

Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming

Camilla Colombo, Massimiliano Vasile, Gianmarco Radice

Department of Aerospace Engineering, University of Glasgow

James Watt South Building, G12 8QQ Glasgow, UK

E-mail: camilla.colombo@strath.ac.uk

Abstract

In this paper an optimisation algorithm based on Differential Dynamic Programming is applied to the design of rendezvous and fly-by trajectories to near Earth objects. Differential dynamic programming is a successive approximation technique that computes a feedback control law in correspondence of a fixed number of decision times. In this way the high dimensional problem characteristic of low-thrust optimisation is reduced into a series of small dimensional problems. The proposed method exploits the stage-wise approach to incorporate an adaptive refinement of the discretisation mesh within the optimisation process. A particular interpolation technique was used to preserve the feedback nature of the control law, thus improving robustness against some approximation errors introduced during the adaptation process. The algorithm implements global variations of the control law, which ensure a further increase in robustness. The results presented show how the proposed approach is capable of fully exploiting the multi-body dynamics of the problem; in fact, in one of the study cases, a fly-by of the Earth is scheduled, which was not included in the first guess solution.

Keywords

Numerical Methods; N-Body; Asteroids; Trajectory Optimisation; Optimisation methods; Optimal control; Low-thrust trajectories; Near Earth Objects; Differential Dynamic Programming; Bellman principle; Multi-Body Problem; Multi-revolution trajectory.

Nomenclature

a	coefficient matrix of the Runge-Kutta-Fehlberg integration scheme, or acceleration vector
A_k	matrix of the DDP algorithm at stage <i>k</i>
b	coefficient matrix of the Runge-Kutta-Fehlberg integration scheme
B_k	matrix of the DDP algorithm at stage <i>k</i>
<i>c</i>	constant between 0 and 1
c	coefficient matrix of the Runge-Kutta-Fehlberg integration scheme
C_k	matrix of the DDP algorithm at stage <i>k</i>
D_k	matrix of the DDP algorithm at stage <i>k</i>
E_k	matrix of the DDP algorithm at stage <i>k</i>
f	discrete-time state transition function
$\tilde{\mathbf{f}}$	function containing the continuous dynamics equations
<i>g</i>	scalar stage-wise loss function
<i>h_k</i>	discretisation step
H_k	matrix of the DDP algorithm at stage <i>k</i>
I(<i>m</i>)	identity matrix of size <i>m</i>
<i>I_{sp}</i>	specific impulse of the spacecraft engine
<i>j</i>	integer number
<i>J</i>	cost function of the minimisation problem
<i>k</i>	integer number indicating the generic stage of DDP and the decision time of the trajectory on which the control law is allowed to change
<i>k_{lim}</i>	state from which the new control law is adopted for the integration of the dynamics
K_k	matrix of the DDP algorithm at stage <i>k</i>
<i>l</i>	number of components of the Lagrange multiplier vector
<i>m</i>	number of components of the control vector, or mass of the spacecraft
<i>n</i>	number of components of the state vector
<i>N</i>	total number of decision times of control stages
P_k	matrix of the DDP algorithm at stage <i>k</i>
Q_k	matrix of the DDP algorithm at stage <i>k</i>

\mathbf{r}	position vector
R_{Earth}	radius of the Earth
$reltol$	relative tolerance
$reltol_{mesh}$	relative tolerance on the mesh selection
\mathbf{R}_k	matrix of the DDP algorithm at stage k
\mathbf{s}	state vector
\mathbf{S}_k	matrix of the DDP algorithm
t	time
\mathbf{T}	thrust vector
tol_r	absolute tolerance of the position error
tol_v	absolute tolerance on the velocity error
\mathbf{u}	control vector
\mathbf{v}	velocity vector
V	optimal return function
w	weight parameter
\tilde{w}	weight parameter
x	Cartesian coordinate along the x axis
y	Cartesian coordinate along the y axis
z	Cartesian coordinate along the z axis
\mathbf{Z}_k	matrix of the DDP algorithm at stage k
α	in-plane angle of the velocity vector with respect to the Earth inertial reference frame
$\boldsymbol{\beta}$	coefficient vector of the feedback control law component proportional to the variation of the state vector
$\boldsymbol{\gamma}$	coefficient vector of the feedback control law component proportional to the variation of the Lagrange multiplier vector
δ	out-of plane angle of the velocity vector with respect to the Earth inertial reference frame
ε	constant between 0 and 1
$\boldsymbol{\lambda}$	vector of Lagrange multipliers
μ_{Earth}	gravitational constant of the Earth
μ_{Sun}	gravitational constant of the Sun

φ	scalar function representing the constraints on the final stage
Θ_k	difference between the optimal return function at state k applying the new control, and the optimal return function at state k applying the nominal control

Subscripts

1	initial condition of a variable
k	stage of the DDP procedure
<i>out</i>	threshold value to exit a computational loop
<i>target</i>	variable related to the target body
x	vector component along the Cartesian x axis
y	vector component along the Cartesian y axis
z	vector component along the Cartesian z axis

Superscripts

*	new nominal control for the algorithm with global variation in control
k	stage of the DDP procedure

Mathematical notations

\square	variable
$\bar{\square}$	nominal value of \square
$\{\square_k\}$	sequence of variable \square in time
\square^T	transposed
$\delta\square$	differential variation of \square
$\Delta\square$	finite difference variation of \square
$QP[\square]$	linear quadratic part of the Taylor expansion of the function \square
\square_s	gradient of the scalar function \square , or Jacobian of the vector function \square with respect to the state \mathbf{s}
\square_{ss}	block components of the Hessian matrix of the scalar or the vector function \square with respect to the state \mathbf{s}
\square_u	gradient of the scalar function \square , or Jacobian of the vector function \square with respect to the state \mathbf{u}

- \square_{uu} block components of the Hessian matrix of the scalar or the vector function \square with respect to the state \mathbf{u}
- $\frac{d\square}{dt}$ derivative of \square over time
- \leftarrow assignment (in an algorithm)
- $\|\square\|_\infty$ norm infinity of the vector \square

1 Introduction

Asteroids are nowadays appealing targets for space missions (Perozzi et al., 2002). As primordial remnants of our solar system, they preserve precious information about its formation; besides, their collision with the early Earth would have influenced the shape and composition of our planet.

The orbit of those asteroids numbered among the near Earth objects comes close to the Earth orbit around the Sun; this makes their exploration viable with the current technologies. In particular, as testified by some missions like Dawn* and Hayabusa†, the employment of low-thrust propulsion proved in the last decade to be a valuable option to decrease propellant consumption, at the expense of longer times of flight.

The design of low-thrust trajectories requires the solution of an optimal control problem, the difficulty of which increases with the complexity of the transfer and the fidelity of the trajectory model. Multi-body dynamics, gravity assist manoeuvres, capture or escape phases concur to increase the complexity of a trajectory design problem (Racca, 2003). Furthermore, the low level of thrust implies long transfer times and a low control authority because the thrust level is comparable with the gravitational forces. Moreover, the design of interplanetary transfers involves dynamics of variable scales, i.e., from planetocentric phases (e.g., during gravity assist manoeuvres) to heliocentric legs.

In order to properly handle the different scales, it would be desirable to have an optimisation method that can adaptively change the discretisation of the numerical integration of the dynamics, during the optimisation itself. Additionally, it should

* <http://dawn.jpl.nasa.gov/>

† <http://www.muses-c.isas.ac.jp/>

be robust enough to converge even when a poor first guess solution is available and accurate enough to reproduce the trajectory with high fidelity, hence exploiting a full dynamical model.

In general, methods for trajectory optimisation are classified under direct or indirect approaches (Betts, 1998). Direct methods are known to be quite robust, convergence being reached even if a poor first guess solution is available; however collocation method efficacy is bounded by the definition of the discretisation of the state variables prior to the optimisation (Conway et al., 2007; Betts and Erb, 2003; Enright and Conway, 1991). Direct shooting methods overcome the disadvantage of collocating the states, but still need the a priori collocation of the control (Scheel and Conway, 1994; Kluever, 1997) and tend to be less robust than collocation methods.

On the contrary, indirect methods guarantee the accuracy of the solution, which satisfies Pontryagin maximum principle (Pontryagin et al., 1962), but, on the other hand, they require a good first guess for the adjoint variables. Common applications usually focus on a single phase of the mission, in which the primary body does not change, such as Earth centred transfers (Ranieri and Ocampo, 2006) or heliocentric leg (Colasurdo and Casalino, 1999; Casalino et al., 1999).

When direct and indirect methods are applied to the design of transfers which involve multi-body dynamics (i.e., include escape and capture phases) or gravity assist manoeuvres (not simplified as impulsive change of velocity), a patched conic approach is usually adopted. The overall trajectory is divided in a sequence of problems, each of them expressed in the primary body reference frame; different segments are then patched together, through boundary constraints at the edge of each segment (direct methods), or through conditions on states and costates (indirect methods). Many applications have been presented, making use of direct methods (Tang and Conway, 1995; Herman and Spencer, 2002; Vasile and Bernelli, 2003), indirect methods (Guellman, 1995; Vadali et al., 2000; Nah et al., 2001; Ranieri and Ocampo, 2005), or hybrid methods (Pierson and Kluever, 1994; Kluever and Pierson, 1995).

The patched conic approach allows handling different scales once at the time, in different segments of the trajectory, hence avoiding numerical sensitivity; however, since the transition conditions from one segment to the following one

are defined a priori, the solution may not fully exploit the multi-dynamics nature of a transfer.

Previous works attempted to optimise multi-body low-thrust problems, treating the trajectory as a whole, without resorting to the patched conic approach; Whiffen et al. presented many interplanetary trajectories, including escape, capture and fly-bys, computed with the Static/Dynamic Control (SDC) algorithm (Whiffen and Sims, 2001; 2002), Lantoine and Russel (2008) proposed a hybrid differential dynamic programming algorithm and applied it to a LEO to GEO orbital transfer and Olympio (2008) developed a gradient-based method to address the problem of interplanetary transfers with escape and captures.

In this paper we investigate the use of Differential Dynamic Programming (DDP) (Jacobson and Mayne, 1969) for designing interplanetary trajectories to the rendezvous and fly-by of near Earth objects, including the escape phase of the Earth. This technique can be classified among direct methods, but, unlike the other approaches, the time dependence is not removed from the parameterisation.

DDP is derived from the theory of dynamic programming (Bellman, 1957), and overcomes its inherent “curse of dimensionality” (Yakowitz and Rutherford, 1984) by replacing the cost function of the problem with its quadratic expansion in the neighbourhood of a nominal non-optimal trajectory. The optimisation process bases on successive iterations, in which the coefficients for a feedback control law are generated through the stage-wise solution, backward in time, of Bellman partial differential equation, and the consequent improved trajectory and control policy are then propagated forward in time.

Because the minimisation is performed through successive approximations around a nominal solution, the large scale problem, associated with the optimisation of a low-thrust trajectory, is translated into a series of problems of small dimensions. In other words, the stage-wise approach allows efficiently handling problems with a large number of stages; this overcomes the limit of direct transcription methods, which lead to the solution of systems of nonlinear algebraic equations of increasing dimension with the number of discretisation steps (or stages). For example the solution representative of SMART-1 mission, computed by Betts and Erb (Betts and Erb, 2003) required the solution of a sparse optimisation problem with 211031 variables and 146285 constraints.

Moreover, DDP is based on Bellman's principle of optimality which is a necessary and sufficient condition for a solution to be locally optimal (Bertsekas, 2005); hence the solution of the optimal control problem preserves the accuracy as indirect methods, without requiring a first guess solution for the adjoint variables. In this work, we exploited the stage-wise feature of DDP to integrate an adaptive variable step discretisation scheme within the optimisation process. The discretisation grid is adjusted at each iteration, to better adapt to the non-linear dynamics of the problem. A Runge-Kutta explicit method was selected for the numerical integration and the derivatives of the dynamics scheme were analytically derived. The stage-wise approach also allows handling a multi-phase trajectory in a whole, without recurring to the patched conic approximation. The algorithm developed applies global variation of control (Jacobson and Mayne, 1969), through the use of DDP and non-linear programming techniques. The constraints on the target state at the end of the trajectory are included in the optimisation problem as an additional term of the cost objective, through a time invariant vector of Lagrange multipliers, whose value is modified along the convergence process (Gershwin and Jacobson, 1970).

The paper presents an analysis of some mission opportunities for the rendezvous and fly-by of a selected number of asteroids; some solutions with a long time of flight will be presented. The classical DDP approach is introduced in paragraph 2, while sections 3 and 4 present the modified method, which was adopted for designing trajectories to asteroids; some cases will be shown in section 5.

2 Differential Dynamic Programming

Differential dynamic programming, firstly introduced by Jacobson and Mayne in 1969, is a successive approximation technique for finding the optimal control of a non-linear system. It overcomes the issue of dimensionality linked to dynamic programming (Bellman, 1957), introducing in the optimisation process a linear-quadratic approximation of the cost function in the neighbourhood of the nominal trajectory.

Given a nominal control strategy, each iteration of DDP produces, through the backward propagation of the difference Bellman equation, a feedback control strategy which is forward propagated, to give an improved trajectory and a

reduction in the cost function. The control laws, produced within successive iterations, approach the optimal control solution of the problem.

2.1 Differential dynamic programming for trajectory optimisation

The standard DDP technique works with two variable classes: the system state vector $\mathbf{s}(t)$ and the dynamic control vector $\mathbf{u}(t)$.

A low-thrust trajectory is characterised by a continuous-time dynamics. However, for solving the low-thrust optimisation problem through DDP, the discrete-time approach is usually used; the continuous-time problem is transcribed in a discrete-time system and approximated by difference equations. Given a sequence of controls $\{\mathbf{u}_k\}_{k=1}^N$, the resulting trajectory $\{\mathbf{s}_k\}_{k=1}^{N+1}$ is computed by the recursive formula:

$$\begin{aligned} \mathbf{s}_{k+1} &= \mathbf{f}(\mathbf{s}_k, \mathbf{u}_k; t_k) & k = 1, \dots, N \\ \mathbf{s}_1 &= \bar{\mathbf{s}}_1 \end{aligned} \quad (1)$$

where $\bar{\mathbf{s}}_1$ is the initial condition at time t_1 , which is assumed fixed and \mathbf{f} is the discrete-time state transition function, which expresses the state vector at time $k+1$ as a function of state and control vector at the previous time step. We define $k=1, \dots, N$ as the stages of this problem, i.e., the decision times over which the control law is allowed to change.

The optimisation problem is described by a cost function to be minimised; we define the cost function of a trajectory with initial condition $\bar{\mathbf{s}}_1$ and control schedule $\{\mathbf{u}_k\}_{k=1}^N$ as:

$$J(\{\mathbf{u}_k\}; \bar{\mathbf{s}}_1) = \sum_{k=1}^N g(\mathbf{s}_k, \mathbf{u}_k; t_k) \quad (2)$$

where g represents the scalar stage-wise loss function of $(\mathbf{s}_k, \mathbf{u}_k; t_k)$. Eq. (2) corresponds to the integral term of the cost function for the continuous-time problem. The optimisation problem is to determine the sequence of control $\{\mathbf{u}_k\}_{k=1}^N$ that minimises Eq. (2) under certain constraints. The constraints considered at this point are equality constraints on the final state:

$$\varphi(\mathbf{s}_{N+1}; t_{N+1}) = 0 \quad (3)$$

where the final time t_{N+1} is supposed to be given explicitly. The constrained optimisation is converted into an unconstrained one by including Eq. (3) into the cost function in Eq. (2), through a time invariant set of Lagrange multipliers λ (Gershwin and Jacobson, 1970):

$$J(\{\mathbf{u}_k\}; \bar{\mathbf{s}}_1) = \sum_{k=1}^N g(\mathbf{s}_k, \mathbf{u}_k; t_k) + \lambda^T \cdot \varphi(\mathbf{s}_{N+1}; t_{N+1}) \quad (4)$$

If we try to minimise Eq. (4) through dynamic programming, we need to apply Bellman's principle of optimality for discrete-time systems (Jacobson and Mayne, 1969):

$$V_k(\mathbf{s}_k) = \min_{\mathbf{u}_k} [g(\mathbf{s}_k, \mathbf{u}_k; t_k) + V_{k+1}(\mathbf{s}_{k+1})] \quad (5)$$

Eq. (5) gives the optimal return function at stage k , $V_k(\mathbf{s}_k)$, defined as the cost $J(\{\mathbf{u}_k\}; \mathbf{s}_k)$ associated to the segment of the trajectory starting at point \mathbf{s}_k , if the optimal control policy is employed (see Fig. 1).

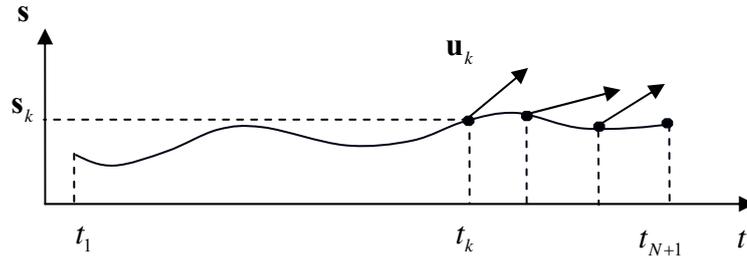


Fig. 1 Dynamic programming approach

The value of $V_k(\mathbf{s}_k)$ results from the minimisation of the optimal return function at stage $k+1$ added to the term of the k -stage-wise loss function g . Starting from the final condition at the end-point of the trajectory:

$$V_{N+1}(\mathbf{s}_{N+1}) = \lambda^T \cdot \varphi(\mathbf{s}_{N+1}; t_{N+1})$$

dynamic programming requires the solution of Eq. (5) from stage N backward until stage 1. The limitation of dynamic programming for continuous problem is the high dimensional problem resulting from the application of Eq. (5) to every stage k . In fact this is equivalent to find a family of optimal solutions, one from each different initial point $\mathbf{s}_k, k = 1, \dots, N$.

In order to overcome this computational limitation, *differential* dynamic programming, applies the principle of optimality in the neighbourhood of a nominal trajectory. At each stage k , the full expression of the stage-wise cost function g and the optimal return function from the next iteration onward V_{k+1} are replaced by their quadratic approximation about the current nominal control and trajectory.

The state and control vectors at each discretisation step can be written as a variation from their nominal values:

$$\begin{aligned}\mathbf{s}_k &= \bar{\mathbf{s}}_k + \delta\mathbf{s}_k \\ \mathbf{u}_k &= \bar{\mathbf{u}}_k + \delta\mathbf{u}_k\end{aligned}\quad (6)$$

where the superscript dash indicates the nominal conditions. With this notation, $\{\bar{\mathbf{u}}_k\}_{k=1}^N$ is the nominal control profile and $\{\bar{\mathbf{s}}_k\}_{k=1}^{N+1}$ the corresponding trajectory, obtained by the integration of Eqs. (1) under the nominal control $\{\bar{\mathbf{u}}_k\}_{k=1}^N$.

Said $QP[\cdot]$ the linear and quadratic part of the Taylor expansion of a generic function, differential dynamic programming reduces Eq. (5) to:

$$V_k(\bar{\mathbf{s}}_k + \delta\mathbf{s}_k) \simeq \min_{\delta\mathbf{u}_k} \left[QP \left[g(\bar{\mathbf{s}}_k + \delta\mathbf{s}_k, \bar{\mathbf{u}}_k + \delta\mathbf{u}_k; t_k) + V_{k+1}(\bar{\mathbf{s}}_{k+1} + \delta\mathbf{s}_{k+1}) \right] \right] \quad (7)$$

Similarly to the procedure for solving Eq. (5), the solution of Eq. (7) is performed *backward* in time, from the final stage N until the initial stage 1, the boundary condition at t_{N+1} being:

$$V_{N+1}(\bar{\mathbf{s}}_{N+1} + \delta\bar{\mathbf{s}}_{k+1}) \simeq QP \left[\boldsymbol{\lambda}^T \cdot \boldsymbol{\varphi}(\bar{\mathbf{s}}_{N+1} + \delta\bar{\mathbf{s}}_{k+1}; t_{N+1}) \right]$$

The necessary requirement is that the new control sequence should produce small variations in the state vector such that the linear-quadratic approximation in Eq. (7) holds true. This may be achieved even with a big variation in the control action, as long as the time duration of this variation is small. This means that the new control \mathbf{u}_k does not need to be restricted to the neighbourhood of $\bar{\mathbf{u}}_k$, therefore the second of Eqs. (6) can be modified as follows:

$$\mathbf{u}_k = \mathbf{u}_k^* + \delta\mathbf{u}_k \quad (8)$$

where the global variation in the nominal control $\bar{\mathbf{u}}_k$ to \mathbf{u}_k^* is computed by minimising Eq. (7), where the nominal trajectory $\bar{\mathbf{s}}_k$ is substituted:

$$V_k(\bar{\mathbf{s}}_k) \approx \min_{\mathbf{u}_k^*} \left[QP \left[g(\bar{\mathbf{s}}_k, \mathbf{u}_k; t_k) + V_{k+1}(\bar{\mathbf{s}}_{k+1}) \right] \right] \quad (9)$$

Therefore the linear-quadratic expansion of Eq. (5) is now evaluated about the point $(\bar{\mathbf{s}}_k, \mathbf{u}_k^*)$:

$$V_k(\bar{\mathbf{s}}_k + \delta \mathbf{s}_k) \approx \min_{\delta \mathbf{u}_k} \left[QP \left[g(\bar{\mathbf{s}}_k + \delta \mathbf{s}_k, \mathbf{u}_k^* + \delta \mathbf{u}_k; t_k) + V_{k+1}(\bar{\mathbf{s}}_{k+1} + \delta \mathbf{s}_{k+1}) \right] \right] \quad (10)$$

This hypothesis was implemented in an algorithm that employs global variations in the control, hence strong variations in the state (Jacobson and Mayne, 1969; Gerswin and Jacobson, 1970).

The necessary condition to minimise the right hand side of Eq. (10) is to set its first derivative to zero. This leads to the definition of a feedback strategy of the form:

$$\delta \mathbf{u}_k = \boldsymbol{\beta}_k \cdot \delta \mathbf{s}_k \quad (11)$$

The variation in control is expressed as a function proportional to the state variation. Eqs. (9) and (10) are computed backward in time for every stage $k = N, \dots, 1$ and the coefficient $\boldsymbol{\beta}_k$ is constructed and stored in memory.

At this point, the trajectory is swept *forward* in time, for every stage $k = 1, \dots, N$: the successor control policy \mathbf{u}_k is constructed and the new trajectory is propagated through the state transition function \mathbf{f} , with the initial condition $\bar{\mathbf{s}}_1$:

$$\begin{cases} \mathbf{u}_k = \mathbf{u}_k^* + \boldsymbol{\beta}_k (\mathbf{s}_k - \bar{\mathbf{s}}_k) \\ \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{u}_k; t_k) \\ \mathbf{s}_1 = \bar{\mathbf{s}}_1 \end{cases} \quad k = 1, \dots, N$$

A posteriori we need to verify that the variations of the control do not break the assumption of linear-quadratic approximations in Eq. (10). To this purpose, a method was proposed by Jacobson and Mayne in 1969 and later refined by Gerswin and Jacobson in 1970, to determine the section of the trajectory over which the new control strategy can be applied.

The nominal control is applied over an initial segment of the trajectory, up to step k_{lim} , afterwards the new strategy is adopted:

$$\mathbf{u}_k = \begin{cases} \bar{\mathbf{u}}_k & k = 1, \dots, k_{\text{lim}} - 1 \\ \mathbf{u}_k^* + \boldsymbol{\beta}_k \delta \mathbf{s}_k & k = k_{\text{lim}}, \dots, N \end{cases} \quad (12)$$

The resulting control law and the associated trajectory are represented respectively in Fig. 2 and Fig. 3:

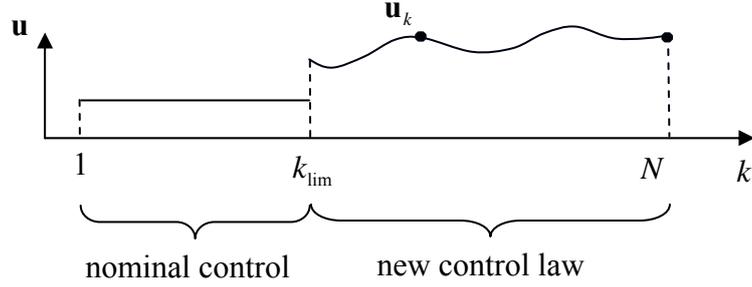


Fig. 2 Control law schedule according to Jacobson's algorithm

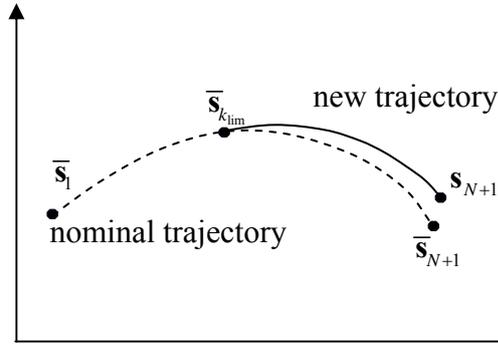


Fig. 3 Trajectory associated to the control law in Eq. (12)

The guess value of k_{lim} is initially set to 1 and is progressively increased, until an improvement in the value of the cost function $J(\{\mathbf{u}_k\}; \bar{\mathbf{s}}_1)$, with respect to its nominal value $J(\{\bar{\mathbf{u}}_k\}; \bar{\mathbf{s}}_1)$ is registered. This procedure is called *step-size adjustment method*.

In summary, the core of the DDP technique consists in a backward recursion followed by a forward recursion. A nominal trajectory and control policy are required as input and an improved control law and trajectory are provided as output, which ensures a decrease of the value of the cost function. Successive iterations of the backward and forward recursions produce control laws that progressively approximate the optimal control of the problem. Fig. 4 depicts the history of the control magnitude during the convergence process for a direct transfer from Earth to Mars. The value of k_{lim} selected at the first iteration of the

algorithm is close to the number of discretisation steps N and tends to 1 as convergence is reached.

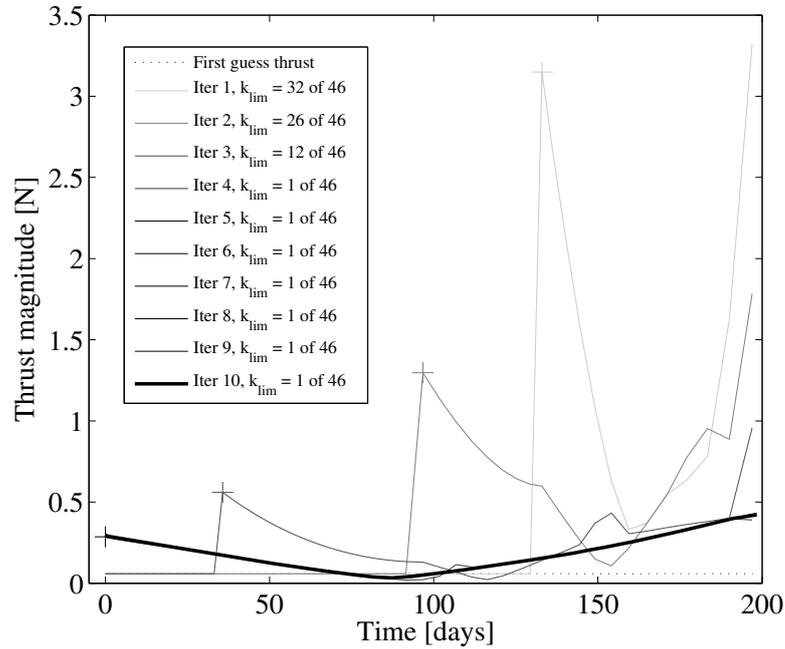


Fig. 4 Control law during the convergence process. Direct transfer Earth to Mars, with a time of flight of 200 days.

The algorithm has quadratic convergence under the assumption that the Hessian matrix of the cost function is positive definite (Murray, 1978; Murray and Yakowitz, 1984).

In the following subsections, the fundamental DDP algorithm is derived, in the case of end-point equality constraints. The purpose is to give a concise exposition of the original method upon which the one proposed here (see section 3) is based on. We report it here because the algorithm derivation is useful to understand the algorithm itself and in order to summarise some part of the theory, presented in different references: the algorithm derivation with global control variations by Jacobson and Mayne (1969), the end-point constraints algorithm by Gershwin and Jacobson (1970) and the matrix algorithmic exposition by Yakowitz and Rutherford (1984). For the entire demonstration the reader should reference to the source references.

The fundamental DDP algorithm

In this section we derive the fundamental DDP algorithm, for an unconstrained problem, starting from the general formulation presented in the previous section.

Both sides of Eq. (10) are expanded in Taylor series about the point $(\bar{\mathbf{s}}_k, \mathbf{u}_k^*)$:

$$\begin{aligned} \frac{1}{2} \delta \mathbf{s}_k^T V_{ss}^k \delta \mathbf{s}_k + V_s^k \delta \mathbf{s}_k + \bar{V}_k(\bar{\mathbf{s}}_k) + \Theta_k &\approx \min_{\delta \mathbf{u}_k} \left[\Theta_{k+1} + g(\bar{\mathbf{s}}_k, \bar{\mathbf{u}}_k; t_k) + \Delta \mathbf{g}_k + \mathbf{g}_s^k \delta \mathbf{s}_k + \right. \\ &+ \mathbf{g}_u^k \delta \mathbf{u}_k + \frac{1}{2} \delta \mathbf{s}_k^T \mathbf{g}_{ss}^k \delta \mathbf{s}_k + \frac{1}{2} \delta \mathbf{u}_k^T \mathbf{g}_{uu}^k \delta \mathbf{u}_k + \delta \mathbf{u}_k^T \mathbf{g}_{us}^k \delta \mathbf{s}_k + \bar{V}_{k+1}(\bar{\mathbf{s}}_{k+1}) + V_s^{k+1} \delta \mathbf{s}_{k+1} + \\ &\left. + \frac{1}{2} \delta \mathbf{s}_{k+1}^T V_{ss}^{k+1} \delta \mathbf{s}_{k+1} \right] \end{aligned} \quad (13)$$

where Θ_k is defined as the difference between the optimal return function obtained by applying $\{\mathbf{u}_j\}_{j=k}^N$ from the state $\bar{\mathbf{s}}_k$ until the end of the trajectory, and the nominal cost computed by using $\{\bar{\mathbf{u}}_j\}_{j=k}^N$ from the state $\bar{\mathbf{s}}_k$ until the end of the trajectory:

$$\Theta_k = V_k(\bar{\mathbf{s}}_k) - \bar{V}_k(\bar{\mathbf{s}}_k) \quad (14)$$

Analogously we define $\Theta_{k+1} = V_{k+1}(\bar{\mathbf{s}}_{k+1}) - \bar{V}_{k+1}(\bar{\mathbf{s}}_{k+1})$, while $\Delta \mathbf{g}_k = g(\bar{\mathbf{s}}_k, \mathbf{u}_k^*; t_k) - g(\bar{\mathbf{s}}_k, \bar{\mathbf{u}}_k; t_k)$. The left-hand side of Eq. (13) contains linear and quadratic terms of $\delta \mathbf{s}_k$ and the right-hand side contains linear and quadratic terms of $\delta \mathbf{s}_k$, $\delta \mathbf{u}_k$ and $\delta \mathbf{s}_{k+1}$, where:

$$\begin{aligned} \delta \mathbf{s}_{k+1} = \mathbf{s}_{k+1} - \bar{\mathbf{s}}_{k+1} &= \mathbf{f}(\bar{\mathbf{s}}_k + \delta \mathbf{s}_k, \mathbf{u}_k^* + \delta \mathbf{u}_k; t_k) - \mathbf{f}(\bar{\mathbf{s}}_k, \bar{\mathbf{u}}_k; t_k) = \\ \Delta \mathbf{f}_k + \mathbf{f}_s^k \delta \mathbf{s}_k + \mathbf{f}_u^k \delta \mathbf{u}_k &+ \frac{1}{2} \delta \mathbf{s}_k^T \mathbf{f}_{ss}^k \delta \mathbf{s}_k + \frac{1}{2} \delta \mathbf{u}_k^T \mathbf{f}_{uu}^k \delta \mathbf{u}_k + \delta \mathbf{u}_k^T \mathbf{f}_{us}^k \delta \mathbf{s}_k \end{aligned} \quad (15)$$

with $\Delta \mathbf{f}_k = \mathbf{f}(\bar{\mathbf{s}}_k, \mathbf{u}_k^*; t_k) - \mathbf{f}(\bar{\mathbf{s}}_k, \bar{\mathbf{u}}_k; t_k)$. By substituting Eq. (15) in Eq. (13) and by grouping the terms of the same order, the resulting equation can be written in a matrix form:

$$\begin{aligned} \delta \mathbf{s}_k^T \mathbf{P}_k \delta \mathbf{s}_k + \mathbf{Q}_k \delta \mathbf{s}_k + \Theta_k &\approx \min_{\delta \mathbf{u}_k} \left[\delta \mathbf{s}_k^T \mathbf{A}_k \delta \mathbf{s}_k + \delta \mathbf{u}_k^T \mathbf{C}_k \delta \mathbf{u}_k + \delta \mathbf{u}_k^T \mathbf{B}_k \delta \mathbf{s}_k + \delta \mathbf{s}_k^T \mathbf{E}_k + \right. \\ &\left. + \delta \mathbf{u}_k^T \mathbf{D}_k + \Theta_{k+1} + \Delta \mathbf{g}_k + \mathbf{Q}_{k+1} \Delta \mathbf{f}_k + \Delta \mathbf{f}_k^T \mathbf{P}_{k+1} \Delta \mathbf{f}_k \right] \end{aligned} \quad (16)$$

where some matrices are introduced for clarity purpose.

$$\begin{aligned}\mathbf{P}_k &= \frac{1}{2} \mathbf{V}_{ss}^k & n \times n \\ \mathbf{Q}_k &= \mathbf{V}_s^k & 1 \times n\end{aligned}$$

denote the linear and quadratic part of the Taylor expansion of the optimal return function at stage k . The matrices \mathbf{A}_k , \mathbf{B}_k , \mathbf{C}_k , \mathbf{D}_k and \mathbf{E}_k , instead, contains the derivatives of the stage-wise loss function g and the state transition function \mathbf{f} at stage k , and the derivatives of the optimal return function of the next stage forward V_{k+1} . If \mathbf{u}_k and \mathbf{s}_k are respectively a $m \times 1$ and $n \times 1$ vector, we define \mathbf{g}_u and \mathbf{g}_s to be respectively the $1 \times m$ and $1 \times n$ gradient of the scalar cost function g with respect to the components of the control and the state vector; \mathbf{g}_{uu} , \mathbf{g}_{ss} and \mathbf{g}_{su} represent the block components of the Hessian matrix of g respectively of size $m \times m$, $n \times n$ and $n \times m$. Said $\mathbf{f}(\mathbf{s}_k, \mathbf{u}_k; t_k)$ the state transition matrix, we denote with \mathbf{f}_u and \mathbf{f}_s the Jacobian of \mathbf{f} with respect to u and s of size $n \times m$ and $n \times n$ and with \mathbf{f}_{uu} , \mathbf{f}_{ss} and \mathbf{f}_{us} the blocks components of the Hessian matrix of \mathbf{f} respectively of size $m \times m \times n$, $n \times n \times n$ and $n \times m \times n$. All the above quantities are evaluated at $(\bar{\mathbf{s}}_k, \mathbf{u}_k^*)$.

$$\begin{aligned}\mathbf{A}_k &= \frac{1}{2} \left(\mathbf{g}_{ss}^k + \sum_{j=1}^n [\mathbf{V}_s^{k+1}]_j [\mathbf{f}_{ss}^k]_j + \mathbf{f}_s^{kT} \mathbf{V}_{ss}^{k+1} \mathbf{f}_s^k + \Delta \mathbf{f}_k \mathbf{V}_{ss}^{k+1} \mathbf{f}_{ss}^k \right) & n \times n \\ \mathbf{B}_k &= \left(\mathbf{g}_{su}^k + \sum_{j=1}^n [\mathbf{V}_s^{k+1}]_j [\mathbf{f}_{su}^k]_j + \mathbf{f}_s^{kT} \mathbf{V}_{ss}^{k+1} \mathbf{f}_u^k + \Delta \mathbf{f}_k \mathbf{V}_{ss}^{k+1} \mathbf{f}_{su}^k \right)^T & m \times n \\ \mathbf{C}_k &= \frac{1}{2} \left(\mathbf{g}_{uu}^k + \sum_{j=1}^n [\mathbf{V}_s^{k+1}]_j [\mathbf{f}_{uu}^k]_j + \mathbf{f}_u^{kT} \mathbf{V}_{ss}^{k+1} \mathbf{f}_u^k + \Delta \mathbf{f}_k \mathbf{V}_{ss}^{k+1} \mathbf{f}_{uu}^k \right) & m \times m \quad (17) \\ \mathbf{D}_k &= \left(\mathbf{g}_u^k + \mathbf{V}_s^{k+1} \mathbf{f}_u^k + \Delta \mathbf{f}_k^T \mathbf{V}_{ss}^{k+1} \mathbf{f}_u^k \right)^T & m \times 1 \\ \mathbf{E}_k &= \left(\mathbf{g}_s^k + \mathbf{V}_s^{k+1} \mathbf{f}_s^k + \Delta \mathbf{f}_k^T \mathbf{V}_{ss}^{k+1} \mathbf{f}_s^k \right)^T & n \times 1\end{aligned}$$

Note that the last terms of the matrices \mathbf{A}_k , \mathbf{B}_k and \mathbf{C}_k have to be rewritten in order to represent a quadratic form respectively with respect to $(\delta \mathbf{s}_k, \delta \mathbf{s}_k)$, $(\delta \mathbf{s}_k, \delta \mathbf{u}_k)$ and $(\delta \mathbf{u}_k, \delta \mathbf{u}_k)$. Moreover the matrices \mathbf{A}_k , \mathbf{C}_k are symmetric.

The constant part of Eq. (16), instead, can be grouped in:

$$\Theta_k = \Theta_{k+1} + \Delta \mathbf{g}_k + \mathbf{Q}_{k+1} \Delta \mathbf{f}_k + \Delta \mathbf{f}_k^T \mathbf{P}_{k+1} \Delta \mathbf{f}_k \quad (18)$$

with the final condition:

$$\Theta_{N+1} = 0 \quad (19)$$

The value of \mathbf{u}_k^* in Eq. (8) is computed by solving the minimisation problem on the right hand side of Eq. (9), which is equivalent to solving the right hand side of Eq. (16) for $\delta \mathbf{s}_k$ and $\delta \mathbf{u}_k$ set to zero:

$$\min_{\mathbf{u}_k^*} [\Delta \mathbf{g}_k + \mathbf{Q}_{k+1} \Delta \mathbf{f}_k + \Delta \mathbf{f}_k^T \mathbf{P}_{k+1} \Delta \mathbf{f}_k] \quad (20)$$

As a consequence at \mathbf{u}_k^* the following condition is satisfied:

$$\nabla_{\mathbf{u}} \left[\Delta \mathbf{g}_k + V_s^{k+1} \Delta \mathbf{f}_k + \frac{1}{2} \Delta \mathbf{f}_k^T V_{ss}^{k+1} \Delta \mathbf{f}_k \right] = 0 \Rightarrow \mathbf{g}_u^k + V_s^{k+1} \mathbf{f}_u^k + \Delta \mathbf{f}_k^T V_{ss}^{k+1} \mathbf{f}_u^k = 0 \Rightarrow \mathbf{D}_k = 0$$

Once \mathbf{u}_k^* is computed, problem Eq. (16) can be solved with respect to $\delta \mathbf{u}_k$. The necessary condition for the minimisation of Eq. (16) with respect to $\delta \mathbf{u}_k$ implies that:

$$2\mathbf{C}_k \delta \mathbf{u}_k + \mathbf{B}_k \delta \mathbf{s}_k = 0 \Rightarrow \delta \mathbf{u}_k = -\frac{1}{2} \mathbf{C}_k^{-1} \mathbf{B}_k \delta \mathbf{s}_k \quad (21)$$

Eq. (21) gives the coefficient $\boldsymbol{\beta}_k$ of the feedback control law in Eq. (11):

$$\boldsymbol{\beta}_k = -\frac{1}{2} \mathbf{C}_k^{-1} \mathbf{B}_k \quad m \times n \quad (22)$$

The variation in control in Eq. (21) can be substituted back in Eq. (16) and by grouping the terms of the same order we obtain:

$$\begin{aligned} \mathbf{Q}_k &= \mathbf{E}_k^T \\ \mathbf{P}_k &= \mathbf{A}_k - \frac{1}{4} \mathbf{B}_k^T \mathbf{C}_k^{-1} \mathbf{B}_k \end{aligned} \quad (23)$$

with the final conditions:

$$\begin{aligned} \mathbf{Q}_{N+1} &= \left[\boldsymbol{\lambda}^T \cdot \varphi(\bar{\mathbf{s}}_{N+1}; t_{N+1}) \right]_s \\ \mathbf{P}_{N+1} &= \frac{1}{2} \left[\boldsymbol{\lambda}^T \cdot \varphi(\bar{\mathbf{s}}_{N+1}; t_{N+1}) \right]_{ss} \end{aligned} \quad (24)$$

Eqs. (20), (17), (18), (22) and (23) are computed backward in time for every stage $k = N, \dots, 1$ with the final condition Eqs. (19) and (24) at stage $N+1$ and the coefficient $\{\boldsymbol{\beta}_k\}_{k=1}^N$ is stored in memory for the forward propagation.

DDP ensures an improvement at each iteration under the condition that the Hessian of the cost function, i.e., the matrix \mathbf{C}_k is positive definite. In case this is not verified, different procedures can be applied (see Mayne, 1966; Jacobson and Mayne, 1969; Yakowitz and Rutherford, 1984; Liao and Shoemaker, 1992). The one implemented in this work replaces the matrix \mathbf{C}_k , for the computation of Eq. (22), with the positive definite matrix

$$\tilde{\mathbf{C}}_k = \mathbf{C}_k + 2|\nu_{\min}|\mathbf{I}(m) \quad (25)$$

where ν_{\min} is the minimum eigenvalue of the matrix \mathbf{C}_k and $\mathbf{I}(m)$ the identity matrix of size m .

The condition on the matrix \mathbf{C}_k is even more stringent; in fact, in order to achieve a sufficient descend direction at each iteration, the matrix \mathbf{C}_k should also be far from non-positive definite condition (Gill et al., 1981); hence the active shift Eq. (25) is applied, also in case the minimum eigenvalue ν_{\min} , although positive, is smaller than a given small positive value (10^{-6} was usually adopted).

Once the backward propagation is terminated, the trajectory is swept forward in time, for every stage $k = 1, \dots, N$; the new control policy is given by Eq. (12) and the corresponding trajectory is computed by Eq. (1). The value of k_{lim} in Eq. (12) has to be chosen such that the following condition is satisfied, c being a constant between 0 and 1.

$$J(\{\mathbf{u}_k\}; \bar{\mathbf{s}}_1) - J(\{\bar{\mathbf{u}}_k\}; \bar{\mathbf{s}}_1) \leq c \cdot \Theta_{k_{\text{lim}}} \quad (26)$$

where $J(\{\mathbf{u}_k\}; \bar{\mathbf{s}}_1)$ is the value of the cost function associated to the new control law, computed with Eq. (4). Following to the definition in Eq. (14), $\Theta_{k_{\text{lim}}}$ is used as a measure of the predicted change in cost applying the control law Eq. (12).

A single iteration of DDP is composed by the backward and the forward recursion that produce an improved control law and trajectory. A number of iterations follow one after the other, until the stopping condition

$$\Theta_1 < \Theta_{\text{out}} \quad (27)$$

is verified, being Θ_{out} a fixed threshold.

Treatment of the terminal equality constraints

The terminal constraints are added to the cost function through a set of Lagrange multipliers λ to give the Lagrange function in Eq. (4).

In this paper we follow the method proposed by Gershwin and Jacobson (1970).

At first Eq. (4) is minimised fixing the value of the Lagrange multipliers $\bar{\lambda}$.

Successive iterations of DDP follow until the convergence criterion Eq. (27) is

satisfied. At this point a variation of $\delta\lambda$ is allowed, in order to find a control law

that decreases the constraints violation. Eq. (5) is now expanded not only in $\delta\mathbf{u}_k$

and $\delta\mathbf{s}_k$ but also in $\delta\lambda$ about the point $(\bar{\mathbf{s}}_k, \mathbf{u}_k^*, \bar{\lambda})$, where $\bar{\lambda}$ is considered to be

the nominal value of the Lagrange multipliers:

$$\begin{aligned}
& \frac{1}{2} \delta \mathbf{s}_k^T V_{ss}^k \delta \mathbf{s}_k + \frac{1}{2} \delta \lambda^T V_{\lambda\lambda}^k \delta \lambda + \delta \mathbf{s}_k^T V_{s\lambda}^k \delta \lambda + V_s^k \delta \mathbf{s}_k + V_\lambda^k \delta \lambda + \bar{V}_k(\bar{\mathbf{s}}_k) + \Theta_k = \\
& = \Theta_{k+1} + g(\bar{\mathbf{s}}_k, \bar{\mathbf{u}}_k; t_k) + \Delta \mathbf{g}_k + \mathbf{g}_s^k \delta \mathbf{s}_k + \mathbf{g}_u^k \delta \mathbf{u}_k + \frac{1}{2} \delta \mathbf{s}_k^T \mathbf{g}_{ss}^k \delta \mathbf{s}_k + \frac{1}{2} \delta \mathbf{u}_k^T \mathbf{g}_{uu}^k \delta \mathbf{u}_k + \\
& + \delta \mathbf{u}_k^T \mathbf{g}_{us}^k \delta \mathbf{s}_k + \bar{V}_{k+1}(\bar{\mathbf{s}}_{k+1}) + V_s^{k+1} \delta \mathbf{s}_{k+1} + \frac{1}{2} \delta \mathbf{s}_{k+1}^T V_{ss}^{k+1} \delta \mathbf{s}_{k+1} + V_\lambda^{k+1} \delta \lambda + \\
& + \frac{1}{2} \delta \lambda^T V_{\lambda\lambda}^{k+1} \delta \lambda + \delta \mathbf{s}_{k+1}^T V_{s\lambda}^{k+1} \delta \lambda
\end{aligned} \tag{28}$$

Substituting Eq. (15) and grouping some terms, Eq. (28) can be written in a matrix form:

$$\begin{aligned}
& \delta \mathbf{s}_k^T \mathbf{P}_k \delta \mathbf{s}_k + \delta \lambda^T \mathbf{R}_k \delta \lambda + \delta \mathbf{s}_k^T \mathbf{S}_k \delta \lambda + \mathbf{Q}_k \delta \mathbf{s}_k + \mathbf{Z}_k \delta \lambda + \Theta_k = \\
& = \delta \mathbf{s}_k^T \mathbf{A}_k \delta \mathbf{s}_k + \delta \mathbf{u}_k^T \mathbf{C}_k \delta \mathbf{u}_k + \delta \mathbf{u}_k^T \mathbf{B}_k \delta \mathbf{s}_k + \delta \mathbf{s}_k^T \mathbf{E}_k + \delta \mathbf{u}_k^T \mathbf{D}_k + \\
& + \delta \lambda^T \mathbf{R}_{k+1} \delta \lambda + \delta \mathbf{s}_k^T \mathbf{H}_k \delta \lambda + \delta \mathbf{u}_k^T \mathbf{K}_k \delta \lambda + \mathbf{Z}_{k+1} \delta \lambda + \Theta_{k+1}
\end{aligned} \tag{29}$$

where some more matrices are introduced for clarity; respectively on the left side:

$$\begin{aligned}
\mathbf{Z}_k &= V_\lambda^k & 1 \times l \\
\mathbf{R}_k &= \frac{1}{2} (V_{\lambda\lambda}^k) & l \times l \\
\mathbf{S}_k &= V_{s\lambda}^k & n \times l
\end{aligned} \tag{30}$$

and on the right side.

$$\begin{aligned}
\mathbf{R}_{k+1} &= \frac{1}{2} (V_{\lambda\lambda}^{k+1}) && l \times l \\
\mathbf{H}_k &= (\mathbf{f}_s^{kT} V_{s\lambda}^{k+1}) = \mathbf{f}_s^{kT} \mathbf{S}_{k+1} && n \times l \\
\mathbf{K}_k &= (\mathbf{f}_u^{kT} V_{s\lambda}^{k+1}) = \mathbf{f}_u^{kT} \mathbf{S}_{k+1} && m \times l \\
\mathbf{Z}_{k+1} &= V_{\lambda}^{k+1} && 1 \times l
\end{aligned} \tag{31}$$

Note that the variation of Lagrange multipliers is introduced only once an optimal control law has been found with $\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}}$; as a consequence, from Eq. (28), $\mathbf{u}_k^* \simeq \bar{\mathbf{u}}_k$ and hence $\Delta \mathbf{g}_k = 0$ and $\Delta \mathbf{f}_k = 0$. This is equivalent to use the small control variation algorithm (Jacobson and Mayne, 1979). Now, by differentiating Eq. (29) with respect to $\delta \mathbf{u}_k$ we obtain:

$$\begin{aligned}
2\mathbf{C}_k \delta \mathbf{u}_k + \mathbf{B}_k \delta \mathbf{s}_k + \mathbf{K}_k \delta \boldsymbol{\lambda} &= 0 \Rightarrow \\
\delta \mathbf{u}_k &= -\frac{1}{2} \mathbf{C}_k^{-1} \mathbf{B}_k \delta \mathbf{s}_k - \frac{1}{2} \mathbf{C}_k^{-1} \mathbf{K}_k \delta \boldsymbol{\lambda}
\end{aligned}$$

Hence the variation of the control contains also a term proportional to the variation of the multipliers:

$$\delta \mathbf{u}_k = \boldsymbol{\beta}_k \delta \mathbf{s}_k + \boldsymbol{\gamma}_k \delta \boldsymbol{\lambda} \tag{32}$$

The associated coefficient $\boldsymbol{\gamma}_k$ is computed during the backward recursion and stored in memory together with coefficient $\boldsymbol{\beta}_k$:

$$\boldsymbol{\gamma}_k = -\frac{1}{2} \mathbf{C}_k^{-1} \mathbf{K}_k \quad m \times l \tag{33}$$

By substituting back Eq. (32) in Eq. (29) we obtain:

$$\begin{aligned}
\mathbf{S}_k &= \mathbf{H}_k - \frac{1}{2} \mathbf{B}_k^T \mathbf{C}_k^{-1} \mathbf{K}_k \\
\mathbf{R}_k &= \mathbf{R}_{k+1} - \frac{1}{4} \mathbf{K}_k^T \mathbf{C}_k^{-1} \mathbf{K}_k \\
\mathbf{Z}_k &= \mathbf{Z}_{k+1}
\end{aligned} \tag{34}$$

with the final conditions:

$$\begin{aligned}
\mathbf{S}_{N+1} &= \varphi_s (\bar{\mathbf{s}}_{N+1}; t_{N+1})^T \\
\mathbf{R}_{N+1} &= \mathbf{0} \\
\mathbf{Z}_{N+1} &= \varphi (\bar{\mathbf{s}}_{N+1}; t_{N+1})
\end{aligned} \tag{35}$$

The backward recursion is performed for every stage $k = N, \dots, 1$, in which the same equations of the main DDP loop are solved, with the addition of Eqs. (31), (33) and (34), with the final condition Eqs. (35); the coefficients $\{\boldsymbol{\beta}_k\}_{k=1}^N$ and $\{\boldsymbol{\gamma}_k\}_{k=1}^N$ are stored in memory.

At this point we can determine the variation of Lagrange multipliers $\delta\boldsymbol{\lambda}$, by maximising Eq. (28) at t_1 and $\bar{\mathbf{s}}_1$, with respect to $\delta\boldsymbol{\lambda}$ (see Jacobson and Mayne, 1969); this gives:

$$\delta\boldsymbol{\lambda} = -\frac{1}{2}\mathbf{R}_1^T\mathbf{Z}_1 \quad (36)$$

under the requirement that \mathbf{R}_1 is negative definite (hence invertible).

The new control law and trajectory are propagated for every stage $k = 1, \dots, N$:

$$\begin{cases} \mathbf{u}_k = \bar{\mathbf{u}}_k + \boldsymbol{\beta}_k\delta\mathbf{s}_k + \boldsymbol{\gamma}_k\delta\boldsymbol{\lambda} \\ \mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{u}_k; t_k) \\ \mathbf{s}_1 = \bar{\mathbf{s}}_1 \end{cases} \quad k = 1, \dots, N \quad (37)$$

Also in this case, $\delta\boldsymbol{\lambda}$ has to be verified not to exceed the range of validity of the linear-quadratic expansion, hence the constant $0 < \varepsilon \leq 1$ is introduced in Eq. (36):

$$\begin{aligned} \delta\boldsymbol{\lambda} &= -\varepsilon\frac{1}{2}\mathbf{R}_1^T\mathbf{Z}_1 & 0 < \varepsilon \leq 1 \\ \boldsymbol{\lambda} &= \bar{\boldsymbol{\lambda}} + \delta\boldsymbol{\lambda} \end{aligned} \quad (38)$$

The value of ε is chosen, through a linear search method, so that the following condition is satisfied (Gershwin and Jacobson, 1970):

$$\begin{aligned} & \left| J(\{\mathbf{u}_k\}, \boldsymbol{\lambda}(\varepsilon); \bar{\mathbf{s}}_1) - J(\{\bar{\mathbf{u}}_k\}, \bar{\boldsymbol{\lambda}}; \bar{\mathbf{s}}_1) \right| - \Theta_1 + \\ & + \frac{1}{2}\left(\varepsilon - \frac{1}{2}\varepsilon^2\right) \cdot \boldsymbol{\varphi}(\bar{\mathbf{s}}_{N+1}; t_{N+1})^T \mathbf{R}_1^{-1} \boldsymbol{\varphi}(\bar{\mathbf{s}}_{N+1}; t_{N+1}) < \text{reltol} \left| J(\{\bar{\mathbf{u}}_k\}; \bar{\mathbf{s}}_1) \right| \end{aligned} \quad (39)$$

where *reltol* is a relative tolerance. Eq. (39) compares the actual improvement in the cost function to the one predicted through the linear-quadratic expansion. Moreover the change in $\boldsymbol{\lambda}$ has to reduce the violation of the terminal constraints:

$$\boldsymbol{\varphi}(\mathbf{s}_{N+1}; t_{N+1}) - \boldsymbol{\varphi}(\bar{\mathbf{s}}_{N+1}; t_{N+1}) < 0 \quad (40)$$

3 Modified DDP method

When the optimisation problem is not very sensitive, for example when designing a two-body problem transfer, the conventional DDP technique, described in paragraph 2, can be applied to find the optimal control. However if the problem involves a more complex dynamics, such as escape or capture phases, or gravity assist passages, the propagation of the dynamics becomes a crucial point. In particular, the use of a time mesh fixed a priori can jeopardise the high fidelity representation of the problem; on the other hand, the coupling between the integration scheme and the optimisation process must be handled very carefully, in order not to compromise convergence.

The approach proposed in this paper uses a variable step integration method for the propagation of the dynamics and the integration mesh is refined at each iteration of DDP.

3.1 Discretisation scheme

The low-thrust continuous problem, characterised by the dynamic system:

$$\begin{aligned} \dot{\mathbf{s}}(t) &= \tilde{\mathbf{f}}(\mathbf{s}(t), \mathbf{u}(t); t) & t \in [t_0 \quad t_f] \\ \mathbf{s}(t_0) &= \bar{\mathbf{s}}_1 \end{aligned} \quad (41)$$

is approximated by difference equations as shown in Eq. (1), where the state transition function \mathbf{f} represents the explicit scheme for the numerical approximation of Eq. (41):

$$\begin{aligned} \mathbf{s}_{k+1} &= \mathbf{f}(\mathbf{s}_k, \tilde{\mathbf{f}}(\mathbf{s}_k, \mathbf{u}_k; t_k); h_k) & k = 1, \dots, N \\ \mathbf{s}_1 &= \bar{\mathbf{s}}_1 \end{aligned} \quad (42)$$

where h_k is the discretisation step. Note that in the rest of the paper the dependences of the function \mathbf{f} were written in the simplified form:

$$\mathbf{f}(\mathbf{s}_k, \mathbf{u}_k; t_k) = \mathbf{f}(\mathbf{s}_k, \tilde{\mathbf{f}}(\mathbf{s}_k, \mathbf{u}_k; t_k); h_k)$$

In this paper we use the discrete-time form of DDP; according to this approach, the N steps identify both the decision times of the trajectory (i.e., the points where the feedback control is computed) and the steps of the numerical propagation, as shown in Fig. 5.

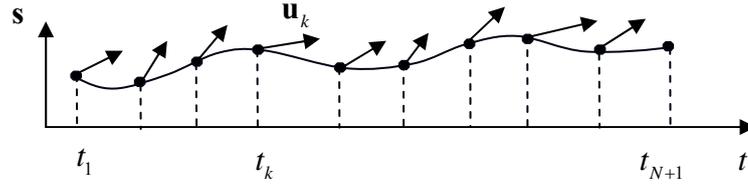


Fig. 5 Trajectory discretisation within the optimisation problem.

In a previous application of the discrete-time DDP algorithm to orbital transfer, a fixed step size Euler integration scheme was used (see Gershwin and Jacobson, 1970). However, such a simple integration scheme is not appropriate when the dynamics becomes highly non-linear. In other more recent DDP-based approaches, the issue was solved by dividing the trajectory in a number of segments over which the thrust is constant (Whiffen, 2002; Lantoine and Russell, 2008). Within a single segment Whiffen integrates backward a system of coupled ordinary differential equations which are the integral form of the discrete-time DDP matrices, while Lantoine and Russel introduce a second order state transition matrix to map the propagation of the dynamics. In these approaches, decision times and integration steps do not coincide.

In our work, the classical discrete formulation is used (see Fig. 5) but the mesh is discretised with a scheme more accurate than the one adopted by Gershwin and Jacobson (1970): a variable step-size Runge-Kutta-Fehlberg integration scheme, with a six stage pair of approximation of the forth and fifth order (Dormand and Prince, 1980):

$$\begin{aligned}
 \mathbf{s}_{k+1} &= \mathbf{f}(\mathbf{s}_k, \mathbf{u}_k; t_k) = \mathbf{s}_k + h_k \sum_{r=1}^{\sigma} \mathbf{b}_r \tilde{\mathbf{f}}_r \\
 \tilde{\mathbf{f}}_r &= \tilde{\mathbf{f}} \left(\mathbf{s}_k + h_k \sum_{r=1}^{\sigma} \mathbf{a}_{j,r} \tilde{\mathbf{f}}_r, \mathbf{u}_k; t_k + \mathbf{c}_r h_k \right) \quad j=1, \dots, \sigma; \quad \sigma=6
 \end{aligned} \tag{43}$$

where $\tilde{\mathbf{f}}$ is the continuous dynamic of the problem, \mathbf{a} , \mathbf{b} and \mathbf{c} the coefficient matrices of the integration scheme and h_k the length of the discretisation step.

Note that the integration scheme Eq. (43) was chosen to be explicit, as it allows the analytic evaluation of its derivatives which are required in the DDP procedure (in Eqs. (17) and Eqs. (31)).

The identity between decision times and integration steps increases the computational requirements but ensures high fidelity of the dynamics and allows varying the control at each integration step.

Note that, if \mathbf{u}_k is kept constant over a certain number of discretisation steps, Eq. (43) reduces to the trajectory model used by Whiffen and Lantoiné (see Fig. 6).

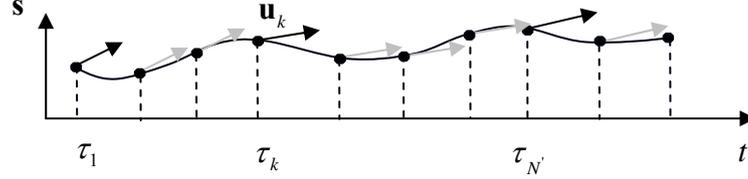


Fig. 6 Trajectory discretisation in the Static/Dynamic Control approach. The gray arrows show that the control is kept constant within a segment.

3.2 Mesh definition

If the dynamic system Eq. (42) is not correctly integrated, the optimisation of the control law could lead to an incorrect solution. This is likely to occur if a fixed step size is used.

For this reason, in the approach proposed in this paper, the step-size of the integration scheme was adapted at each iteration of the DDP algorithm.

We define a nominal time-mesh grid $\{\bar{N}\}$ together with the first guess trajectory and control sequences $\{\bar{\mathbf{s}}_k\}_{k=1}^{\bar{N}+1}$ and $\{\bar{\mathbf{u}}_k\}_{k=1}^{\bar{N}}$. The backward and forward propagation of DDP are then executed with the nominal mesh.

Once a value of k_{lim} is determined, according to condition Eq. (26), the trajectory selected for the next DDP iteration follows, within the range $[0 \ k_{\text{lim}}]$, the nominal path, while the segment of the trajectory $\{\mathbf{s}_k\}_{k=k_{\text{lim}}}^{\bar{N}+1}$ implements the new control strategy $\{\mathbf{u}_k\}_{k=k_{\text{lim}}}^{\bar{N}}$, according to Eq. (12).

The segment $\{\mathbf{s}_k\}_{k=k_{\text{lim}}}^{\bar{N}+1}$ of the trajectory is propagated through the adaptive-step integration algorithm and a new mesh $\{N\}$ is defined for $\{\mathbf{s}_k\}_{k=k_{\text{lim}}}^{N+1}$ and $\{\mathbf{u}_k\}_{k=k_{\text{lim}}}^N$.

In order to perform this operation, the control law, which is given on the original mesh points, needs to be interpolated in the new points required by the integration algorithm. Handling properly the interpolation is essential to preserve the DDP

performances; in fact a bad interpolation could introduce errors which can result in rejecting the control computed by the DDP.

Two interpolation schemes were adopted in this study; the first one, called *complete interpolation* in the following, directly interpolates the control $\{\mathbf{u}_k\}_{k=k_{\text{im}}}^{\bar{N}}$ on the new mesh. Recalling Eq. (12), the complete interpolation technique interpolates the left hand side of Eq. (44):

$$\mathbf{u}_k = \mathbf{u}_k^* + \boldsymbol{\beta}_k (\mathbf{s}_k - \bar{\mathbf{s}}_k) \quad k = k_{\text{im}}, \dots, N \quad (44)$$

The second interpolation technique, although more computationally expensive, ensures a higher accuracy. Rather than interpolating the control computed on the nominal mesh through the forward recursion $\{\mathbf{u}_k\}_{k=k_{\text{im}}}^{\bar{N}}$, each term on the right hand side of Eq. (44), namely \mathbf{u}_k^* , $\boldsymbol{\beta}_k$, $\bar{\mathbf{s}}_k$, is independently interpolated. In this way the feedback nature of the control variation computed by the DDP is fully exploited: if the state $\{\mathbf{s}_k\}_{k=k_{\text{im}}}^{N+1}$ moves away from the one computed on the nominal mesh $\{\mathbf{s}_k\}_{k=k_{\text{im}}}^{\bar{N}+1}$, the term of the control $\boldsymbol{\beta}_k (\mathbf{s}_k - \bar{\mathbf{s}}_k)$ changes as a consequence. The piecewise cubic spline interpolation method (De Boor, 1978) is adopted. We will call this technique as *term-wise interpolation*.

In some cases (see the transfer problem presented in section 5.1) the complete interpolation technique is enough to reach convergence, while in more sensitive and complex cases, the complete interpolation introduces small errors in the interpolated control that, propagated through a sensitive dynamics, may result in an unrecoverable increase of the final constraints violation. For example for the transfer problem presented in section 5.2, the term-wise interpolation technique was essential to reach the convergence. In particular, the section of the trajectory where the spacecraft passes close to the Earth gravity field, highlighted in Fig. 18, showed to be very sensitive to the control profile and hence required a very high accuracy in the control law interpolation.

Once the new mesh is defined, an additional test is performed, to assess whether the refinement of the mesh did not introduce errors in the dynamics. The cost computed with the new mesh $J(\{\mathbf{u}_k\}_{k=1}^N; \bar{\mathbf{s}}_1)$ must not differ from the cost computed with the nominal mesh $J(\{\mathbf{u}_k\}_{k=1}^{\bar{N}}; \bar{\mathbf{s}}_1)$ by a predefined quantity:

$$J\left(\{\mathbf{u}_k\}_{k=1}^N; \bar{\mathbf{s}}_1\right) - J\left(\{\mathbf{u}_k\}_{k=1}^{\bar{N}}; \bar{\mathbf{s}}_1\right) < reltol_{mesh} \cdot J\left(\{\mathbf{u}_k\}_{k=1}^{\bar{N}}; \bar{\mathbf{s}}_1\right) \quad reltol_{mesh} \ll 1 \quad (45)$$

By using the term-wise interpolation technique, condition Eq. (45) was always satisfied. In the cases in which the complete interpolation technique was adopted, instead, Eq. (45) was used as verification of the failure of the interpolation technique; when that occurred, the DDP mesh refinement was performed again, with the term-wise interpolation technique.

Note that the mesh refinement during the optimisation process increases the computational time, but only in this way one can ensure that the algorithm converges to a correct solution.

4 Algorithm

We now report a summary of the algorithm adopted in this work. The algorithm is composed by the following steps:

Initialisation

A nominal set of Lagrange multipliers $\bar{\lambda}$ and a control law $\{\bar{\mathbf{u}}_k\}_{k=1}^{\bar{N}}$ is given as an input to the algorithm; the associated nominal trajectory $\{\bar{\mathbf{s}}_k\}_{k=1}^{\bar{N}+1}$ is propagated through Eq. (1), where $\bar{\mathbf{s}}_1$ has also been fixed. The first guess trajectory also determines the nominal mesh of the problem $\{\bar{N}\}$. The cost function $J(\{\bar{\mathbf{u}}_k\}; \bar{\mathbf{s}}_1)$ associated to the nominal strategy and trajectory is evaluated through Eq. (4). Moreover the derivatives of the state transition function and the stage-wise loss function g are analytically computed.

Loop1: Control law loop

Step 1: The parameters needed for starting the recursive computation of Eq. (16) are initialised at step $N+1$, through Eq. (19) and Eqs. (24), computed with the nominal value of the Lagrange multipliers $\bar{\lambda}$.

Step 2: Backward propagation performed for each stage k from stage N to backward until stage 1:

The nominal control $\bar{\mathbf{u}}_k$ is varied to the new nominal policy \mathbf{u}_k^* , by minimising Eq. (20). The local minimisation of Eq. (20) is performed numerically, through a

subspace trust-region method, based on the interior-reflective Newton method (Coleman and Li, 1996; Coleman and Li, 1994). The analytic expression of the gradient is supplied.

The derivative of the state transition function and the stage-wise loss function g are evaluated at $(\bar{\mathbf{s}}_k, \mathbf{u}_k^*)$ and the matrices in Eq. (17) are constructed.

Eq. (18) represents the prevision of the improvement in the cost function associated with stage k , while the matrices \mathbf{Q}_k and \mathbf{P}_k are computed through Eq. (23) and replaced to the one of the next step forward. The coefficient $\boldsymbol{\beta}_k$ is computed with Eq. (22) and stored in memory for the forward propagation. If the matrix \mathbf{C}_k is not positive definite, Eq. (25) is used for the computation of $\boldsymbol{\beta}_k$:

$$\boldsymbol{\beta}_k = -\frac{1}{2}\tilde{\mathbf{C}}_k^{-1}\mathbf{B}_k \quad m \times n$$

Step 3: Forward propagation performed from step 1 to step N .

The trajectory is propagated through Eq. (1), with the improved control law Eq. (12). The value of k_{lim} is determined through Eq. (26), to provide a decrease in the objective function, and not to exceed the range of accuracy of the linear-quadratic expansion. The constant c in Eq. (26) was set in a value between 0.5 and 0.1.

Step 4: When a new control sequence $\{\mathbf{u}_k\}_{k_{\text{lim}}}^{\bar{N}}$ is selected, the corresponding leg of trajectory is integrated with the adaptive step integration method, by interpolating the control through the complete interpolation technique or the term-wise interpolation technique, and a new discretisation of the control is obtained $\{\mathbf{u}_k\}_{k_{\text{lim}}}^N$. The value of the cost function associated to the new discretisation and the value computed on the original mesh are compared through Eq. (45).

The new mesh, together with the improved control law and the associated trajectory are set as the nominal conditions for the next DDP iteration (*Step 1*).

$$\begin{aligned} \{\bar{N}\} &\leftarrow \{N\} \\ \{\bar{\mathbf{u}}_k\}_{k=1}^{\bar{N}} &\leftarrow \{\mathbf{u}_k\}_{k=1}^N \\ \{\bar{\mathbf{s}}_k\}_{k=1}^{\bar{N}+1} &\leftarrow \{\mathbf{s}_k\}_{k=1}^{N+1} \\ J\left(\{\bar{\mathbf{u}}_k\}_{k=1}^{\bar{N}}; \bar{\mathbf{s}}_1\right) &\leftarrow J\left(\{\mathbf{u}_k\}_{k=1}^N; \bar{\mathbf{s}}_1\right) \end{aligned}$$

Convergence Criterion:

The first loop of DDP is stopped when $k_{\text{lim}} = 1$ and the increase of the cost function is under a small value, set for stability analysis:

$$\Theta_1 \leq \Theta_{\text{out}} \cdot \max \left[1, \left| J \left(\left\{ \mathbf{u}_k \right\}_{k=1}^N ; \bar{\mathbf{s}}_1 \right) \right| \right] \quad (46)$$

Usually Θ_{out} is set to be around 10^{-6} but it can be increased up to 10^{-4} if the problem is very sensitive in order to filter the numerical error introduced by the integration over a long time of flight.

Loop 2: Equality constraints loop

Step 5: Backward propagation performed for each stage k from stage N to backward until stage 1:

The matrices in Eq. (17) and Eq. (23) are constructed, together with the new matrices in Eq. (31) and Eq. (34) with the initial condition Eqs. (24) and Eq. (35). All the derivatives are now evaluated at point $(\bar{\mathbf{s}}_k, \bar{\mathbf{u}}_k, \bar{\boldsymbol{\lambda}})$.

Coefficients $\boldsymbol{\beta}_k$ and $\boldsymbol{\gamma}_k$ are computed through Eq. (22) and Eq. (33) and stored in memory for the forward propagation.

Step 6: Forward propagation performed from step 1 to step N .

The value of the Lagrange multiplier vector is updated with Eq. (38) and the new control law is propagated with Eq. (37). The value of ε is set according to condition Eq. (39).

Test on the final constraints:

The violation of the constraints is updated and, if condition Eq. (40) is verified, the new value of $\boldsymbol{\lambda}$ is set as the nominal one, together with the control sequence and trajectory; else ε is further decreased. The algorithm goes back to *Step 1* for further DDP iteration.

Stopping condition

The overall algorithm terminates at the end of loop 1, if condition Eq. (46) is satisfied and the constraints violation is under a required tolerance.

The overall algorithm is sketched in Fig. 7:

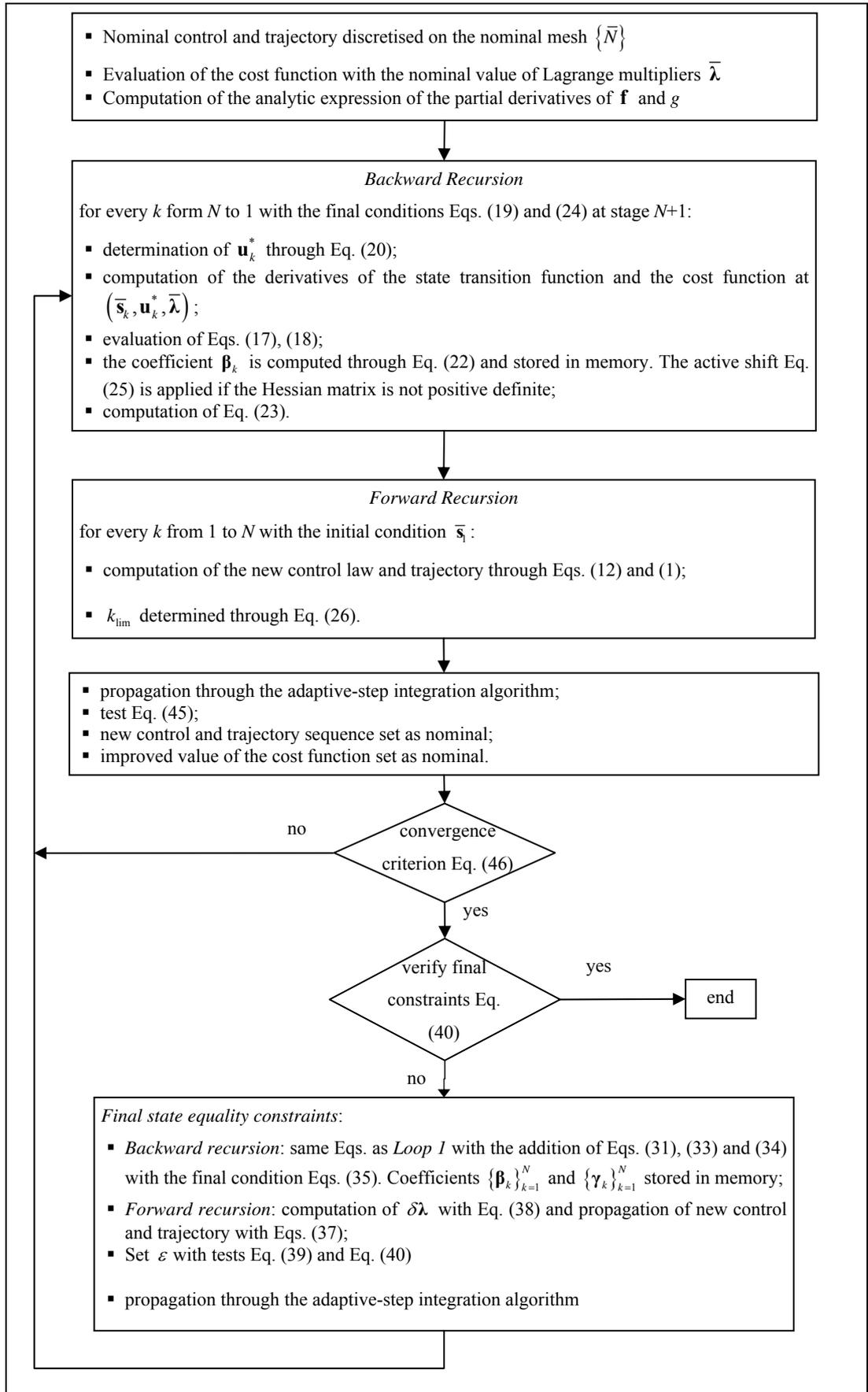


Fig. 7 Modified DDP algorithm.

4.1 Heuristics to improve the convergence rate

As mentioned above, DDP has quadratic convergence if the Hessian matrix of the problem is positive definite, i.e., the problem is locally convex. Conversely for non-convex control problems, the convergence rate downgrades to linear (Yakowitz and Rutherford, 1984); this is a common difficulty in direct methods as well.

Within the DDP procedure, this can be diagnosed in different ways; the matrix \mathbf{C}_k is not positive definite and the search of the local minimum \mathbf{u}_k^* of Eq. (20) may fail in *Step 2* of the backward propagation.

Another common issue, linked to the inaccuracy caused by the numerical approximation of the derivatives through finite-difference (Gill et al., 1981) is here avoided, because the derivatives of the cost function and the state transition function are analytically computed; this assures higher accuracy and allows saving computational time.

Moreover, the problem variables were scaled to have the same weight in the neighbourhood of the problem solution, thus preventing ill-conditioning of the Hessian matrix (Gill et al., 1981).

The cost function chosen for the constrained optimisation problem is the Lagrange function in Eq. (4). Eq. (4) is used both as cost function to be minimised and as a merit function to measure a progress of each iteration of DDP (Betts, 2000). Both the expressions of the integral term $\sum_{k=1}^N g(\mathbf{s}_k, \mathbf{u}_k; t_k)$ and the equality constraints $\varphi(\mathbf{s}_{N+1}; t_{N+1})$ were chosen to be quadratic forms. For this reason, numerical instability may occur if, in the equality constraints loop, any component of the Lagrange multiplier λ becomes negative.

Different heuristics were introduced in order to improve the convergence rate or to speed up the optimisation process.

When the search of the local minimum \mathbf{u}_k^* of Eq. (20) fails, in *Step 2* of the backward propagation, the nominal control $\bar{\mathbf{u}}_k$ is used in place of \mathbf{u}_k^* , in the following of the k -iteration. As a consequence the new control is restricted to be in the neighbourhood of the nominal strategy, according to:

$$\mathbf{u}_k = \bar{\mathbf{u}}_k + \boldsymbol{\beta}_k \delta \mathbf{s}_k$$

Anyway, only a limited number of iterations are allowed to fail in the search of the control \mathbf{u}_k^* . After a fixed number of iterations fail, the backward propagation is broken, and the value of k_{lim} for initialising the step-size adjustment method is set equal to the last value of the index k . In this case, the algorithm with global control variations showed to be very efficient, because, if at a given iteration k of the backward propagation the problem is locally non-convex, we do not need anyway to terminate the backward propagation up to step 1.

Another heuristic was adopted when, in the equality constraints loop, any component of the Lagrange multiplier $\boldsymbol{\lambda}$ becomes negative. The negative component itself is set to zero and the value of the integral term of the objective

function $\sum_{k=1}^N g(\mathbf{s}_k, \mathbf{u}_k; t_k)$ is multiplied by a weight parameter \tilde{w} :

$$\tilde{w} = \frac{\boldsymbol{\lambda}^T \cdot \varphi(\mathbf{s}_{N+1}; t_{N+1})}{\tilde{\boldsymbol{\lambda}}^T \cdot \varphi(\mathbf{s}_{N+1}; t_{N+1})}$$

where $\boldsymbol{\lambda}$ indicates the original Lagrange vector, and $\tilde{\boldsymbol{\lambda}}$ is the modified one, where the negative component is set to 0.

The end-point constraints loop may terminate without a decrease of the constraints violation, if condition Eq. (39) and Eq. (40) are never satisfied for any value of ε . If this occurs, rather than stopping the process, a trial value of $\boldsymbol{\lambda}$ is set and the algorithm continues with *Loop 1*.

$$\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}} + \frac{10 \cdot (\tilde{\boldsymbol{\lambda}}^T \cdot \varphi(\mathbf{s}_{N+1}; t_{N+1}))}{\sum_{j=1}^l \varphi_j^2} + \varphi(\mathbf{s}_{N+1}; t_{N+1})$$

Finally an important consideration must be done on the convergence rate of the process. The algorithm with global control variations usually converges faster than the traditional small control variations algorithm (Yakowitz and Rutherford, 1984), especially when far from the optimal solution.

This was verified on the design of the trajectories presented in section 5. With the small control variations algorithm, it was necessary to resort to a continuation technique on the specific impulse, while it was possible to find directly the final solution with the global control variations algorithm. However, the convergence

of the global control variations algorithm gets slower, as the value of Θ_1 becomes small. This was handled by switching to the small control variations algorithm, when close to the convergence within the first DDP loop, and switching back to the global control variation algorithm, once the value of λ is modified by the end-point constrains loop.

5 Asteroid rendezvous and fly-by missions

The algorithm presented in the previous section was applied to the optimisation of low-thrust trajectories to fly-by and rendezvous of near-Earth objects. The whole trajectory was described in an Earth inertial reference frame, centred in the Earth with the x-axis in the γ -point direction and the z-axis normal to the ecliptic plane, in the direction of the orbit angular momentum of the Earth (the y-axis completes the reference frame). The variables of the problem are the state vector $\mathbf{s}(t) = \{\mathbf{r} \quad \mathbf{v} \quad m\}^T$, made of position, velocity and mass of the spacecraft and the control vector, made of the three components of thrust along the coordinate directions $\mathbf{u}(t) = \mathbf{T}$. The dynamic equations governing the motion of the spacecraft are:

$$\begin{cases} \frac{d\mathbf{r}}{dt} = \mathbf{v} \\ \frac{d\mathbf{v}}{dt} = -\frac{\mu_{\text{Earth}}}{\|\mathbf{r}\|^3} \mathbf{r} - \mu_{\text{Sun}} \left(\frac{\mathbf{r}_{\text{Sun-s/c}}}{\|\mathbf{r}_{\text{Sun-s/c}}\|^3} - \frac{\mathbf{r}_{\text{Sun-Earth}}}{\|\mathbf{r}_{\text{Sun-Earth}}\|^3} \right) + \frac{\mathbf{T}}{m} \end{cases} \quad (47)$$

where μ_{Earth} and μ_{Sun} are respectively the Earth and Sun gravitational constant. \mathbf{r} is the position vector with respect to the Earth inertial reference frame, $\mathbf{r}_{\text{Sun-Earth}}$ is the position vector of the Earth in a Sun centred inertial reference frame and $\mathbf{r}_{\text{Sun-s/c}}$ is:

$$\mathbf{r}_{\text{Sun-s/c}} = \mathbf{r}_{\text{Sun-Earth}} + \mathbf{r}$$

The state vector of the Earth $\mathbf{s}_{\text{Sun-Earth}}$ was taken from analytic ephemerides which approximate JPL ephemerides de405[‡].

[‡] <http://naif.jpl.nasa.gov/naif/pds.html>

The terminal conditions at the asteroid, either rendezvous or fly-by, are included in the cost function through a quadratic term. In the case of a rendezvous mission the terminal constraints are:

$$\varphi(\mathbf{s}_{N+1}; t_{N+1}) = \left\{ \begin{array}{l} (x_{N+1} - x_{\text{target}})^2 \\ (y_{N+1} - y_{\text{target}})^2 \\ (z_{N+1} - z_{\text{target}})^2 \\ (v_{x,N+1} - v_{x,\text{target}})^2 \\ (v_{y,N+1} - v_{y,\text{target}})^2 \\ (v_{z,N+1} - v_{z,\text{target}})^2 \end{array} \right\}$$

while in the case of fly-by are:

$$\varphi(\mathbf{s}_{N+1}; t_{N+1}) = \left\{ \begin{array}{l} (x_{N+1} - x_{\text{target}})^2 \\ (y_{N+1} - y_{\text{target}})^2 \\ (z_{N+1} - z_{\text{target}})^2 \end{array} \right\}$$

where $\mathbf{s}_{\text{target}} = \{x_{\text{target}} \quad y_{\text{target}} \quad z_{\text{target}} \quad v_{x,\text{target}} \quad v_{y,\text{target}} \quad v_{z,\text{target}}\}^T$ is the state vector representing the position of the asteroid at the final time of the trajectory. For measuring the constraints satisfaction, the infinity norm of the error in position $\|\mathbf{r} - \mathbf{r}_{\text{target}}\|_{\infty}$ and velocity $\|\mathbf{v} - \mathbf{v}_{\text{target}}\|_{\infty}$ was required to be under a given tolerance; specifically an absolute tolerance of 10000 km was set for the positions and 0.01 km/s for the velocities.

The integral term of the cost function instead was selected to be the integral of the square of the norm of the thrust vector:

$$g(\mathbf{s}_k, \mathbf{u}_k; t_k) = \frac{1}{2} w \mathbf{T}_k^T \mathbf{T}_k \cdot h_k \quad (48)$$

being w a weight factor and $h_k = t_{k+1} - t_k$ the integration interval at step k . In summary the cost function of the problem is:

$$J(\{\mathbf{T}_k\}; \bar{\mathbf{s}}_1) = \sum_{k=1}^N \frac{1}{2} w \mathbf{T}_k^T \mathbf{T}_k \cdot h_k + \lambda^T \cdot (\mathbf{s}_{N+1} - \mathbf{s}_{\text{target}})^2$$

Generation of a first guess solution

A first guess was generated by patching together two low-thrust arcs with fixed thrust aligned to the velocity vector: one inside the sphere of influence of the Earth (i.e., until the distance from the Earth reaches the radius of the Earth sphere of influence) and one from the Earth to the asteroid. The first guess solution was a function of a reduced set of parameters: the departure time from the Earth and the angular position on a fixed parking orbit, the time of flight, and the thrust magnitude out of the Earth sphere of influence. The thrust magnitude of the spiralling-out phase was set outside the optimisation. Thus, a global search for optimal first guesses was performed using Differential Evolution (Price et al., 2005). The objective for the global search is to minimise the error in distance (for a fly-by mission) or in distance and velocity (for rendezvous mission) between the state of the spacecraft and the target position at the final time:

$$\begin{aligned} \text{rendezvous:} \quad J_{\text{global search}} &= \sum_{i=1}^3 \frac{|\mathbf{r}_{N+1,i} - \mathbf{r}_{\text{target},i}| - \text{tol}_r}{\text{tol}_r} \\ \text{fly-by:} \quad J_{\text{global search}} &= \sum_{i=1}^3 \frac{|\mathbf{r}_{N+1,i} - \mathbf{r}_{\text{target},i}| - \text{tol}_r}{\text{tol}_r} + \sum_{i=1}^3 \frac{|\mathbf{v}_{N+1,i} - \mathbf{v}_{\text{target},i}| - \text{tol}_v}{\text{tol}_v} \end{aligned}$$

where $\text{tol}_r = 10000$ km and $\text{tol}_v = 0.01$ km/s.

In the following three trajectories will be presented, the first two are rendezvous transfers to asteroid Apophis, the third one is a fly-by of asteroid 2002 AA29. In each of the three cases, the first guess is of course non optimal from an optimal control point of view and does not satisfy the terminal constraints; therefore the DDP algorithm is used to compute a locally optimal and feasible trajectory. The two transfers to Apophis presented are quite different for initial orbit around the Earth, thrust magnitude, time of flight and mass of the spacecraft. In the second case (section 5.2), being the initial orbit elliptical, the escape phase from the Earth is particularly sensitive to the three-body dynamics and the optimal solution differs pretty much from the first guess solution. In fact a fly-by of the Earth is scheduled, by the DDP algorithm, which was not included in the first guess trajectory. The transfer trajectory to asteroid 2002 AA29 (section 5.3) presents as well some interesting features, in correspondence of the passage in vicinity of the Lagrangian point L2.

5.1 Rendezvous with asteroid Apophis

The problem is to design an optimal low-thrust trajectory to rendezvous the asteroid Apophis, starting from an initial circular orbit, lying on the Earth equatorial plane. The spacecraft has an initial mass of 500 kg and is equipped with an engine capable of delivering a variable thrust at a fixed specific impulse $I_{sp} = 3250$ s. A first guess solution for the transfer is computed with the global search procedure illustrated in the preceding section. The departure date and transfer time were imported from the first guess. Table 1 summarises the main mission parameters.

Table 1: Mission parameters

Initial mass	500 kg
Specific impulse	3250 s
Departure date	19/08/2023 (8630.95 MJD2000)
Time of flight	990.4 d
Initial orbit radius	42328 km

The optimal solution found has a propellant mass consumption of 133.15 kg and the thrust profile represented in Fig. 8. The first guess (dashed line) is obtained with tangential thrust 0.15 N until exiting the Earth sphere of influence and 0.0109 N afterwards. The value of the thrust magnitude outside the Earth sphere of influence was imported from the first guess. The thrust evolution has an oscillatory behaviour with the spiralling-out from the Earth and the oscillation of the x and y-components are higher than the one along the z-axis (see Fig. 9a). The mass evolution follows approximately the first guess solution along the spiralling-out phase, while moves away from the first guess solution when out of the Earth gravitation (see Fig. 9b). Note that the optimal solution has a mass consumption higher than the first guess because the constraints are satisfied under the required tolerance. The constraints violation in km and km/s of the first guess solution is:

$$\left| \mathbf{s}_{N+1} - \mathbf{s}_{\text{target}} \right|_{\text{first guess}}^T = [3708314 \quad 9936689 \quad 8362980 \quad 0.51 \quad 3.09 \quad 0.038]$$

while is under the required tolerance for the optimal solution:

$$\left| \mathbf{s}_{N+1} - \mathbf{s}_{\text{target}} \right|_{\text{optimal}}^T = [4210.7 \quad 8089 \quad 1481.4 \quad 0.006 \quad 0.0045 \quad 0.00006]$$

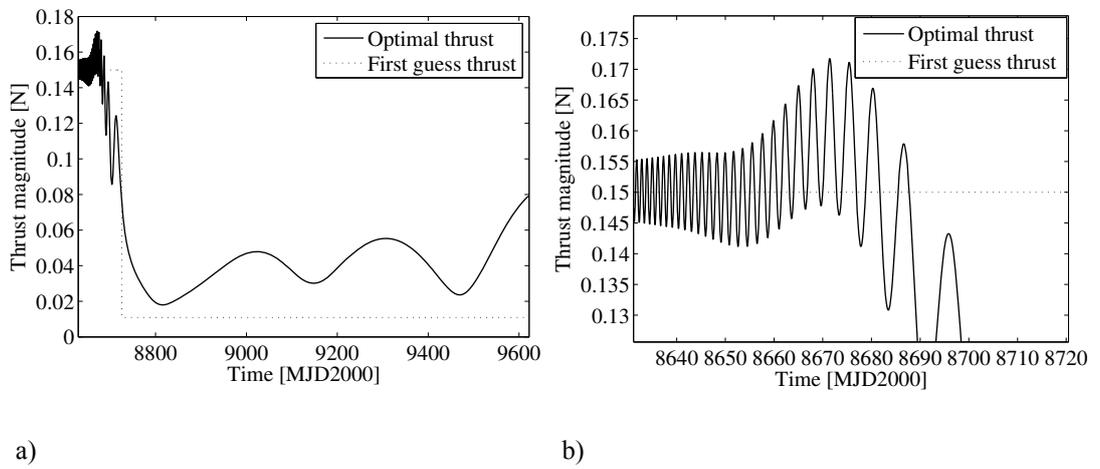


Fig. 8 Thrust magnitude. The dashed line represents the first guess solution provided to the DDP algorithm, the continuous line is the optimal thrust profile. a) Entire trajectory. b) Close-up on the escape phase.

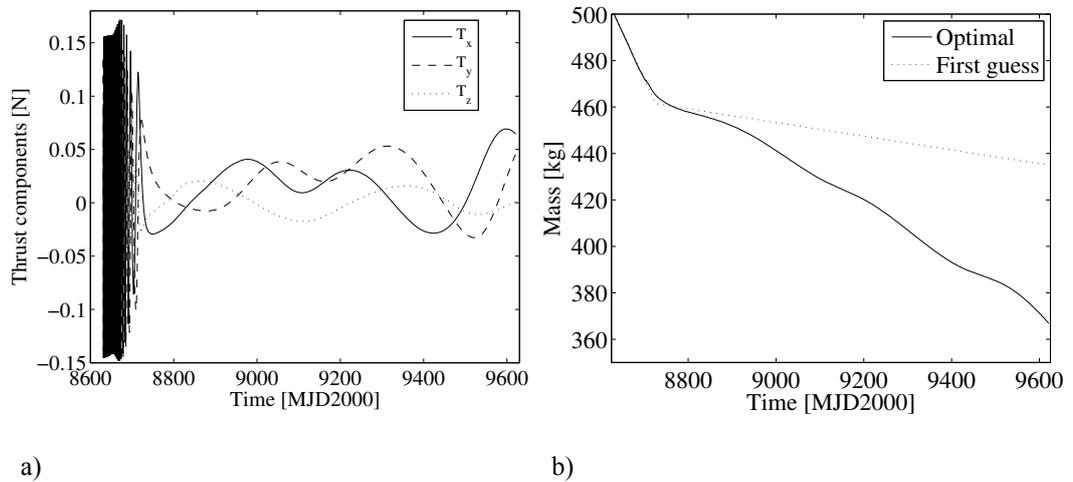


Fig. 9 Time evolution of thrust and mass. a) Thrust components. b) Mass. The dashed line represents the first guess solution; the continuous line is the optimal profile.

The transfer trajectory (see Fig. 10) is represented in the Earth inertial system, the reference frame used for the optimisation process.

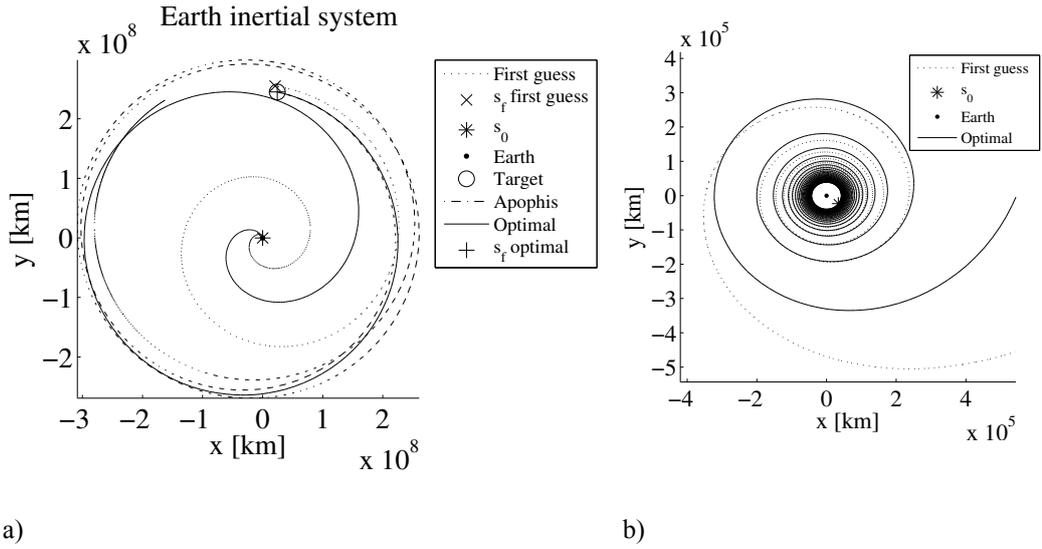


Fig. 10 Rendezvous trajectory to Apophis represented in the Earth inertial reference frame. a) Entire trajectory. b) Close-up on the escape phase.

Fig. 11 and Fig. 12 show respectively the trajectory represented in the Sun inertial frame and the time evolution of semi-major axis, inclination and eccentricity, during the escape phase (computed with respect to the Earth relative system, until the semi-major axis becomes negative and the eccentricity becomes smaller than 1). The optimal solution is characterised by a monotonic increase of the semi-major axis (see Fig. 11b).

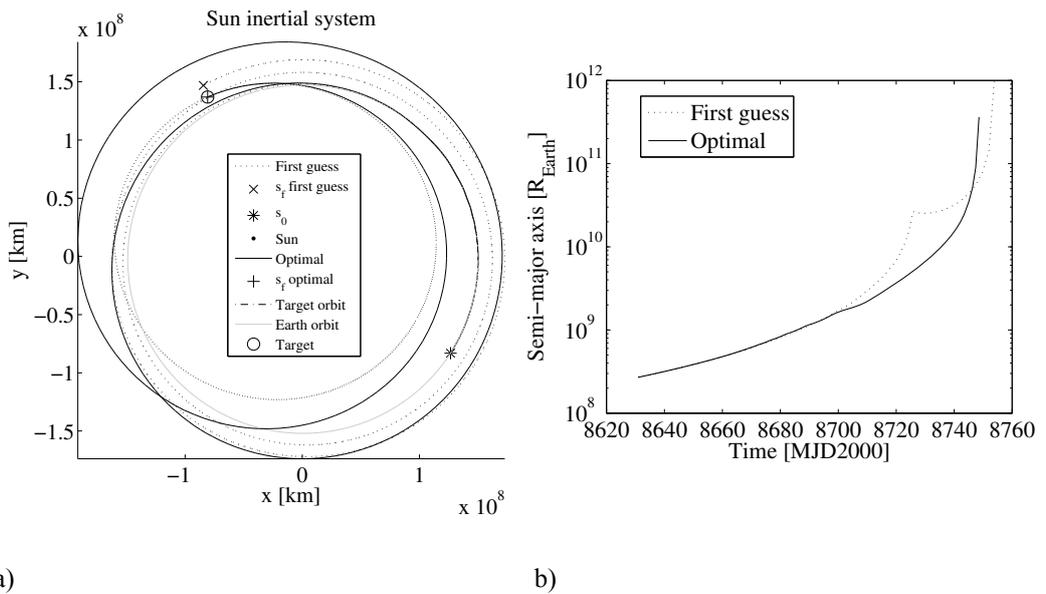


Fig. 11 Trajectory to Apophis rendezvous. a) Transfer in the Sun inertial reference frame. The dashed line represents the first guess transfer solution; the continuous line is the optimal trajectory.

Apophis and Earth orbit are represented respectively in dashed-dotted and gray continuous lines.
 b) Semi-major axis evolution during the escape phase expressed in Earth radii.

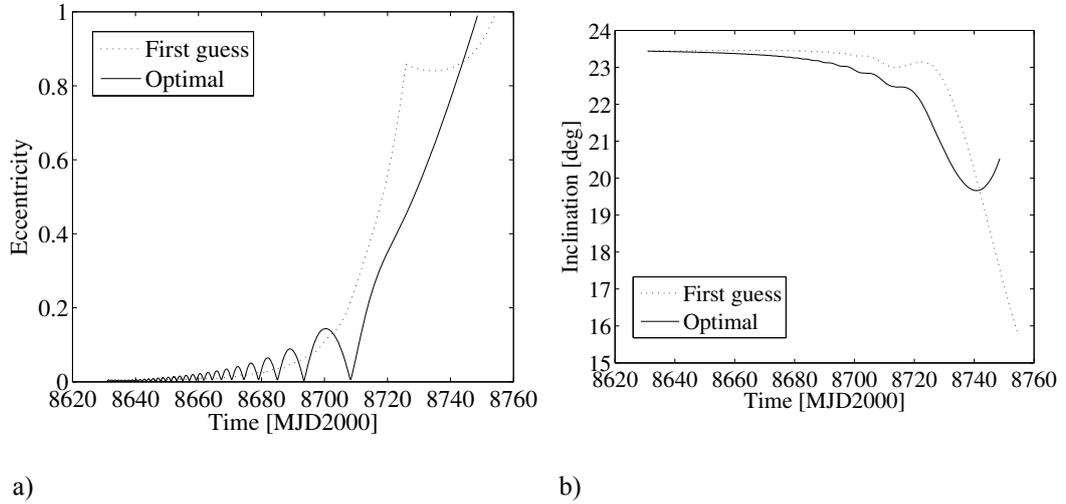


Fig. 12 Evolution of the Keplerian elements during the escape phase. The dashed line represents the first guess solution; the continuous line is the optimal profile. a) Eccentricity. b) Inclination.

We then studied the evolution of the objective function for different times of flight, in the range of [700 1450] days. The result is reported in Fig. 13.

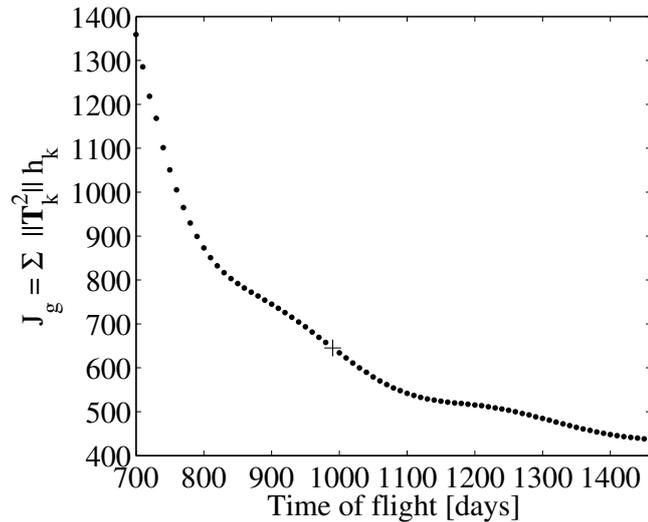


Fig. 13 Time of flight sensitivity. The integral term of the cost function (normalised to the weight parameter w) is represented on the y axis. Each point represents an optimised transfer (with final constraints satisfied) with a given time of flight. The cross shows the result corresponding to the solution fully presented in the previous section.

5.2 Rendezvous with asteroid Apophis from a Geostationary Transfer Orbit

Another mission to asteroid Apophis was studied, with departure from a Geostationary Transfer Orbit (GTO). The arrival date was fixed to 19/08/2033 (12283.5 MJD since 2000), based on a previous study on mission to deviate asteroid Apophis. In fact, this launch allows having 1000 days (see Colombo et al.) before the possible impact of the asteroid with the Earth in 15/05/2036. The spacecraft, with initial mass of 1300 kg, is equipped with a low-thrust engine able to deliver a variable thrust at a fixed specific impulse of 3250 s.

The parking orbit of this transfer is a Geostationary Transfer Orbit (GTO) with an inclination of 23 deg with respect to the ecliptic; the transfer orbit injection point was fixed at the pericentre of the GTO and a midday launch is considered. The parking orbit parameters are reported in Table 2 and the other mission parameters are summarised in Table 3; the value of the time of flight and the magnitude of the thrust out of the sphere of influence of the Earth were fixed from the first guess solution. The DDP algorithm was used to find the solution to the optimal control problem and to satisfy the final constraints. The constraints violation in km and km/s of the first guess solution is:

$$\left| \mathbf{s}_{N+1} - s_{\text{target}} \right|_{\text{first guess}}^T = [20041710 \quad 60297.6 \quad 1954650 \quad 3.07 \quad 1.5 \quad 1.19]$$

while is under the required tolerance in the optimal solution:

$$\left| \mathbf{s}_{N+1} - s_{\text{target}} \right|_{\text{optimal}}^T = [24 \quad 3.4 \quad 24.7 \quad 0.00046 \quad 0.0086 \quad 0.0017]$$

Table 2: Parking orbit parameters

Apocentre height	35950 km
Pericentre height	500 km
Inclination	23.44 deg
Anomaly of the ascending node	0 deg
Anomaly of the pericentre	185.24 deg (midday launch)
True anomaly	0 deg (pericentre)

Table 3: Mission parameters

Initial mass	1300 kg
Specific impulse	3250 s
Departure date	28/09/2029 (10862.6 MJD2000)
Time of flight	1420.9 d

The optimal solution found has a propellant mass consumption of 336.95 kg; the optimal solution has a higher mass consumption because the final constraints are satisfied (see Fig. 15a). The thrust profile is represented in Fig. 14 and Fig. 15b. The first guess (dashed line) is obtained with tangential thrust 1 N until exiting the Earth sphere of influence and 0.0374 N afterwards. The oscillatory behaviour of the thrust with the spiralling-out from the Earth (see Fig. 14b and Fig. 15b) causes small oscillations of the instantaneous eccentricity around the initial value, while the eccentricity of the first guess solution (tangential thrust) decreases with time. This can be appreciated in Fig. 16, which represents the evolution of the eccentricity with respect to the time and a close-up of the spiralling-out phase.

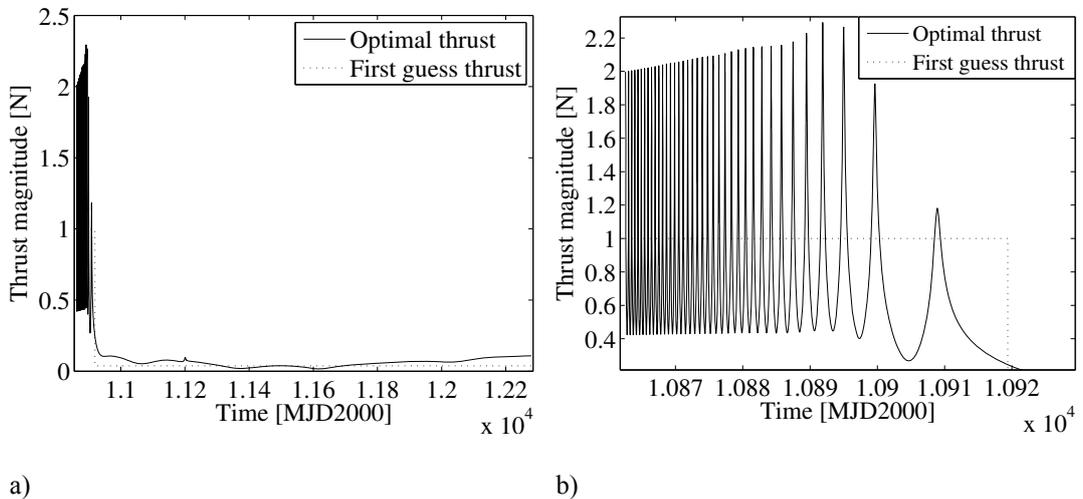


Fig. 14 Thrust magnitude. The dashed line represents the first guess solution provided to the DDP algorithm, the continuous line is the optimal thrust profile. a) Entire trajectory. b) Close-up on the escape phase.

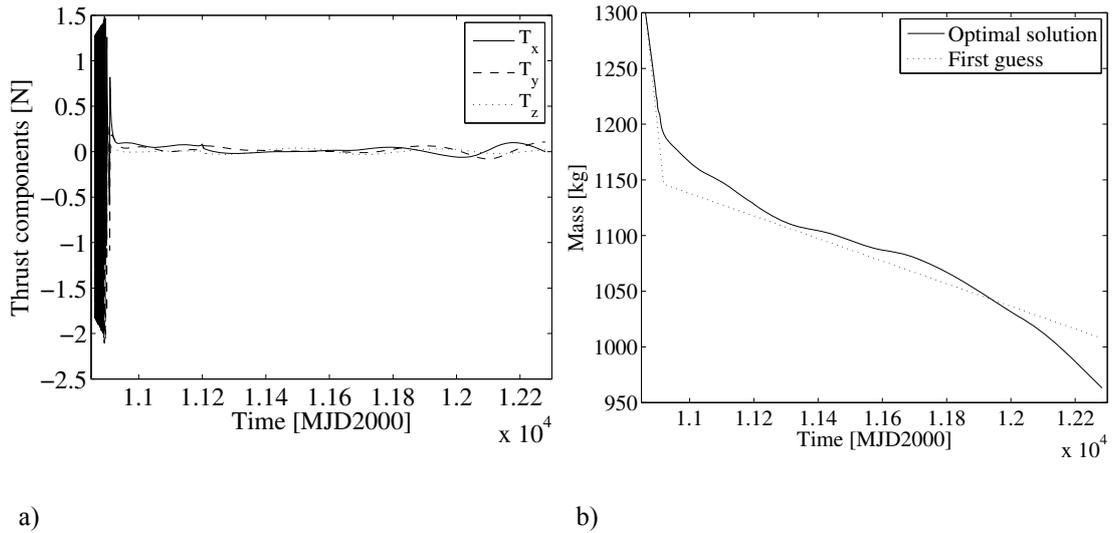


Fig. 15 Time evolution of thrust and mass. a) Thrust components. b) Mass. The dashed line represents the first guess solution; the continuous line is the optimal profile.

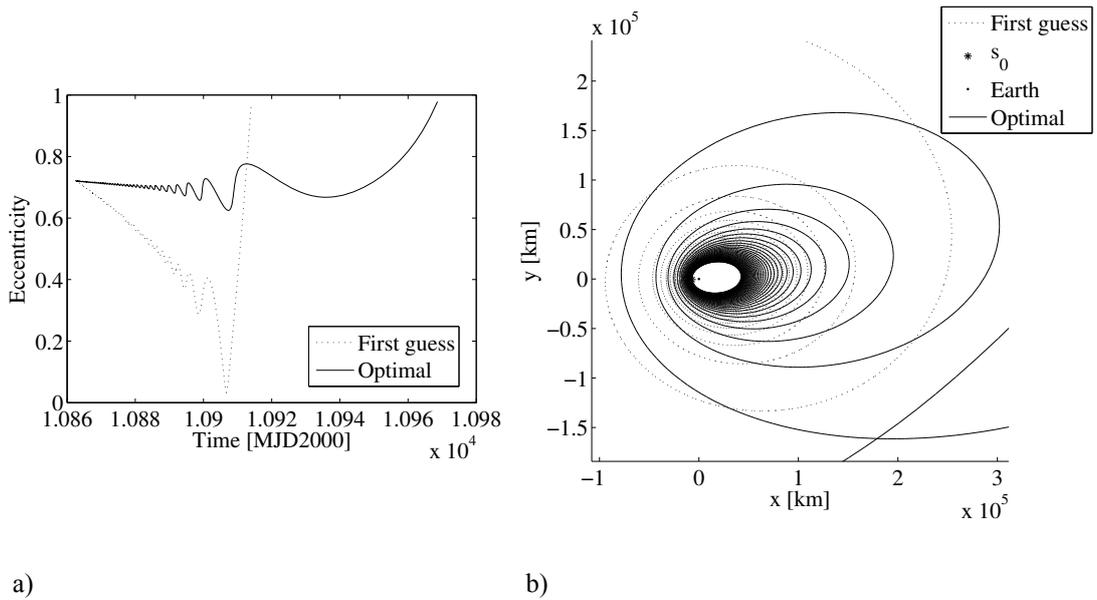
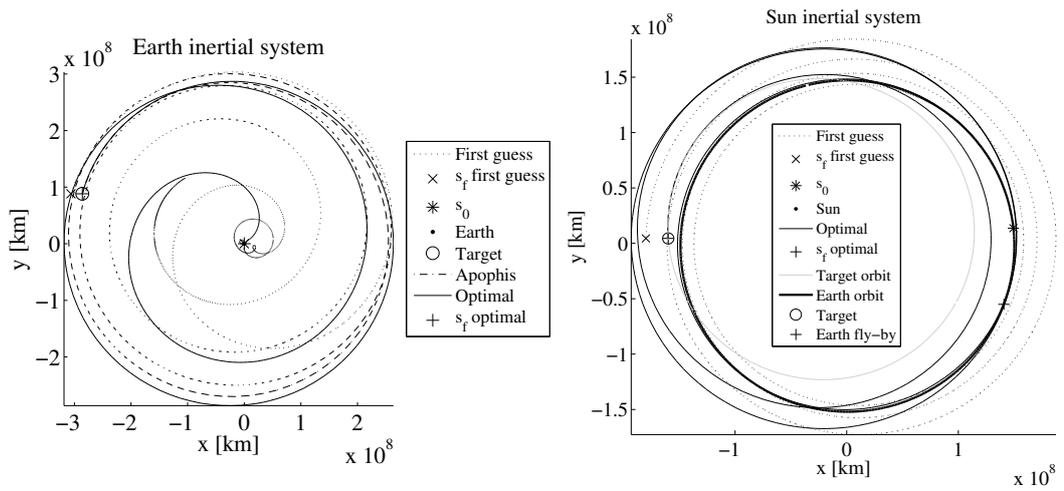


Fig. 16 Spiralling-out phase. The dashed line represent the first guess, the continuous line is the optimal solution a) Evolution of the instantaneous eccentricity with time. b) Close-up on the escape phase.

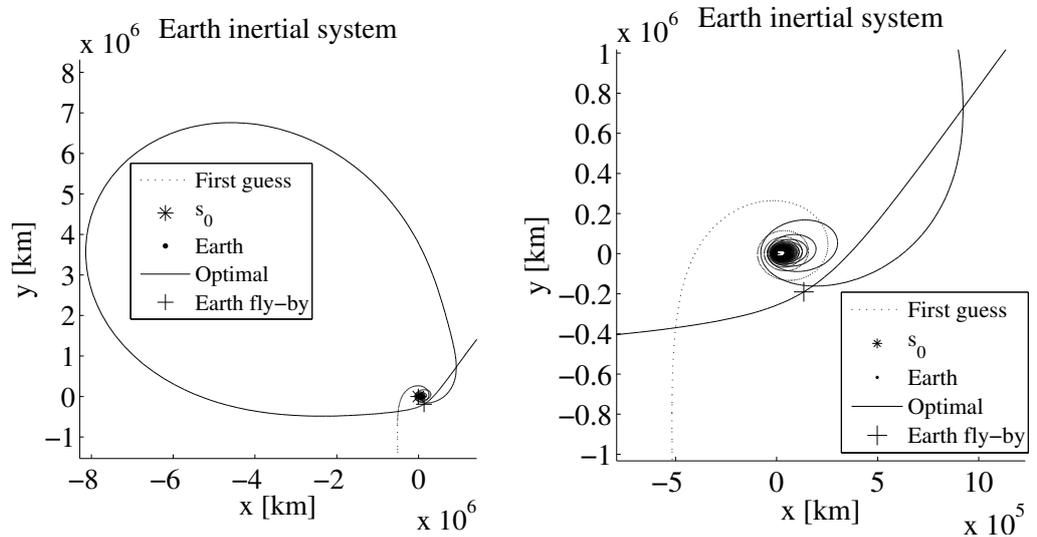
Fig. 17 shows the whole transfer trajectory in the Earth inertial reference frame (see Fig. 17a) and in the Sun inertial reference frame (see Fig. 17b). The dashed line is the first guess solution; the continuous line is the optimal solution. The solution found through the DDP algorithm presents a fly-by of the Earth that was not imposed in the first guess solution. The fly by is indicated in Fig. 17b with a cross and is show in Fig. 18 in the Earth inertial reference frame.



a)

b)

Fig. 17 Trajectory to Apophis rendezvous. The dashed line represents the first guess transfer solution; the continuous line is the optimal trajectory. a) Transfer in the Earth inertial reference frame. The circle represent the target position, the cross is the final state of the optimal trajectory. b) Transfer in the Sun inertial reference frame. Apophis and Earth orbit are represented respectively in dashed-dotted and gray continuous lines.



a)

b)

Fig. 18 Fly-by of the Earth. The cross represents the pericentre of the hyperbola with respect to the Earth. a) Fly-by phase. b) Close-up of the passage from the pericentre of the hyperbola.

Fig. 19 and Fig. 20 are shown to demonstrate the presence of the fly-by phase. Fig. 19 depicts the evolution of the thrust magnitude and the velocity magnitude

during the fly-by phase, Fig. 20 shows the in-plane angle α and the out-of-plane δ of the velocity with respect to the inertial reference frame centred at the Earth. The peak in all the graphs in Fig. 19 and Fig. 20, in fact, is in correspondence of the passage from the pericentre (cross symbol in the figures). The velocity magnitude is almost unchanged at the entrance and exit from the fly-by (see Fig. 19a); while the in-plane and out-of plane angles, which represent the direction of the velocity vector, have a quasi-instantaneous change in correspondence of the pericentre passage (see Fig. 20a and Fig. 20b). In correspondence of the pericentre passage, a peak of the optimal control thrust is scheduled; this allows the following escape from the Earth.

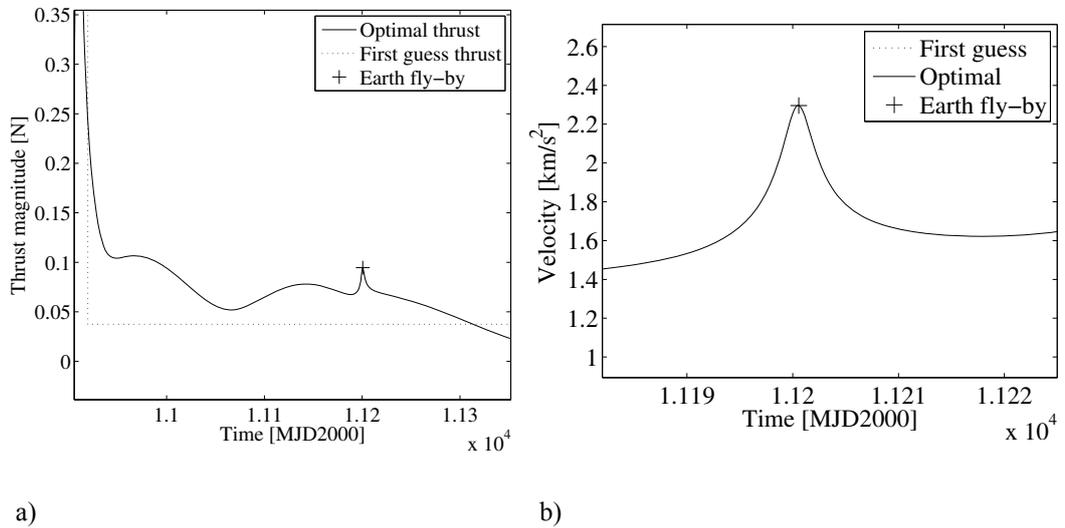


Fig. 19 Evolution of the thrust and velocity magnitude during the fly-by. The dashed line represents the first guess solution; the continuous line is the optimal profile. The cross symbol is in correspondence with the pericentre passage. a) Thrust magnitude. b) Velocity magnitude with respect to the Earth.

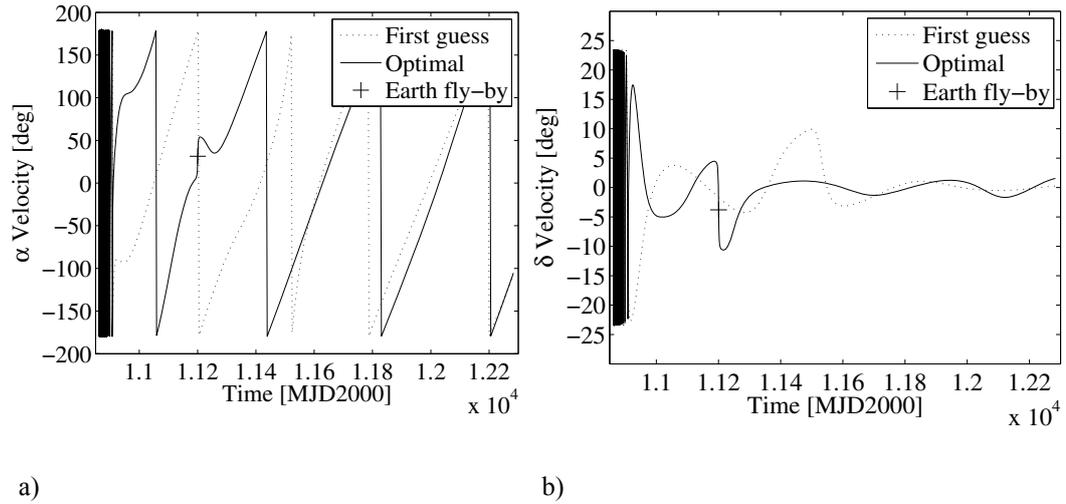


Fig. 20 Evolution of the angles of the velocity vector with respect to the Earth inertial reference frame, during the fly-by. The dashed line represents the first guess solution; the continuous line is the optimal profile. The cross symbol is in correspondence of the pericentre passage. a) In-plane angle of the velocity vector. b) Out-of-plane angle of the velocity vector.

5.3 Fly-by of asteroid 2002 AA29

Asteroid 2002 AA29 is a near-Earth asteroid characterised by a “horseshoe orbit” with a full revolution of 95 years (see Fig. 21a). The latest nearest approach of the asteroid to the Earth was in January 2003, after that it reversed its direction once again[§]. A mission to the fly-by of 2002 AA29 was studied, whose parameters are reported in Table 4. The initial orbit is circular on the Earth equatorial plane.

Table 4: Mission parameters

Initial mass	500 kg
Specific impulse	2500 s
Departure date	27/04/2003 (1212.2 MJD2000)
Time of flight	256.6 d
Initial orbit radius	42328 km

The trajectory in the Sun inertial reference frame is depicted in Fig. 21b, while Fig. 22 reports the thrust magnitude with a close-up on the spiralling-out from the Earth. The dashed line shows the magnitude of the first guess thrust: a constant thrust of 0.15 N is planned until a distance equal to the Earth’s sphere of influence

is reached; afterwards a constant thrust of 0.0088 N is applied along the direction of the velocity around the Sun. The continuous line in Fig. 22 is the optimal solution computed through the DDP method. Fig. 23 represents the time evolution of the thrust components and the mass of the spacecraft. The propellant mass needed for the asteroid interception is 49.3 kg.

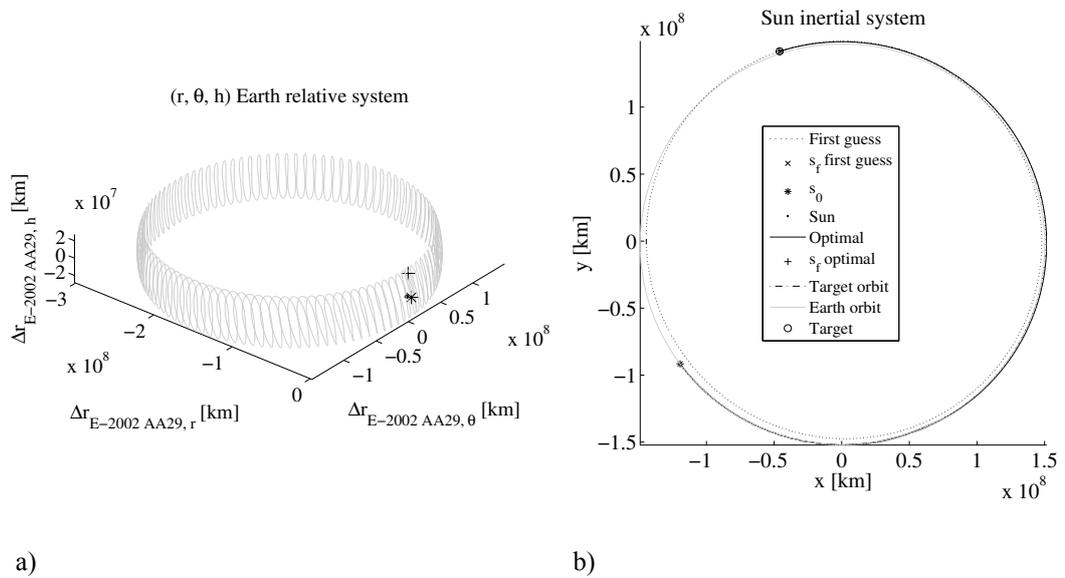
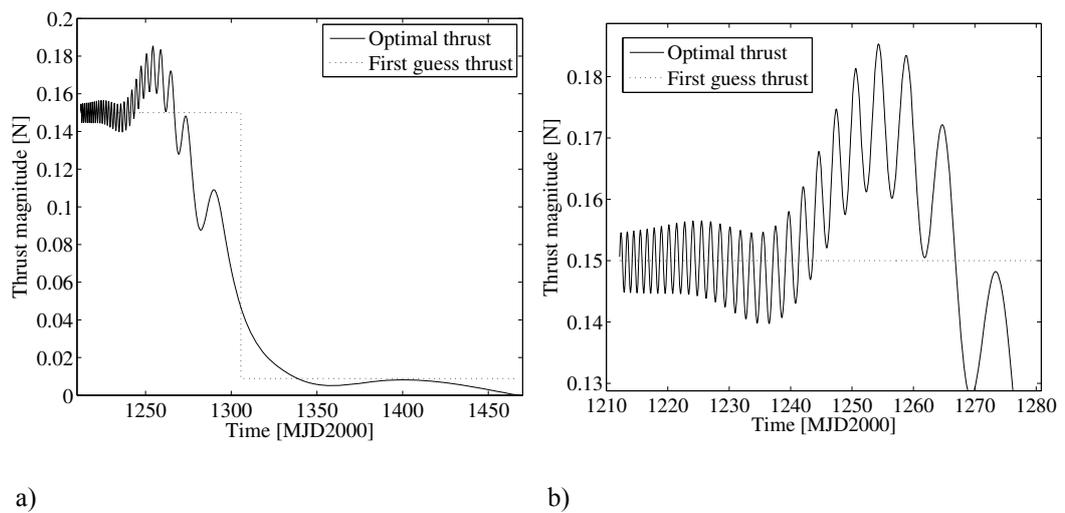


Fig. 21 Trajectory. a) Asteroid 2002 AA29 relative motion with respect to the Earth. b) Transfer trajectory to 2002 AA29 fly-by in the Sun inertial reference frame. The dashed line represents the first guess transfer solution; the continuous line is the optimal trajectory. 2002 AA29 and Earth orbit are represented respectively in dashed-dotted and gray continuous lines.



[§] <http://neo.jpl.nasa.gov/2002aa29.html>

Fig. 22 Thrust magnitude. The dashed line represents the first guess solution provided to the DDP algorithm, the continuous line is the optimal thrust profile. a) Entire trajectory. b) Close-up on the escape phase.

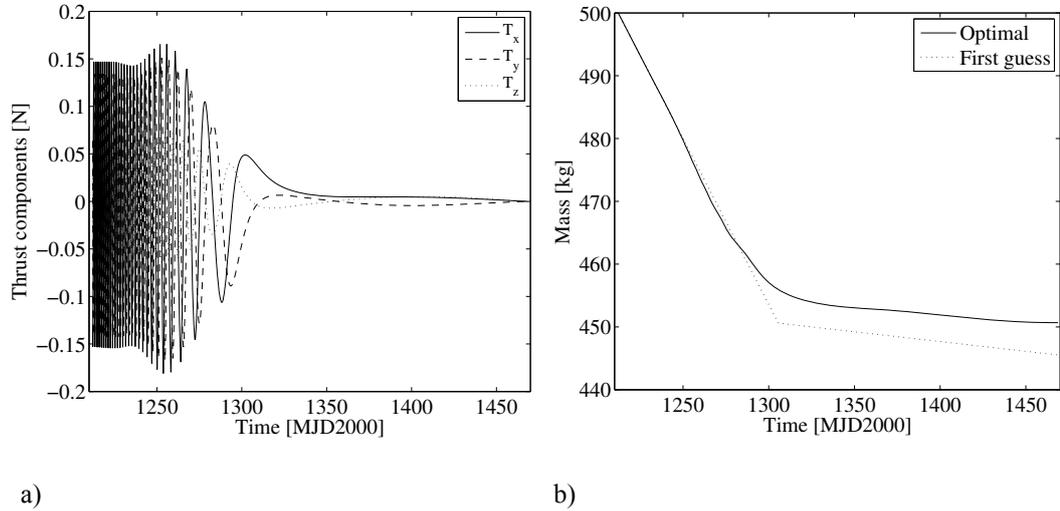
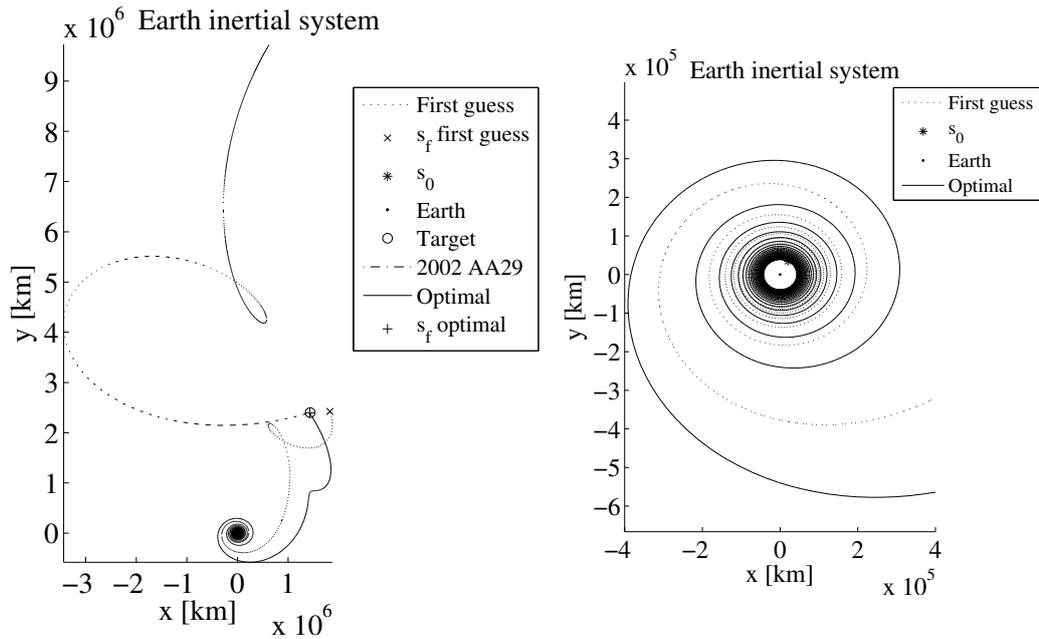


Fig. 23 Time evolution of thrust and mass. a) Thrust components. b) Mass. The dashed line represents the first guess solution; the continuous line is the optimal profile.

Fig. 24 represents the trajectory in the Earth inertial reference frame. The tolerance required for the fly-by of the asteroid is a maximum error of 10000 km on the components of the relative position to the asteroid. No constraints on the velocity were imposed, instead; hence the spacecraft intercepts the asteroid with a relative velocity of 5.56 km/s. The dashed line represents the first guess trajectory which has a constraints violation on the three components of the position of $\Delta \mathbf{r} = \{390574 \quad 24805 \quad -908.6\}^T$ km, the continuous line indicates the optimal solution for the trajectory, with a violation of the position at the asteroid of $\Delta \mathbf{r} = \{-8278.5 \quad -5982.5 \quad -1803.1\}^T$ km. The dashed-dot line describes the motion of 2002 AA29 with respect to the Earth inertial system.

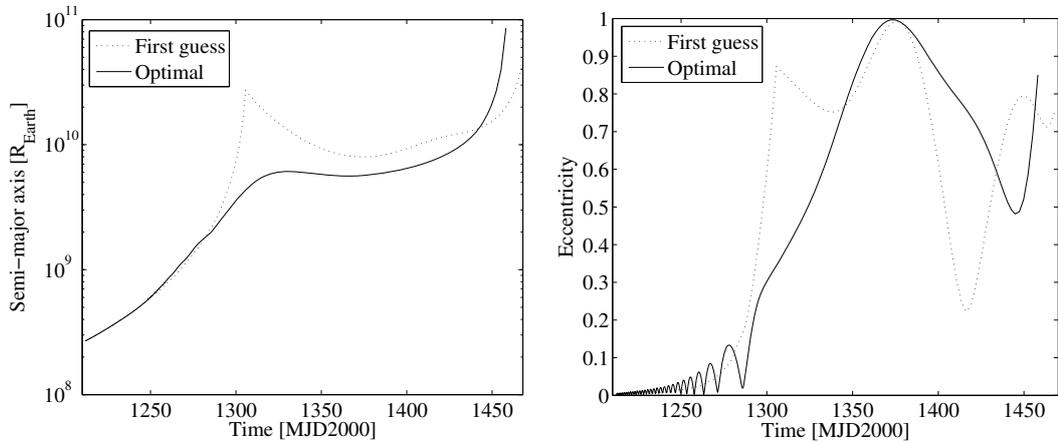


a)

b)

Fig. 24 Rendezvous trajectory to 2002 AA29 represented in the Earth inertial reference frame. a) Entire trajectory. b) Close-up on the escape phase.

Fig. 25 and Fig. 26 report the trend of the instantaneous Keplerian elements (computed with respect to the Earth relative system) along the trajectory until the escape from the Earth (i.e., semi-major axis become negative and eccentricity becomes bigger than 1). The escape occurs slightly before for the optimal trajectory than the first guess one.



a)

b)

Fig. 25 Evolution of the Keplerian elements during the escape phase. The dashed line represents the first guess solution; the continuous line is the optimal profile. a) Semi-major axis. b) Eccentricity.

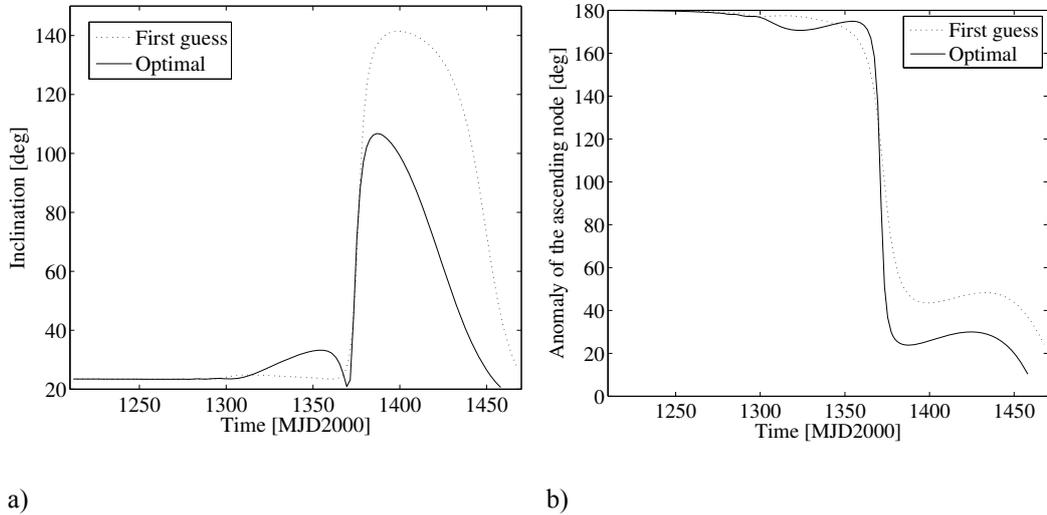


Fig. 26 Evolution of the Keplerian elements during the escape phase. The dashed line represents the first guess solution; the continuous line is the optimal profile. a) Inclination. b) Anomaly of the ascending node.

As it can be seen from Fig. 26, there is a sudden change of the orbital elements, especially inclination and anomaly of the ascending node, in a range of 20 days between 1370 and 1390 MJD since 2000. This occurs when the spacecraft passes in vicinity of the Lagrangian point L2, as it can be appreciated from Fig. 27. When passing in the vicinity of L2, a small change in the direction of the thrust vector (see Fig. 28) produces a big variation of the orbital elements.

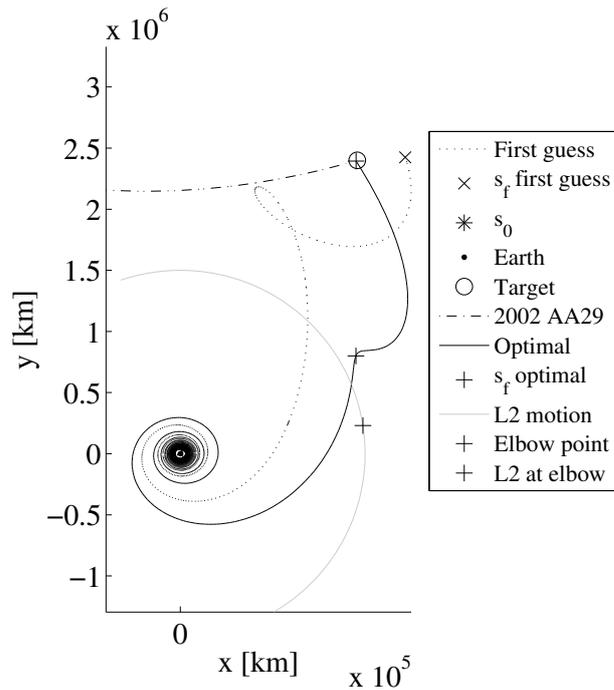


Fig. 27 Lagrange point passage. The cross highlines the position of the Lagrange point L2 when the trajectory changes its inclination.

Fig. 28 shows the angles of the thrust vector, the in-plane right ascension angle (Fig. 28a), counted from the tangential direction along the velocity vector to the projection of the thrust vector on the orbital plane and the out-of-plane declination angle (Fig. 28b) from the projection of the thrust vector on the orbital plane up to the thrust vector itself.

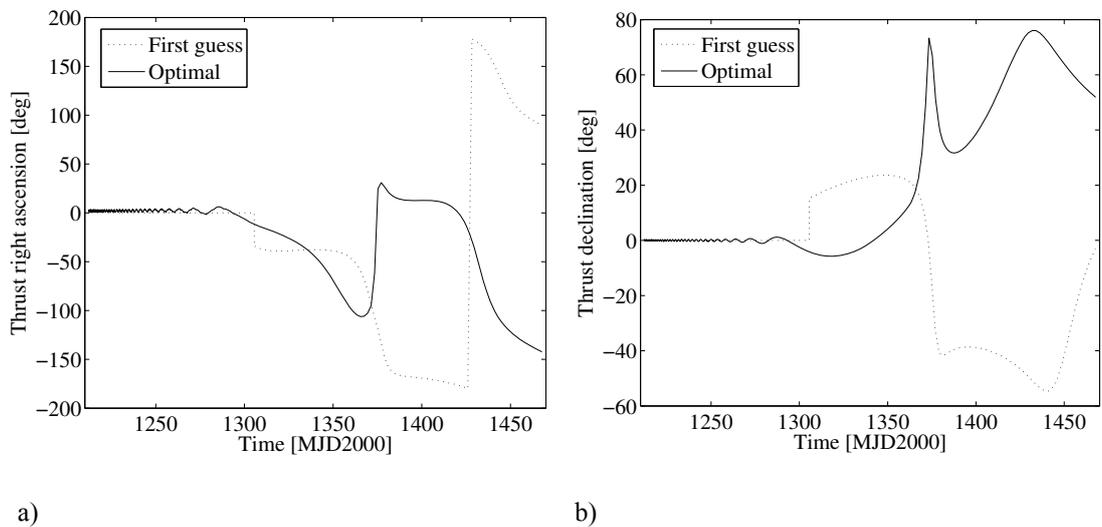


Fig. 28 Angles of the thrust vector. The dashed line represents the first guess solution; the continuous line is the optimal profile. a) Right ascension. b) Declination.

Finally Fig. 29 represents the components of the acceleration acting on the spacecraft, in the first guess (dashed line) and optimal (continuous line) solution. The components represented are respectively the acceleration due to the Earth's gravity field a_E (black lines), the disturbing components due to the interaction between Sun-Earth and Sun-spacecraft a_d (bold black lines), and the acceleration produced by the engines, a_T (bold gray lines). Focusing on the acceleration magnitude (Fig. 29a) it can be noticed that around 1370 MJD since 2000, the acceleration component due to the Sun become bigger than the Earth's gravitation. Fig. 29b, c and b, instead, contain the x, y and z component of the acceleration.

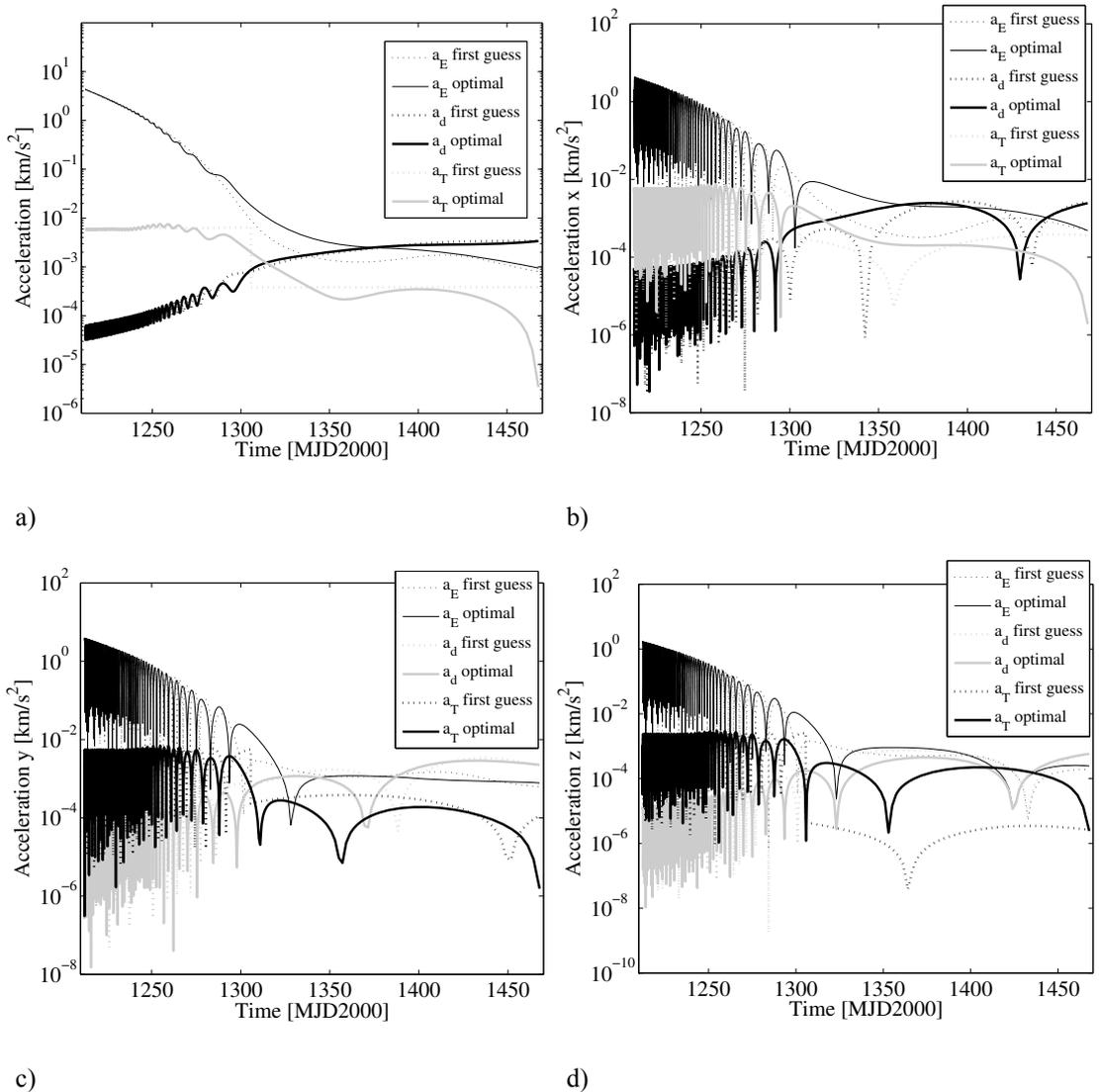


Fig. 29 Acceleration components. The dashed line represent the first guess solution, the continuous line is the optimal solution. The black line indicated the acceleration due to the Earth's gravity field, the black bold line indicates the disturbing acceleration due to the Sun and the bold gray line indicates the thrust acceleration. a) Acceleration magnitude. b) x component of the acceleration. c) y component of the acceleration. d) z component of the acceleration.

6 Conclusion

This paper presents a modified Differential Dynamic Programming algorithm for the optimisation of low-thrust trajectories. The principal advantage of the proposed algorithm is that the problem is discretised in a number of decision steps, so that the optimisation process requires the solution of a great number of small dimensional problems (one for each stage). The stage-wise approach allows for the use of an accurate adaptive integration of the dynamics during the optimisation process. The main advantage is that high fidelity dynamic model can be used. A Runge-Kutta-Fehlberg integration scheme was incorporated in the DDP scheme, together with a particular interpolation technique that preserves the feedback nature of the control variation. This particular technique improves the robustness of the algorithm against some approximation errors that are introduced during the adaptation process. A further increase in robustness was obtained by the use of global control variations, which showed to be more appropriate than the small control variations algorithm, to the solution of the problem presented in this paper.

In particular, the case of a transfer to asteroid Apophis, starting from a Geostationary Transfer Orbit around the Earth, demonstrates as differential dynamic programming is able to introduce additional fly-by, not included in the first guess solution. Further work will extend the present method to deal with minimum final mass problem and with time of flight and departure date given implicitly.

7 Acknowledgements

The authors would like to thank Dr. Paolo De Pascale of the European Space Operation Centre for his support during the preliminary study of the differential dynamic programming technique.

8 References

- Bellman, R.: 1957, **Dynamic Programming**, Princeton University Press, Princeton, New Jersey.
- Bertsekas, D. P.: 2005-2007, **Dynamic Programming and Optimal Control**, 3rd Edition, Athena Scientific, Belmont, MA.
- Betts, J. T. and, Erb, S. O.: 2003, 'Optimal Low Thrust Trajectories to the Moon', **SIAM Journal on Applied Dynamical Systems**, Vol. 2, No. 2, pp. 144-170.
- Betts, J. T.: 2000, **Practical Methods for Optimal Control using Nonlinear Programming**.
- Betts, J. T.: 1998, 'Survey of Numerical Methods for Trajectory Optimization', **Journal of Guidance, Control, and Dynamics**, Vol. 21, No. 2, March-April 1998, pp.193-207, doi:10.2514/2.4231.
- Casalino, L., Colasurdo, G., Pastrone, D.: 1999, 'Optimal Low-Thrust Escape Trajectories Using Gravity Assist', **Journal of Guidance, Control, and Dynamics**, Vol. 22, No. 5, September-October 1999, pp. 637-642.
- Colasurdo, G. and Casalino, L.: 1999, 'Trajectories towards Near-Earth-Objects Using Solar Electric Propulsion', **Advances in the Astronautical Sciences**, Vol. 103, Astroynamics 1999, (AAS 99-339), pp. 593-608.
- Coleman, T. F. and Li, Y.: 1996, 'An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds', **SIAM Journal on Optimization**, Vol. 6, pp. 418-445.
- Coleman, T. F. and Li, Y.: 1994, 'On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds', **Mathematical Programming**, Vol. 67, Vol. 2, pp. 189-224.
- Colombo, C., Vasile M., Radice G.: 2009, 'Semi-Analytical Solution for the Optimal Low-Thrust Deflection of Near Earth Objects', **Journal of Guidance, Control, and Dynamics**, Vol. 32, No. 3, doi:10.2514/1.40363.
- Conway, B., Chilan, C., Wall, B.: 2007, 'Evolutionary principles applied to mission planning problems', **Celestial Mechanics and Dynamical Astronomy**, Vol. 97, No. 2, pp. 73-86, doi:10.1007/s10569-006-9052-7.
- De Boor, C.: 1978, **A Practical Guide to Splines**, Springer-Verlag.
- Dormand, J. R. and Prince, P. J.: 1980, 'A family of embedded Runge-Kutta formulae', **Journal of Computational and Applied Mathematics**, Vol. 6, pp. 19-26.
- Enright, P. J. and Conway, B. A.: 1991, 'Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming', **Journal of Guidance, Control, and Dynamics**, Vol.14, No.5, pp. 981-985, doi:10.2514/3.20739.
- Gershwin, S. B. and Jacobson, D. H.: 1970, 'A Discrete-Time Differential Dynamic Programming with Application to Optimal Orbit Transfer', **AIAA Journal**, Vol. 8, No. 9, pp. 1616-1626.
- Gill, P. E., Murray, W. and Wright, M. H.: 1981, **Practical Optimization**, Academic Press.
- Guelman, M.: 1995, 'Earth-to-Moon Transfer with a Limited Power Engine', **Journal of Guidance, Control, and Dynamics**, Vol. 18, No. 5, September-October 1995, pp.1113-1138.

- Herman, A. L. and Spencer, D. B.: 2002, 'Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods', **Journal of Guidance, Control, and Dynamics**, Vol. 25, No. 1, January–February 2002, pp. 40-47, doi:10.2514/2.4873.
- Jacobson, D. H. and Mayne, D. Q.: 1969, **Differential Dynamic Programming**, American Elsevier, New York.
- Kluever, C. A.: 1997, 'Optimal Low-Thrust Interplanetary Trajectories by Direct Method Techniques', **Journal of Astronautical Sciences**, Vol. 45, No. 3, pp. 247–262.
- Kluever, C. A. and Pierson, B. L.: 1995, 'Optimal Low-Thrust Three-Dimensional Earth-Moon Trajectories', **Journal of Guidance, Control, and Dynamics**, Vol. 18, No. 4, July-August 1995, pp. 830-837, doi:10.2514/3.21466.
- Lantoine, G. and Russel, R. P.: 2008, 'A Hybrid Differential Dynamic Programming Algorithm for Low-Thrust Optimization', **AIAA/AAS Astrodynamics Specialist Conference and Exhibit**, 18-21 August 2008, Honolulu, Hawaii.
- Liao, L. and Shoemaker, C. A.: 1992, 'Advantages of Differential Dynamic Programming over Newton's Method for Discrete-Time Optimal Control Problems', **Technical Report Cornell University**, July 1992, available at <http://ecommons.library.cornell.edu/handle/1813/5474>.
- Mayne, D. A.: 1966, 'A Second Order Gradient Method for Determining Optimal Trajectories for Nonlinear Discrete-Time Systems', **International Journal on Control**, Vol. 3, 1966, pp. 85-95.
- Murray, D. M.: 1978, 'Differential Dynamic Programming for the Efficient Solution of Optimal Control Problems', Ph.D. Dissertation, Department of Mathematics, University of Arizona, Tucson, Arizona, University Microfilm Inc.
- Murray, D. M. and Yakowitz, S. J.: 1984, 'Differential Dynamic Programming and Newton's Method for Discrete Optimal Control Problem', **Journal of Optimization Theory and Applications**, Vol. 43, pp. 395-414.
- Nah, R. S., Vadali, S. R. and Braden, E.: 2001, 'Fuel-Optimal, Low-Thrust, Three-Dimensional Earth–Mars Trajectories', **Journal of Guidance, Control, and Dynamics**, Vol. 24, No. 6, November–December 2001, pp. 1100-1107, doi:10.2514/2.4844.
- Olympio, J. T.: 2008, 'Algorithm for Low-Thrust Optimal Interplanetary Transfers with Escape and Capture Phases', **AIAA/AAS Astrodynamics Specialist Conference and Exhibit**, 18-21 August 2008, Honolulu, Hawaii.
- Perozzi, E., Casalino, L., Colasurdo, G., Rossi, A. and Valsecchi, G. B.: 2002, 'Resonant Fly-by Missions to Near Earth Asteroids', **Celestial Mechanics and Dynamical Astronomy**, Vol. 83, No. 1, pp. 49-62, doi:10.1023/A:1020122511548.
- Pierson, B. L. and Kluever, C. A.: 1994, 'Three-Stage Approach to Optimal Low-Thrust Earth-Moon Trajectories', **Journal of Guidance, Control, and Dynamics**, Vol. 17, No. 6, November-December 1994, pp.1275-1282, doi:10.2514/3.21344.
- Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., Mishenko E. F.: 1962, **The Mathematical Theory of Optimal Processes**, Wiley (Interscience), New York.
- Price, K., Storn, R. and Lampinen, J.: 2005, **Differential Evolution - A Practical Approach to Global Optimization**, Springer, Berlin.

- Racca, G. D.: 2003, 'New Challenges to Trajectory Design by the Use of Electric Propulsion and Other New Means of Wandering in the Solar System', **Celestial Mechanics and Dynamical Astronomy**, Vol. 85, No. 1, pp. 1-24, doi:10.1023/A:1021787311087.
- Ranieri, C. L. and Ocampo, C. A.: 2006, 'Indirect Optimization of Spiral Trajectories', **Journal of Guidance, Control, and Dynamics**, Vol. 29, No. 6, November-December 2006, pp. 1360-1366, doi:10.2514/1.19539.
- Ranieri, C. L. and Ocampo, C. A.: 2005, 'Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories via an Indirect Method', **Journal of Guidance, Control, and Dynamics**, Vol. 28, No. 2, March-April 2005, pp. 306-314, doi:10.2514/1.5540.
- Scheel, W. A. and Conway, B. A.: 1994, 'Optimization of Very-Low-Thrust, Many-Revolution Spacecraft Trajectories', **Journal of Guidance, Control, and Dynamics**, Vol. 17, No. 6, November-December 1994, pp. 1185-1192, doi:10.2514/3.21331.
- Tang, S. and Conway, B. A.: 1995, 'Optimization of Low-Thrust Interplanetary Trajectories Using Collocation and Nonlinear Programming', **Journal of Guidance, Control, and Dynamics**, Vol. 18, No. 3, May-June 1995, pp. 599-604, doi:10.2514/3.21429.
- Vadali, S. R., Nah, R. S., Braden, E. and Johnson Jr., I. L.: 2000, 'Fuel-Optimal Planar Earth-Mars Trajectories Using Low-Thrust Exhaust-Modulated Propulsion', **Journal of Guidance, Control, and Dynamics**, Vol. 23, No. 3, May-June 2000, pp. 476-482, doi:10.2514/2.4553.
- Vasile, M., Bernelli-Zazzera, F.: 2003, 'Targeting a Heliocentric Orbit Combining Low-Thrust Propulsion and Gravity Assist Manoeuvres', **Operational Research in Space & Air**, Vol. 79, ISBN 1-4020-1218-7, Book Series in Applied Optimization, Kluwer Academy Press.
- Vasile, M. Bernelli-Zazzera, F.: 2003, 'Optimizing Low-Thrust and Gravity Assist Maneuvres to Design Interplanetary Trajectories', **The Journal of the Astronautical Sciences**, Vol. 51, No. 1, January-March 2003, pp. 13-35.
- Whiffen, G. J.: 2002, 'Static/Dynamic Control for Optimizing a Useful Objective', No. Patent 6496741, Dec. 2002.
- Whiffen, G. J. and Sims, J. A.: 2001, 'Application of a Novel Optimal Control Algorithm to Low-Thrust Trajectory Optimization', No. AAS 01-209, Feb. 2001.
- Whiffen, G. J. and Sims, J. A.: 2002, 'Application of the SDC Optimal Control Algorithm To Low-Thrust Escape and Capture Including Fourth-Body Effects', **2nd International Symposium on Low-Thrust Trajectories**, Toulouse, France, June 18-20, 2002.
- Yakowitz, S.: 1988, 'Theoretical and Computational Advances in Differential Dynamic Programming', **Control and Cybernetics**, Vol. 17, No. 2-3, pp. 173-189.
- Yakowitz, S. and Rutherford, B.: 1984, 'Computational Aspects of Discrete-Time Optimal Control', **Applied Mathematics and Computation**, Vol. 15, pp. 29-45.