial \bar{H}^*}{\partial \bar{x}} \\ \frac{d\bar{m}}{ds} = - \theta_1 \frac{\bar{\delta}}{g_0} = - \theta_1 \delta \bar{x}_1^{3/2} \\ \frac{d\bar{p}_m}{ds} = \theta_1 \frac{1}{g_0} \frac{\|V\|}{m^2} \\ \frac{d\theta_1}{ds} = 0 \\ \frac{d\bar{p}_{\theta_1}}{ds} = \frac{\partial \bar{H}^*}{\partial \theta_1} = \frac{\bar{H}^*}{\theta_1} \end{array} \right.$$

where the average is performed relatively to L and with the same boundary conditions as in the initial problem.

PROBLEM (P_2-3D) RESOLUTION

The problem (P_2-3D) is a 3D transfer where the criterium to be minimized is the energy of the transfer which is porportional to the square of the acceleration. The control vector \bar{u} is here the acceleration and with the constraint $\|\bar{u}\| \leq u_{\max}$, the associated problem is:

$$(P_2-3D) \left\{ \begin{array}{l} \min_{\|\bar{u}\| \leq u_{\max}} \int_0^{t_1} \frac{1}{2} u^2 dt \\ \frac{dx}{dt} = f(x, L) \bar{u} \\ \frac{dL}{dt} = g_o(x, L) + g_1(x, L) \bar{u} \\ x(t_0) = x_0 \quad x(t_1) = x_1 \\ L(t_0) = L_0 \quad L(t_1) = L_1 \\ t_1 \text{ free} \end{array} \right.$$

Considering the scaling :

$$\underline{t} = t \sqrt{\frac{\mu}{a_1^3}}$$

$$\underline{a} = \frac{a}{a_1}$$

$$\underline{u} = \frac{u}{u_{\max}}$$

and introducing $\varepsilon = \frac{u_{\max} a_1^2}{\mu}$, the new independent variable $\theta = \varepsilon L$ and $\tau = \varepsilon \underline{t}$ the problem becomes :

$$(P_2 - 2D) \left\{ \begin{array}{l} \min_{\|\underline{u}\| \leq 1} \int_{\theta_0}^{\theta_1} \frac{1}{2} \frac{\underline{u}^2}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} d\theta \\ \frac{d\underline{x}}{d\theta} = \frac{f(\underline{x}, \frac{\theta}{\varepsilon})}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} \underline{u} \\ \frac{d\tau}{d\theta} = \frac{1}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} \\ \underline{x}(\theta_0) = \underline{x}_0 \quad \underline{x}(\theta_1) = \underline{x}_1 \\ \tau(\theta_0) = L_0 \quad \tau(\theta_1) \text{ free} \\ \theta_1 \text{ fixed} \end{array} \right.$$

Let us introduce the adjoint state vector $p^T = (p_x^T, p_\tau)$ associated to the augmented state vector $z^T = (\underline{x}^T, \tau)$. The Hamiltonian of this problem can be written as:

$$H(z, p, \underline{u}, \theta) = \frac{1}{g_o(\underline{x}, \frac{\theta}{\varepsilon})} \left(\frac{u^2}{2} - V^T \underline{u} - p_\tau \right)$$

where V is the vector defined as:

$$V = \underline{f}^T \cdot p_x$$

To solve this problem, we assume that the modulus of the acceleration is unconstrained. In such a case, the minimization of the Hamiltonian relatively to \underline{u} gives the optimal command:

$$\underline{u}^*(\underline{x}, p_x, \theta) = V$$

As p_τ is zero (the Hamiltonian and the cost function are time-independent), the minimized Hamiltonian becomes :

$$H^*(z, p, \theta) = -\frac{V^2}{2g_o(\underline{x}, \frac{\theta}{\varepsilon})}$$

The state equation of τ is dropped as p_τ is equal to zero and as $\tau(\theta_1)$ is free. Therefore the resulting boundary problem consists of 10 equations where the 5 initial unknown parameters $p_x(0)$ are set to zero as initial guess to solve the TPBVP.

PROBLEM (P_1) CONVERGENCE

As for the (P_2 -3D) problem, the derivation of the Hamiltonian problem is performed first and the averaging is performed using Gauss-Legendre quadrature algorithm. The unknown parameters are initialized as follow:

- the 5 value of p_x are taken from the previous (P_2 -3D),
- the initial value of p_m is taken to -1 as it can be demonstrated [1] that the optimal value is negative,
- the final longitude L_f is initialized with the value $L_f^{P_2-3D}$ obtained from the solution of (P_2 -3D).

The initial guess of L_f with is the independent variable must be sufficiently accurate, otherwise the boundary value problem fail to converge. This is the reason why a “tuning parameter” k was introduced in the software and the value of L_f is computed as:

$$L_f = \frac{L_f^{P_2-3D}}{k}$$

A good value of k is around 2 as the minnum-time strategy leads to less revolutions than the fuel-saving one. For some long transfers we need to increase this value up to 4 or 8.

DESCRIPTION OF MIPELEC SOFTWARE

The low-trust optimization tool developed by CNES, called MIPELEC, which stands for Satellite Positioning with Electric propulsion, is presented now.

GENERAL DESCRIPTION

MIPELEC is distributed by CNES as a freeware. The package distribution includes all the sources of the software, the executable file (mipelec.exe) and a Man Machine Interface (MMI) developed especially for this tool (mipelec.ihm). Executable files are compiled to work on SUN Work station.

The software uses the TPBVP shooting routine developed by CNES to solve the TPBVP problem. This subroutine is based on the NLEQ1 routine written by U. Nowak, L. Weimann (see [4]) and the RKSUITE routines written by R.W. Brankin, I. Gladwell and L.F. Shampine. Theses two last routines were taken from the NETLIB (<http://netlib.bell-labs.com>).

The computation of Hamiltonian and state equations were performed using MAPLE software. The MAPLE source code can be furnished on request.

MMI DESCRIPTION

The tool can be used with or without the MMI. The MMI is only a simpler way to enter the values of a given problem. The MMI is composed of three sub-window which are the control panel, the error text message and the data panel. The control panel includes buttons such as "run", "load", "save" and "print" which allow the user to run the optimization tool, load, save and print data into ASCII files. The error message informs the user of error that may occurred during the optimization process. The data panel allows the user to enter all the data needed to run a transfer.

The executable file of the optimal tool is mipelec.exe. It can be recompiled and built with the sources included in the distributed package. The executable file uses a unique ASCII file as input-output, named MIPELEC, where initial orbit, final orbit and adjoint variables are written. At the end of the run, if the program succeeds, the unknown variables of the boundary problem of the (P_i) or ($P2-3D$) problem are stored in the MIPELEC file. This allows the user to keep the results for future use. The executable file writes also the optimal trajectory in a file named resulP1 for the ($P1$) problem where the fields are defined as follow :

t (days), v (deg), a (km), e_x , e_y , h_x , h_y , p_a , p_{ex} , p_{ey} , p_{hx} , p_{hy} , m (kg), p_m , ξ (deg), ψ (deg)

an a file named resulP23D for the ($P2-3D$) problem where the fields are defined as follow :

t (days), v (deg), a (km), e_x , e_y , h_x , h_y , p_a , p_{ex} , p_{ey} , p_{hx} , p_{hy} , J , u/u_{max} , ξ (deg), ψ (deg)

v is the longitude (number of turn), ξ and ψ are the steering angles which describe the direction of the thrust vector and p denotes the adjoint state variables (see previous section).

The initial parameters that the user must enter to perform a run can be seen on figure 2 which is a snapshot of the MMI screen (this screen is formally identical to the content of the MIPELEC file).

Theses values are :

- Initial orbit, expressed in standard keplerian parameters with apogee and perigee altitude,
- Final orbit,
- Characteristics of the transfer including the thrust modulus, the specific impulse and the initial mass.
- Precision parameter used for the shooting routine.

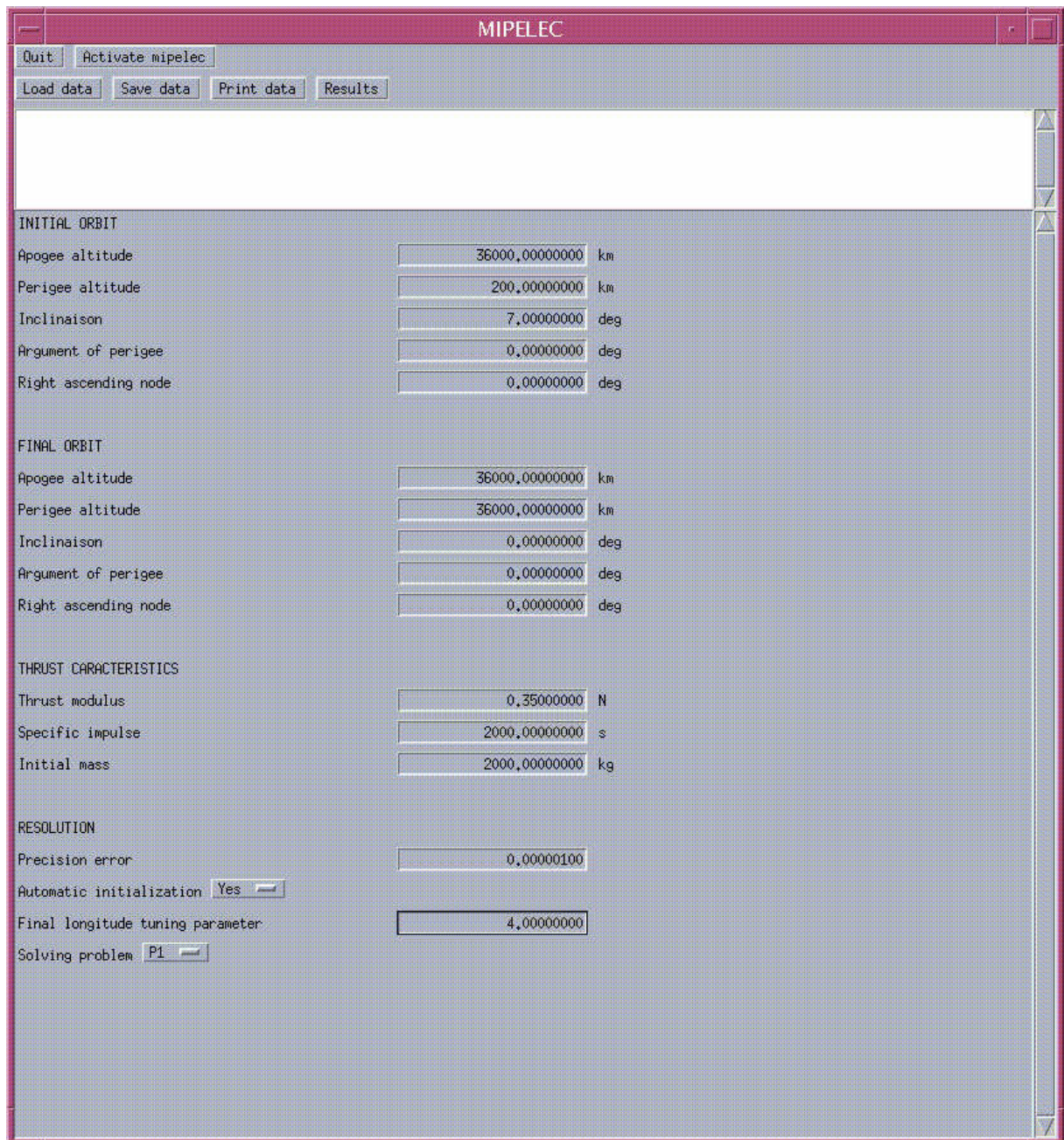


Figure 2 : MMI interface screen 1

Then the user has the choice of the problem ($P1$ or $P23D$) and can run the software with or without automatic mode.

- When the automatic mode is selected for the ($P1$) problem the software try to solve first the problem ($P2-3D$) and then the problem ($P1$). As, stated before, in this case there is a “tuning parameter”, named “Final longitude tuning parameter” which provides the user with different choices for initializing the final longitude of the ($P1$) problem.
- When the automatic mode is not selected, the user must enter initial guess for the unknown values of the concerned boundary problem as defined in the previous section.

The non-automatic mode screen of the MMI is represented in the following figure.

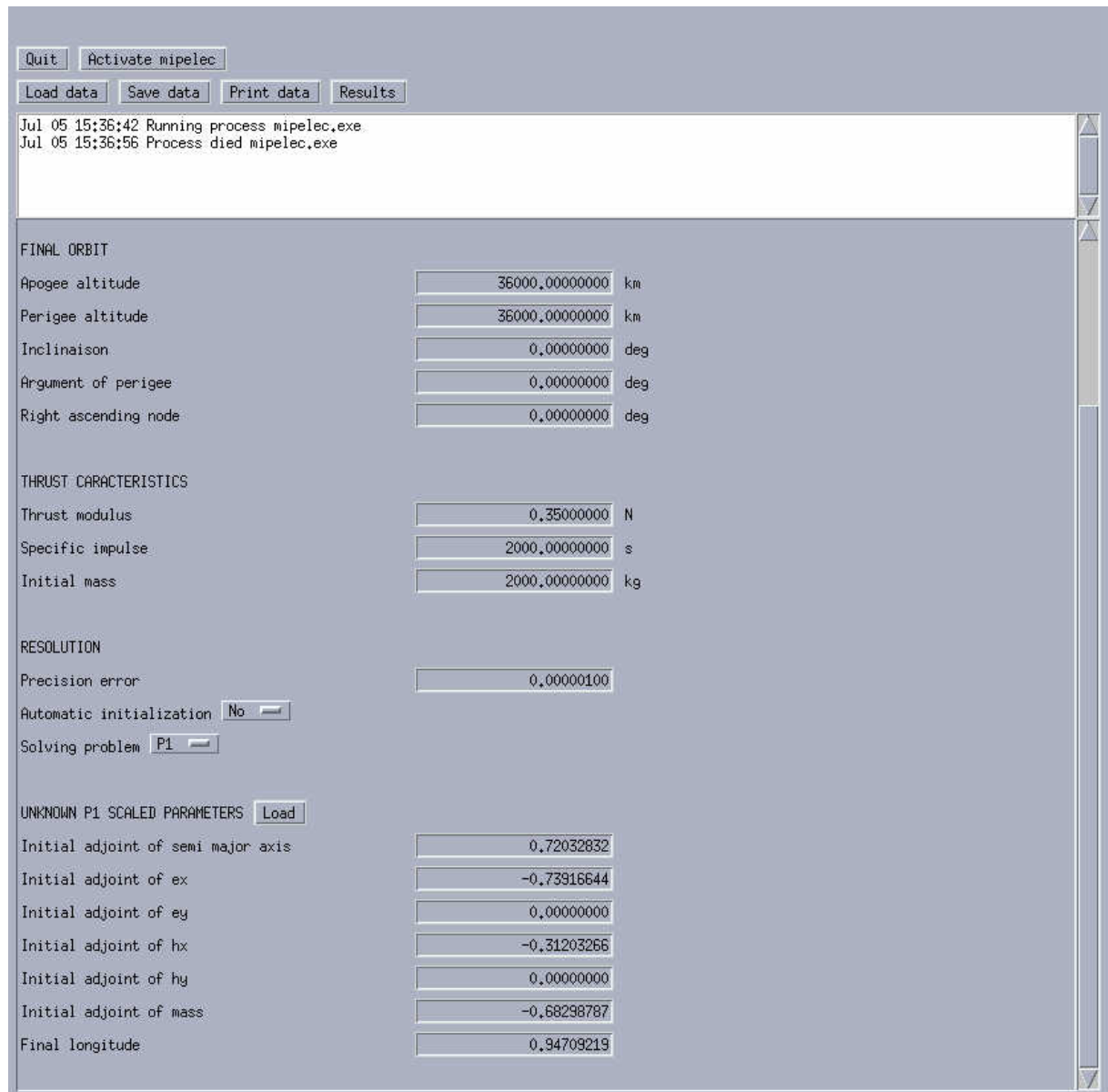


Figure 3: MMI interface screen 2

CONVERGENCE OF THE SOFTWARE TOOL

In most cases, the software converges when the automatic mode is selected.

In automatic mode, the (P_2-3D) may sometime fail if there is a big change in the inclination. In this case, the user shall try to solve a related but simpler problem. For instance, the same transfer with a smaller variation in inclination lead to a convergence of the (P_2-3D) problem.

Most of unsuccessful runs of the software come from the (P_1) problem. If the (P_1) problem fail the user may change the final longitude estimate with the tuning parameter. Usually two or three attempts are sufficient to find a good estimate. If the (P_1) problem persists to fail, there is no other solution that to try a related simpler problem.

When the user fail to solve his initial problem, but succeeds to solve a related simpler problem, the strategy is to run several times the software in non-automatic mode where the initial guess of the unknown parameters are taken from the previous successful run. That way the user can start an “homotopic” process to draw nearer step by step to his initial problem.

The MMI allows to perform rapidly these several runs without any data acquisition from the user, as it loads automatically adjoint variables from a previous successful run.

With some little practice, most of minimum-time transfers are rapidly solved.

APPLICATION TO GEOSTATIONARY MISSION

To give an example of a software run, let us consider the transfer between the standard Ariane4 Geostationary Transfer Orbit – apogee=36000 km, perigee=200 km, $i_0=7^\circ$ – to the Geostationary Orbit – altitude=36000 km, $e_1=i_1=0$. The thruster is supposed to be the SPT100 developed by SEP company with $F=0.35$ N and $I_{sp}=2000$ s. The initial mass of the satellite is supposed to be 2000 kg.

The minimum-time strategy leads to a transfer of 138.5 days with 191 revolutions and 213.5 kg of fuel.

The time evolution of the averaged trajectory is presented in Fig. 4 and the angular evolution of the command in Fig. 5. The strategy consists in raising the apogee higher than the geosynchronous orbit

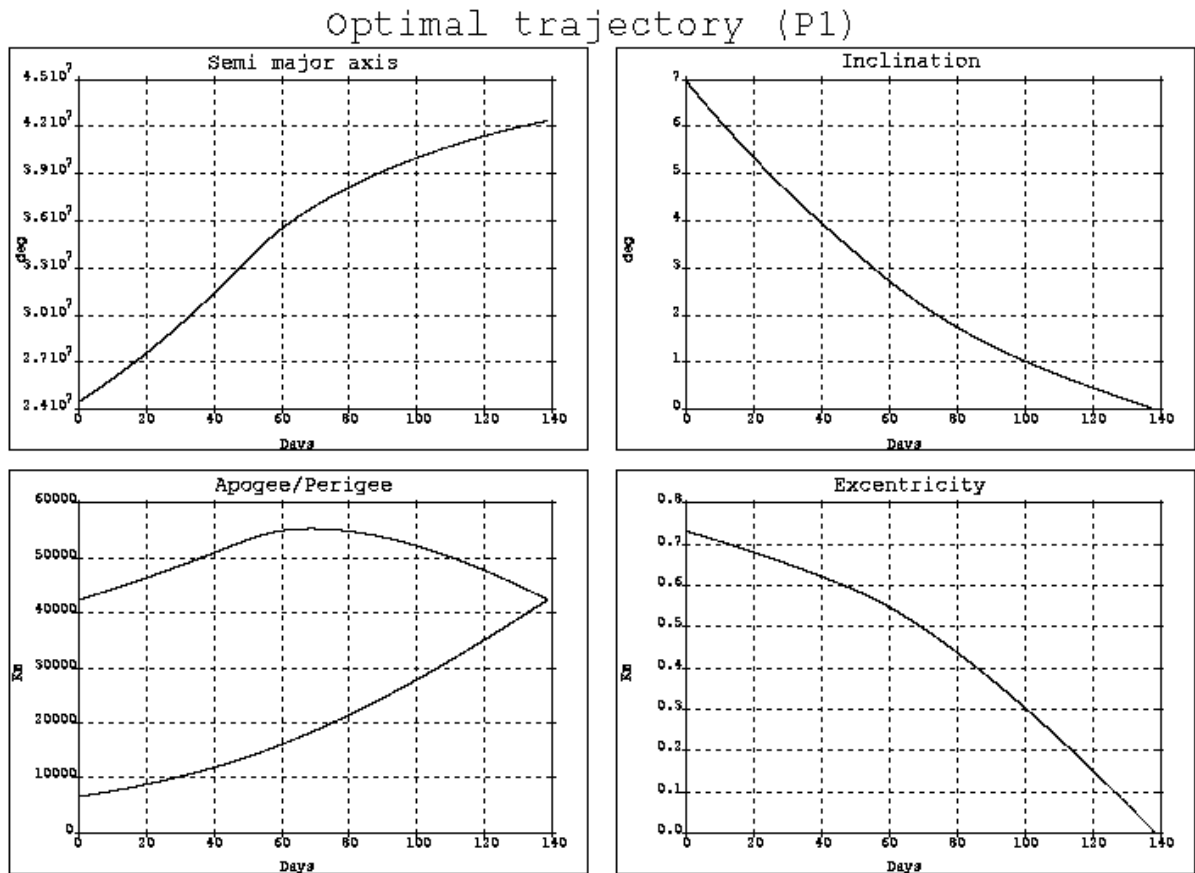


Figure 4: Trajectory evolution

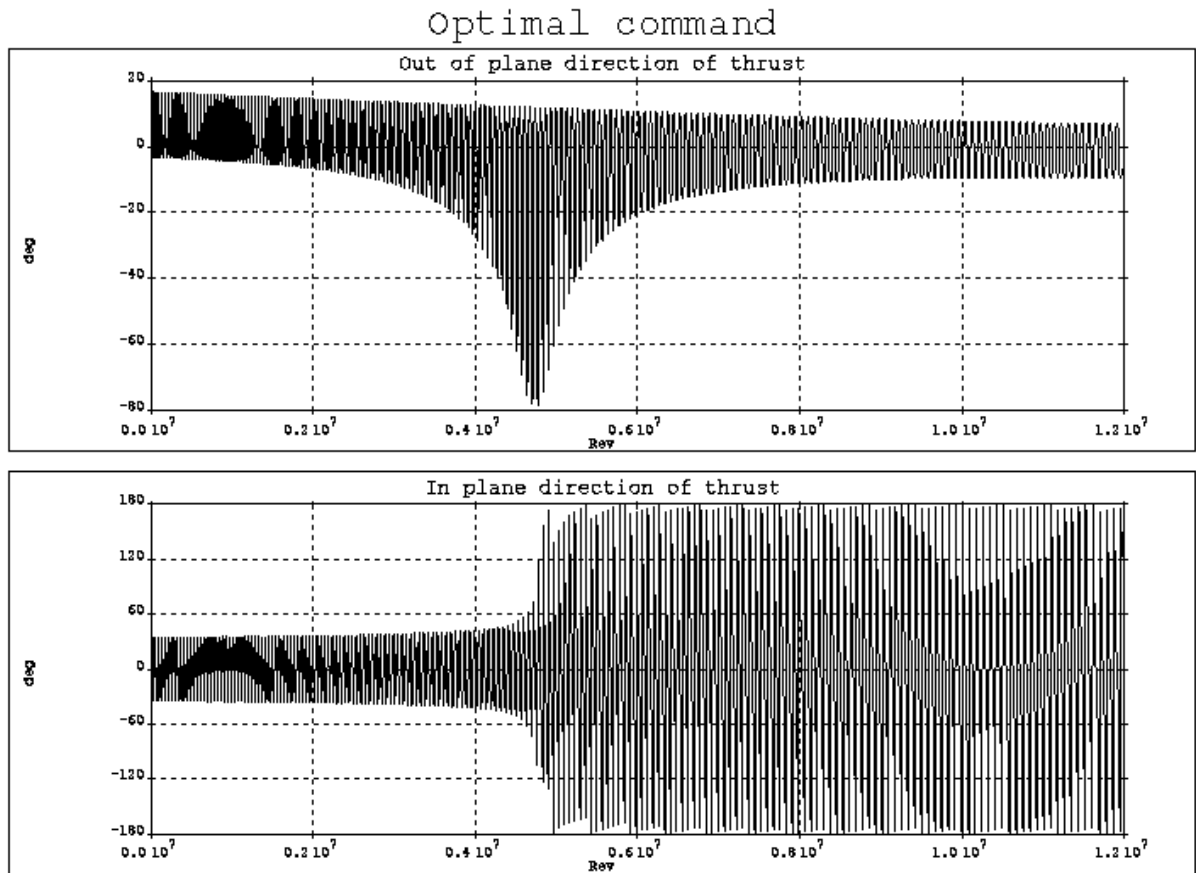


Figure 5: Optimum command evolution

REVIEW OF ASSUMPTIONS

In order to use the software with success for a particular analysis, the assumptions on which equations are based must be clearly understood.

The assumptions used are as follows:

- This tool leads to find optimal minimum-time transfer with bounded thrust and variable spacecraft mass.
- The value of the thrust modulus should not exceed, let us say, 1.10^{-3} m/s^2 . Not only the error between the averaged and exact solution increases with the value of the thrust, but also, in most cases, the software fail to converge for higher acceleration values.
- The equations of motion are modeled under the hypothesis of one central body without taking into account any perturbation force except the low-thrust one.
- The thrust direction is not constrained.
- There is no final rendezvous in longitude.
- No shadowing or power duty-cycle is considered.

CONCLUSION

This paper presents a tool for solving minimum-time low-thrust transfers, which is distributed by CNES as freeware. This software, based on averaging techniques in optimal control, proved to be a powerful tool to solve any low-thrust transfer and find optimal solution. The possibility to solve a transfer with automatic initialization of adjoint variables leads to find the optimal solution of the problem, even when its structure is unknown.

This low-thrust optimization tool is, moreover, rather flexible since recent studies have allowed several generalizations to be taken into account : J2-J6 Earth zonal effect and Moon-Solar potential, Earth Shadowing constraints, thrust direction constraint, rendezvous in longitude, fuel-saving strategy.

REFERENCES

1. Geffroy, S. Epenoy, R "Optimal low-thrust transfers with constraints-generalization of averaging techniques", Acta Astronautica, Vol. 41, No. 3, 1997
2. Kluever, C.A. and Oleson, S.R. "Direct approach for computing near-optimal low-thrust Earth-orbit transfers". Journal of Spacecraft and Rockets, Vol 35 Nov, 1998.
3. Edelbaum, T.N., Sackett, L.L. and Malchow, H.L., "Optimal low-thrust geocentric transfer", AIAA 73-1074, AIAA 10th Electric Propulsion Conference, Lake Tatiol, NE, 1973.
4. U. Nowak, L. Weimann. "A Family of Newton Codes for Systems of Highly Nonlinear Equations - Algorithm, Implementation, Application", ZIB, Technical Report TR 90-10 (December 1990)