Direct Lunar Descent Optimisation by Finite Elements in Time Approach

Massimiliano Vasile and Amalia Ercoli Finzi

Dipartimento di Ingengeria Aerospaziale, Politecnico di Milano, Milano, Italy

Keywords: **optimal control, finite elements in time, trajectory optimisation**

Abstract

In this paper a direct approach to trajectory optimisation, based on Finite Elements in Time (FET) discretisation is presented. Trajectory optimisation is performed combining the effectiveness and flexibility of Finite Elements in Time in solving complex boundary values problems with a common nonlinear programming algorithm. In order to avoid low accuracy proper to direct approaches, a mesh adaptivity strategy is implemented which exploits the ability of finite elements to represent both continuous and discontinuous functions. The effectiveness and accuracy of direct transcription by FET are proved by a selected number of sample problems. Finally an optimal landing manoeuvre is presented to show the power of the proposed approach in solving even complex and realistic problems.

1 Introduction

One of the most important aspects in space mission projects is the design of trajectories (e.g. transfer orbits, homing trajectory, rendezvous), under several constraint conditions arising from mission requirements. The optimisation of the above mentioned trajectories, with respect to one or more objectives, is of primary importance to the actual realisation of the mission. This requires the solution of complex and strongly nonlinear boundary value problems, which is commonly obtained by numerical means.

Most of numerical methods to trajectory optimisation or optimal control problem can be classified in two categories: direct methods and indirect methods [1]. The direct methods include those that transcribe the continuous problem to a finite-dimensional nonlinear programming problem (NLP) by some parameterisation of the control and state histories.

Indirect methods are based on finding the solution of a boundary-value problem that results from the first-order necessary conditions of optimal control in terms of the adjoint differential equations, the maximum principle, and associated boundary (transversality) conditions [2]. This translates into a nonlinear multipoint boundary value problem, whose solution is usually found by a root-finding algorithm. Generally indirect methods converge rapidly in the immediate neighbourhood of the optimal solution providing an extremely accurate solution.

However there are three primary drawbacks to this approach in practice. First, it is necessary to derive analytic expressions for the necessary conditions, and for complicated nonlinear dynamics this can become quite daunting even using dedicated software. Second, the region of convergence for a root-finding algorithm may be

Contact author: *Massimiliano Vasile,*
vasile@aero.polimi.it
Politecnico di Milano
Dipartimento di Ingegneria Aerospaziale
via La Masa 34, 20158 Milano Italy

surprisingly small, especially when it is necessary to guess values for the adjoint variables that may not have an obvious physical interpretation. Third, for problems with path inequalities it is necessary to guess the sequence of constrained and unconstrained sub arcs (referred to as the switching structure) before iteration can begin.

On the other hand, a direct method requires neither an analytical expression for the necessary conditions nor an initial guess for the adjoint variables. All direct methods can be applied without explicit derivation of the necessary conditions, i.e., adjoint, transversality, maximum principle, and do not require any specification of the arc sequence for problems with path inequalities. Instead, the dynamic variables are adjusted to directly optimise the objective function, and at the end of the optimisation process it is possible to obtain a good estimation of the adjoint variables [3].

However direct approaches generally seam not to be able to achieve high accuracy on final solution, even after many iterations, especially for the controls, when discontinuities in the solution occur. In this case the output of a direct algorithm is quite noisy and should be refined by an indirect method [4].

Anyway it is sometimes desirable to have a unique general tool which can converge easily to the solution with good accuracy without the need to derive specific equations for specific problems.

The idea, presented in this paper, is to exploit the high flexibility of Finite Elements in Time (FET) in representing both continuous and discontinuous time histories to the robustness of a common NLP algorithm. Indeed using the appropriate representation of the controls and the states, during the transcription process, could lead to accurate solution even without a further refinement by an indirect approach.

FET have been already used to solve complex problem proving their effectiveness and robustness compared to shooting or multiple shooting techniques [5],[6],[7]. Anyway so far, all FET approaches to optimal control problem can be classified as indirect methods and still suffer from most of the major drawbacks of general indirect methods. In particular the need to explicitly derive transversality and adjoint equations and the necessity to identify the switching structure previously to begin the optimisation process.

In this paper the general development of direct transcription by FET discretization is derived and applied to some meaningful sample problems of different nature with boundary and path constraints.

2 Trajectory Optimisation Problem

A general trajectory optimisation problem (or optimal control problem) consists of finding a control vector **u***(t)* and the final time t_f, that minimise the performance index:

$$J[\mathbf{x}(t),\mathbf{u}(t),t_f] = \phi(\mathbf{x}(t),\mathbf{u}(t),t)\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f} L(\mathbf{x}(t),\mathbf{u}(t),t)dt \qquad (1)$$

subject to a system of *n* non-linear differential equations:

$$\dot{x}_i(t) = F_i(\mathbf{x}(t),\mathbf{u}(t),t), \quad i=1,...,n, \quad t_0 \le t \le t_f \qquad (2)$$

2n boundary conditions:

$$\psi_i(\mathbf{x}(t_0),\mathbf{x}(t_f),t_f) = 0, \quad i=1,...,k \le 2n \qquad (3)$$

and *m* inequality constraints on the controls and states of the form:

$$G_i(\mathbf{x}(t),\mathbf{u}(t),t) \ge 0, \quad i=1,...,m, \quad t_0 \le t \le t_f \qquad (4)$$

The vector of control variables is denoted by $\mathbf{u}(t)=(u_1(t),...,u_l(t))^T$ and the vector of state variables is denoted by $\mathbf{x}(t)=(x_1(t),...,x_n(t))^T$. The functions $J: \Re_{n+1} \to \Re$, $\mathbf{F}$: $\Re_{n+l+1} \to \Re_n$, ψ: $\Re_{2n+1} \to \Re_k$, and $\mathbf{G}$: $\Re_{n+l+1} \to \Re_m$ are assumed to be continuously differentiable. The controls

$u_i:[t_0,t_f] \rightarrow \Re_{i=1,\dots,l}$, are assumed to be bounded and measurable and t_f may be fixed or free.

3 Direct Transcription by FET

In this chapter general development of the direct transcription method by FET is presented. Algorithms based on indirect transcription by finite elements in time discretisation have been already demonstrated to be really effective in solving even difficult problems tanks to their good convergence and accuracy properties [5],[6]. Never the less they still suffer from most of the major drawbacks of indirect methods.

In fact, they need to introduce a sequence of arcs along which state or control constraint shall be alternately active or inactive and to provide a first guess for it [7]. Furthermore they need to explicitly derive the necessary conditions, which can be quite cumbersome for realistic problems. Thus a direct approach should be welcome for several difficult applications where an initial guess of the solution is required at a relative low cost. Starting from this solution a further refinement can be obtained either by indirect transcription, or by the same direct method using a mesh refinement strategy [8].

The basic idea is to transform the optimal control problem into a general non-linear programming problem (NLP) of the form:

$$\min \quad J(\mathbf{y}) \tag{5}$$

subject to

$$\begin{aligned} &\mathbf{c}(\mathbf{y}) \geq 0 \\ &\mathbf{b}_l \leq \mathbf{y} \leq \mathbf{b}_u \end{aligned} \tag{6}$$

where, $\mathbf{y}$ is the vector of NLP variables, J($\mathbf{y}$) the objective function to be minimised, $\mathbf{c}(\mathbf{y})$ a vector of non-linear constraints and $\mathbf{b}_l$ and $\mathbf{b}_u$ respectively lower and upper bounds on NLP variables.

The first step to transcribe optimal control problems by FET, consist of writing differential equations into weighted residual form considering boundary conditions of the weak type (or natural type):

$$\int_{t_0}^{t_f} \delta\mathbf{w}^T (\dot{\mathbf{x}} - \mathbf{F}) dt = \delta\mathbf{w}^T (\mathbf{x} - \mathbf{x}^b)\Big|_{t_0}^{t_f} \tag{7}$$

where $\delta\mathbf{w}(t)$ are generalised weight (or test) functions. Integrating by parts $\delta\mathbf{w}^T\dot{\mathbf{x}}$, one obtains:

$$\int_{t_0}^{t_f} \left\{ \delta\dot{\mathbf{w}}^T \mathbf{x} + \delta\mathbf{w}^T \mathbf{F} \right\} dt - \delta\mathbf{w}_f^T \mathbf{x}_f^b + \delta\mathbf{w}_0^T \mathbf{x}_0^b = 0 \tag{8}$$

Variational equation (8) must be satisfied along with algebraic and boundary constraints:

$$\mathbf{G}(\mathbf{x},\mathbf{u},t) \geq 0 \tag{9}$$

$$\psi(\mathbf{x}_0^b, \mathbf{x}_f^b, t)\Big|_{t_0}^{t_f} = 0 \tag{10}$$

Now let the time domain $D(t_0,t_f) \subset \Re$ be decomposed into N finite time elements:

$$D = \bigcup_{j=1}^{N} D_j(t_{j-1}, t_j) \tag{11}$$

On each time element D_j, states, controls $[\mathbf{x},\mathbf{u}]$ and test functions $\delta\mathbf{w}$ are parameterised as follows:

$$\begin{Bmatrix} \mathbf{x} \\ \mathbf{u} \end{Bmatrix} = \sum_{s=1}^{p} f_s(t) \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{u}_s \end{Bmatrix} \tag{12}$$

$$\delta\mathbf{w} = \sum_{k=1}^{p+1} g_k(t) \delta\mathbf{w}_k \tag{13}$$

The quantities $\boldsymbol{x}_s$, and $\boldsymbol{u}_s$ are called internal node values and the basis functions f_s and g_k are chosen within the space of polynomials of order $p-1$ and p respectively:

$$f_s \in P^{p-1}(D_j); \; g_k \in P^p(D_j) \tag{14}$$

Notice that generally the order *p* of the polynomials can be different for states and controls.

In a more general way the domain *D* could be decomposed as a union of smooth images of the reference time interval [-1,1] where a reference parameter τ is defined as*:*

$$\tau = 2\frac{t - t_{j-1/2}}{t_j - t_{j-1}} = 2\frac{t - t_{j-1/2}}{\Delta t_j} \qquad (15)$$

Polynomials f_s and g_k can be constructed in several ways. One possible choice is to use Lagrangian interpolants associated with internal Gauss-type nodes. Generally speaking if $\{\xi_s\}^p_{s=1}$ are the set of Gauss points on the reference interval [-1,1], $f_s(\tau)$ will be the Lagrangian interpolating polynomial vanishing at all Gauss points except at ξ_s where it equals one.

Each integral of the continuous forms (1) and (8), is then replaced, for each element, by a q-points Gauss quadrature sum. Therefore the objective function (1) becomes a sum of *N* Gauss quadrature formulas:

$$\begin{aligned} J &= \phi(\mathbf{x}_0^b, \mathbf{x}_f^b, t_f) + \\ &\sum_{j=1}^{N} \sum_{i=1}^{q} \sigma_i L[\mathbf{x}(\tau_i), \mathbf{u}(\tau_i), \tau_i] \frac{\Delta t_j}{2} \end{aligned} \qquad (16)$$

while integral (8) is split into *N* integrals of the form:

$$\begin{aligned} &\sum_{k=1}^{p+1} \sum_{i=1}^{q} \sigma_i \left[\delta\dot{\mathbf{w}}_k(\tau_i)^T \mathbf{x}(\tau_i) + \delta\mathbf{w}_k(\tau_i)^T \mathbf{F}(\tau_i) \frac{\Delta t_j}{2} \right] - \qquad j=1,\dots,N \\ &\delta\mathbf{w}^T_{p+1}\mathbf{x}^b_j + \delta\mathbf{w}^T_1 \mathbf{x}^b_{j-1} = 0 \end{aligned} \qquad (17)$$

where σ_i are Gauss weights.
Here and in the following we introduce the notation:

$$\begin{aligned} \mathbf{x}(\tau_i) &= \sum_{s=1}^{p} \mathbf{x}_s f_s(\tau_i) \\ \delta\mathbf{w}_k(\tau_i) &= \delta\mathbf{w}_k g_k(\tau_i) \end{aligned} \qquad (18)$$

$$\begin{aligned} L(\tau_i) &= L(\mathbf{x}_s f_s(\tau_i), \mathbf{u}_s f_s(\tau_i), \tau_i); \\ \mathbf{F}(\tau_i) &= \mathbf{F}(\mathbf{x}_s f_s(\tau_i), \mathbf{u}_s f_s(\tau_i), \tau_i) \end{aligned} \qquad (19)$$

Parameters $\mathbf{x}^b_{j-1}$ and $\mathbf{x}^b_j$ are boundary values at the beginning and at the end of each element.

Different choices of internal Gauss points could lead to different FET algorithms: if Gauss-Lobatto points are chosen both for generating test functions and for the numerical quadrature rule (*q=p+1*), there is no need of an actual integration, but the function **F** can be simply collocated at integration points. Another possibility is to use Gauss-Legendre points to generate control functions and Gauss-Lobatto points to generate states and weight functions. Numerical quadrature of the integral equation (8) is then performed by Gauss-Legendre rule (*q=p*), while integral (16) can be developed either by Gauss-Lobatto or Gauss-Legendre rule. This second choice has given a particularly good result, especially when the control function is non-linear, as can be seen in examples 1 and 3. Whatever f_s and g_k are generated, the linear part of equation (17) can be always integrated only once before the optimisation process begins.

Now equation (24) must be satisfied for every arbitrary value of virtual quantity $\delta\mathbf{w}_k$, as a consequence each element equation (17) is developed into *p+1* equations:

$$\begin{aligned} &\sum_{i=1}^{q} \sigma_i \left(g_1(\tau_i)\mathbf{F}(\tau_i)\frac{\Delta t_j}{2} + \dot{g}_1(\tau_i)\mathbf{x}(\tau_i)\right) + \mathbf{x}^b_{j-1} = 0 \\ &\sum_{i=1}^{q} \sigma_i \left(g_k(\tau_i)\mathbf{F}(\tau_i)\frac{\Delta t_j}{2} + \dot{g}_k(\tau_i)\mathbf{x}(\tau_i)\right) = 0 \qquad k=2,\dots,p \\ &\sum_{i=1}^{q} \sigma_i \left(g_{p+1}(\tau_i)\mathbf{F}(\tau_i)\frac{\Delta t_j}{2} + \dot{g}_{p+1}(\tau_i)\mathbf{x}(\tau_i)\right) - \mathbf{x}^b_j = 0 \end{aligned} \qquad (20)$$

System of equations (20), is written for each element. All the elements are then assembled matching the final boundary node to the initial one of the next element. For continuous solution, in order to preserve the continuity of the states, at matching points, the following condition must hold:

$$x^b_j = x^b_{j+1} \qquad j=1,\dots,N\text{-}2 \qquad (21)$$

Thus all the boundary quantities (21) cancel one another except for those at the initial and final times.

Algebraic constraint equation (4) can be directly collocated at Gauss nodal points:

$$\mathbf{G}\ (\mathbf{x}(\xi_s),\mathbf{u}\ (\xi_s),\xi_s) \geq 0 \qquad (22)$$

The N*(p+1)*n algebraic equations (27) along with system (22) represent the **c**(**y**) constraint of the nonlinear problem while **y**=[**x**$_s$,**u**$_s$,**x**$^b_0$,**x**^b_f,t_0,t_f] the NLP variables.

Notice that the present formulation is bi-discontinuous because continuity at boundaries of each element is only weakly enforced. This means that, generally, there is a gap between the internal nodes, at the boundary nodes. This allows control, for which no continuity requirement is imposed, to be discontinuous at boundaries, providing the possibility to better fit discontinuous time histories.

However, for continuous solution, it can be demonstrated numerically that, using the above development for state and controls, the gap between internal nodes and boundary nodes tends to zero as the accuracy increases. This can be clearly seen in example 1.

3.1 Mesh Refinement

The solution of the above mentioned NLP problem provides a good estimation of the states, the control and the adjoint variables (an extensive discussion on necessary first order optimality conditions of the discretised problem can be found in [8]). A further refinement can be achieved either by an indirect method or adjusting appropriately the mesh grid. Thus, in order to achieve an accurate solution, a mesh refinement algorithm has been implemented. The algorithm consists of five different strategies thought to fit at best each element on each sub arc: p adaptivity, h adaptivity, smoothing, stretching and dynamic tuning.

p adaptivity: for each element the order of the polynomials is increased if the gradient of the solution, found after each macro-iteration, is grater than a given value. Defining on each element the gradient as:

$$\Delta^j_s = \frac{2}{\Delta t_j}\frac{x_{s+1}-x_s}{\zeta_{s+1}-\zeta_s} \quad s = 2,\dots,p \quad j = 1,\dots,N \qquad (23)$$

the number of nodes is increased according to:

$$\forall j : \Delta^j_s \geq \Delta_{\max} \Rightarrow p = p+1 \qquad (24)$$

h adaptivity: if p reaches a limit value p$_{\max}$, the element is split into two lower order elements. Thus defining the splitting point as:

$$\tau_{sp} = \frac{\zeta_{s+1}-\zeta_s}{2} \qquad (25)$$

the element split takes place where the gradient reaches its maximum:

$$sp \in \left\{ s : \Delta^j_s = \max \Delta^j_s \right\} \qquad (26)$$

smoothing: the order of the element is lowered in two easily definable cases:

-if the gradient is smaller than a given value:

$$\forall j : \Delta^j_s \leq \Delta_{\min} \Rightarrow p = p-1 \qquad (27)$$

-if, for *p>2*, after two macro iterations, the gradient remains unchanged and:

$$\Delta^j_{s+1} = \Delta^j_s \qquad (28)$$

stretching: after every macro iteration the length of each element is changed according to the following rule:

$$\begin{aligned}&\forall j : \Delta_s^j \geq \bar{\Delta} \Rightarrow \Delta t_j = \alpha \Delta t_j \\ &\text{with} \quad \alpha \in (0,1]\end{aligned} \tag{29}$$

with, of course, the constraint that:

$$\sum_{j=1}^{N} \Delta t_j = t_f - t_0 \tag{30}$$

dynamic tuning: in this case N different Δt_j are introduced directly into the optimisation process, as NLP variables, along with constraint 30.

The first four strategies will be generally referred to as static refinement.

The gradient (23) is computed both for states and for controls, anyway in many cases the leading parameter, taken into consideration for mesh adaptivity, is just the gradient of the control or of a function of control components.

4 Examples and Numerical Results

In this section the approach is validated by two sample problems characterised by typical difficulties that can be encountered in space mission design. The former demonstrate the accuracy of the method in solving problems characterised by continuous solutions, the latter is to demonstrate the effectiveness of FET discretisation and mesh adaptivity when discontinuous control function are present.

Finally a realistic case, characterised by a strongly non-linear force function quite difficult to be treated by an indirect approach, is addressed to confirm the power of the method in solving even complex and realistic problems.

All the results reported in this section have been obtained implementing the direct trajectory optimisation method, mentioned in the previous sections, in a FORTRAN code called DEMON Toolbox (Direct finite Elements Multiobjective Optimisation Toolbox).

4.1 Minimum Time to Orbit

The first example (taken from reference 2 section 2.7) is a takeoff problem with constant gravity acceleration. Denoting with *x* and *z* the position of a particle at a given time and with *u* and *v* its velocity, the objective is to minimise the time to transfer the particle to a rectilinear path at an altitude *h*, from the surface of the Moon, with an horizontal velocity *U*. In addition the final vertical velocity is constrained to be zero.

The objective function is

$$\min \ t_f \tag{31}$$

and the state equations are defined as

$$\begin{aligned}\dot{x} &= u \\ \dot{z} &= v \\ \dot{u} &= a\cos\beta \\ \dot{v} &= -g + a\sin\beta\end{aligned} \tag{32}$$

with boundary conditions at initial and final time:

$$\psi \Big|_{t_0} = \left.\begin{Bmatrix} x \\ z \\ u \\ v \end{Bmatrix}\right|_{t_0} = 0 \qquad \psi \Big|^{t_f} = \left.\begin{Bmatrix} u-U \\ z-h \\ v \end{Bmatrix}\right|^{t_f} = 0 \tag{33}$$

The acceleration *g* is constant and equal to the gravity acceleration on lunar surface: g=0.0016km/s^2. The thrust angle β is the control variable while the mass is constant and the acceleration *a* is equal to *a*=0.0049 km/s^2. The required final horizontal velocity is U=1.6559 km/s at an altitude h of 50 km over the surface of the Moon.

As stated before in this example state polynomials are generated on Gauss-Lobatto points while control polynomials are generated on Gauss-Legendre points. In Fig. 1,2 and 3 the results for internal nodes for three different mesh distributions are reported: two elements with polynomials of the first order (two internal nodes for each element) referred to as FET1 in table 1, four elements with polynomials of the first order, referred to as FET2, and four elements with polynomials of the second order (three internal nodes for each element), referred to as FET3.

For the two elements solution is quite evident the difference between the internal nodes and the analytical solution. On the other hand, already using four polynomials of the first order internal nodes well fit the analytical solution for the states. Since the final solution is a smooth function of time, a slight increase in the order of polynomials provides with an extremely accurate solution. In this case, internal nodes lie exactly on the analytical solution and no gap at boundaries can be seen. It should be noticed that, thanks to the proper choice of Gauss nodes, controls fit perfectly the analytical solution already for the two elements solution.

	ANALYTICAL	FET1 2:2	FET 2 2:2:2:2	FET 3 3:3:3:3
β_0	43.632°	43.323°	43.644°	43.638°
β_f	-9.700°	-9.976°	-9.655°	-9.691°
$t_f(s)$	373.182	373.171	373.179	373.18

Table 1. Comparison between analytical and numerical results

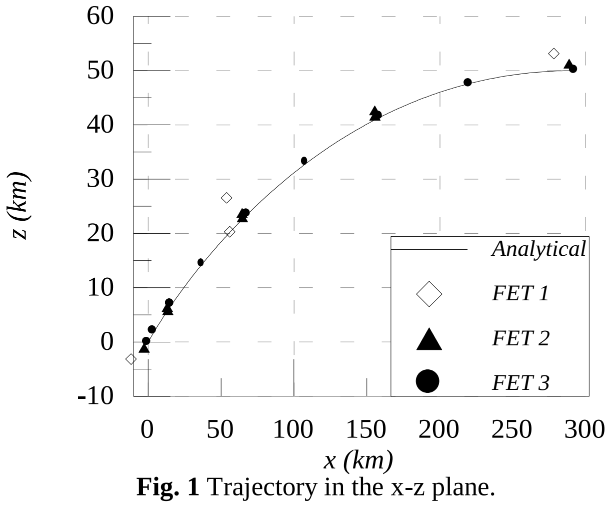

Fig. 1 Trajectory in the x-z plane.

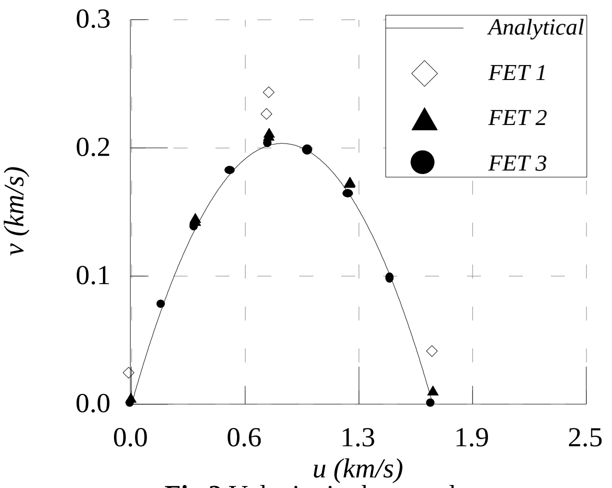

Fig.2 Velocity in the u-v plane

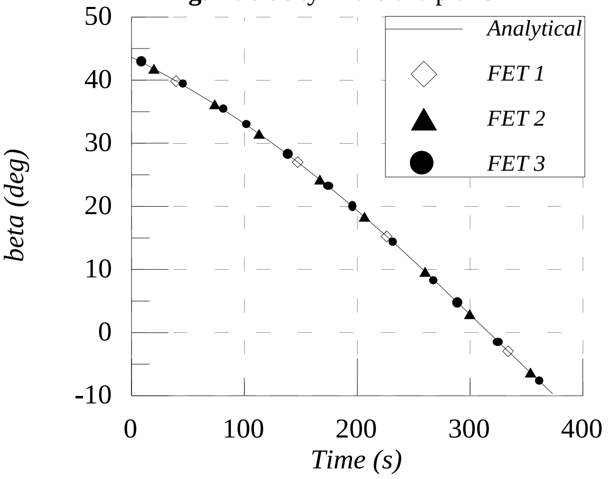

Fig. 3 Control law vs. time.

4.2 Minimum thrust attitude manoeuvre with bounded control (bang-zero-bang control)

The second example (taken from reference 2 section 3.9) consists of a minimum time attitude manoeuvre. Denoting with θ the attitude of a satellite at a given time and with ω its angular velocity, the objective is to minimise the thrust to rotate the satellite from initial state $\theta(t_0)=1$ and $\omega(t_0)=0$ to final state $\theta(t_f)=0$, $\omega(t_f)=0$ in a given time t_f=3.

The objective function and the state equations are defined as

$$\min \; J = \int_0^{t_f} \|u\| dt \qquad (34)$$

$$\begin{aligned}\dot{\theta} &= \omega \\ \dot{\omega} &= u\end{aligned} \tag{35}$$

with boundary conditions at initial and final time:

$$\psi\,\Big|_{t_0} = \left.\begin{Bmatrix}\theta - 1 \\ \omega\end{Bmatrix}\right|_{t_0=0} = 0 \quad \psi\,\Big|^{t_f} = \left.\begin{Bmatrix}\theta \\ \omega\end{Bmatrix}\right|^{t_f} = 0 \tag{36}$$

An additional inequality constraint is imposed on the control variable *u*:

$$\|u\| \le 1 \tag{37}$$

Results for state are reported in Fig. 4 and 5 where internal nodes are represented by a dotted line while analytical solution is represented by a solid line.

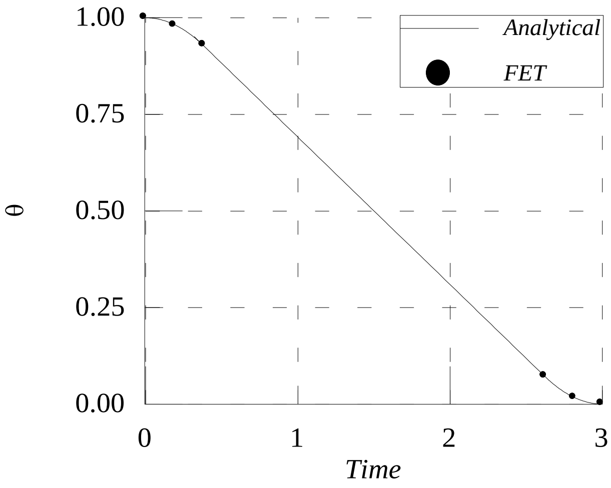

Fig. 4 Attitude angle θ in radiant vs. time.

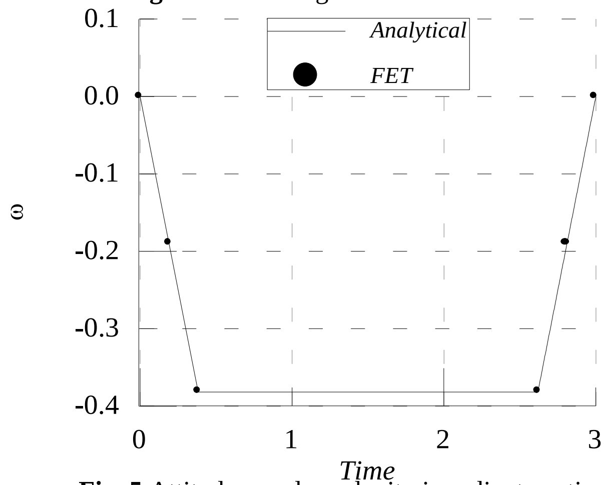

Fig. 5 Attitude angular velocity in radiant vs. time.

Lobatto points have been used both to generate states and controls polynomials and for quadrature formulas.

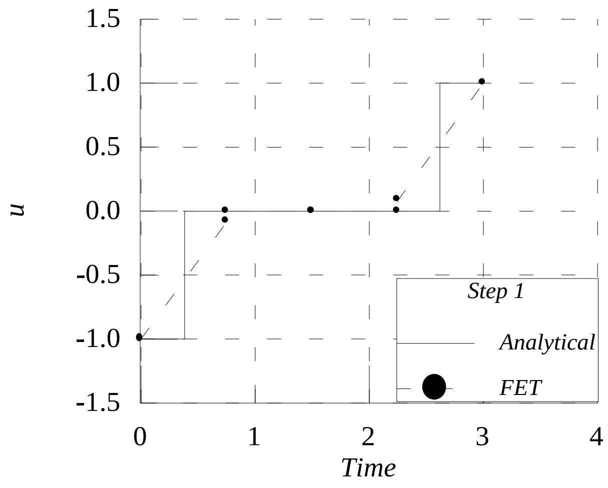

Fig. 6 Control law: initial mesh.

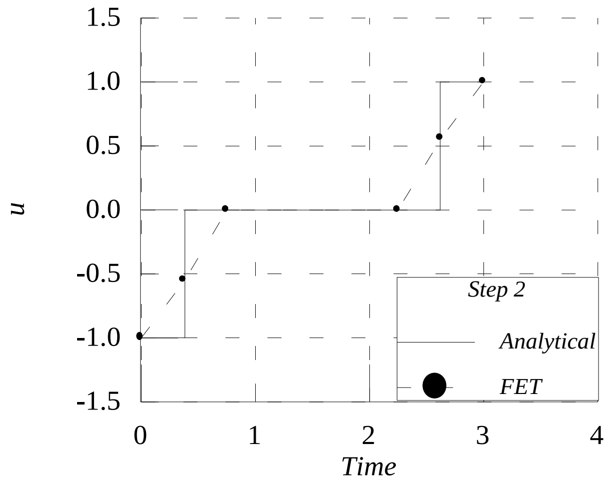

Fig..7. Control law: smoothing and first *p* refinement.

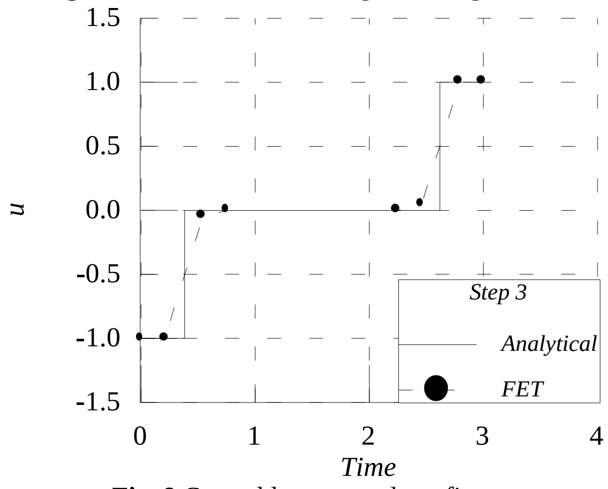

Fig. 8 Control law: second *p* refinement.

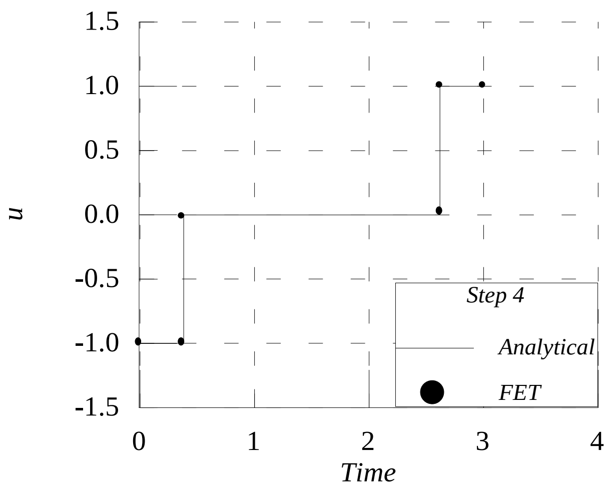

Fig. 9 Control law: final mesh.

	ANALYTICAL	FET STATIC	FET DYNAMIC
t_1	0.381966	0.377	0.3819
t_2	2.618033	2.619	2.6180

Table 2. Switching structure

In this example, the first solution labelled step 1 in Fig.6, has been obtained using a uniform distribution of 4 elements with polynomials of the first order both for the states and for the controls. The leading parameter for mesh refinement is the gradient of the control. As a second step, *p* adaptivity and smoothing have been applied. The two central elements are merged into one single element because the gradient of the solution is almost zero over the whole interval. At the same time the order of the other two elements is increased to 2 in order to overcome the discontinuity in the control function (see Fig.7). Anyway a further increase, up to 3, of the order does not lead to a meaningful improvement (see Fig.8). Therefore at the next iteration, the solution for the control is split into three arcs, one element each arc, and the order of the polynomials is reduced to one on each arc (see Fig.9). A very accurate result has been obtained applying the dynamic strategy directly after step 1. Results for the switching structure are reported in table 2, both for static and dynamic strategies, and compared to the analytical solution.

It should be noted that, due to nature of the solution, control could be represented by three constant functions, while the state requires (especially the position, which is a quadratic function of time) higher order polynomials.

4.3 Constrained Optimal Landing

Several future missions for lunar exploration aim to perform a soft landing on the South Pole of the Moon. The objective of this analysis is to design an optimal landing trajectory, which minimises the initial mass of the spacecraft under several constraint conditions of different nature. The landing manoeuvre is divided in two phases: a coast phase and a homing phase.

During the first one the spacecraft is transferred from a circular parking orbit into an elliptical orbit with the periselenium over the South Pole. From this coasting orbit, homing phase will begin firing the main engine while approaching the South Pole from the far side to the near side of the Moon (see Fig. 10). Here only the homing phase will be analysed while the coasting trajectory is supposed to be known and represents a constraint for the homing trajectory [9],[[10]. The spacecraft is subject to the gravity force of the Moon and a main engine, with limited thrust, controls the descent (see Fig.12).

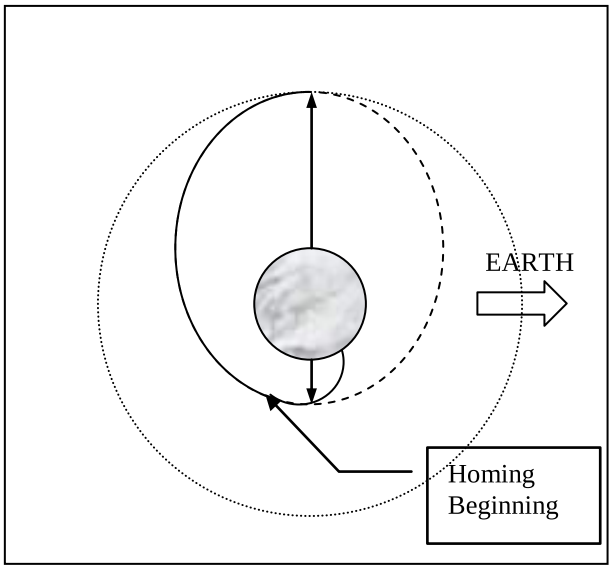

Fig. 10 Coasting orbit geometry

A linear mass flow equation is introduced to take into account mass variation. The resulting system of differential equations which governs the dynamics of the spacecraft is defined as follows:

$$\dot{\mathbf{s}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \nabla U + \frac{\mathbf{u}}{1+m_R} \qquad (38)$$

$$\dot{m}_R = -\frac{u}{I_{sp} g_0} \qquad (39)$$

$$u = \sqrt{u_x^2 + u_y^2 + u_z^2} \le T_{\max} \qquad (40)$$

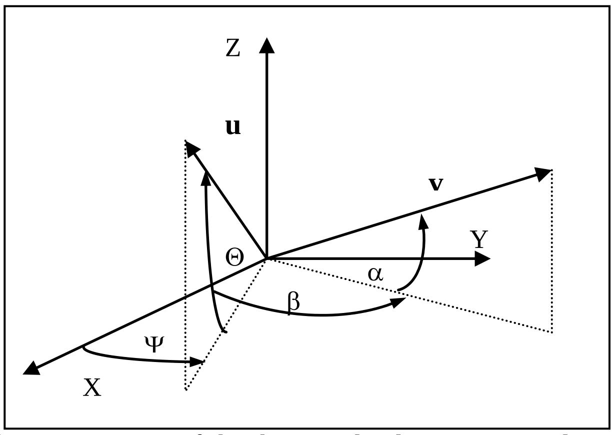

Fig. 11 Direction of the thrust and velocity vector relative to the inertial reference system.

where the state and the control vectors are defined as follows:

$$\mathbf{x} = \{s_x, s_y, s_z, v_x, v_y, v_z, m_R\}^T \qquad (41)$$

$$\mathbf{u} = \{u_x, u_y, u_z\}^T \qquad (42)$$

The mass ratio m_R is defined as the ratio between the propellant mass and the dry mass of the spacecraft, while the thrust T_{max} is the ratio between the actual maximum thrust of the engine and the dry mass of the spacecraft. I_{sp} is the specific impulse of the engine and g_0 the gravity constant on Earth surface.

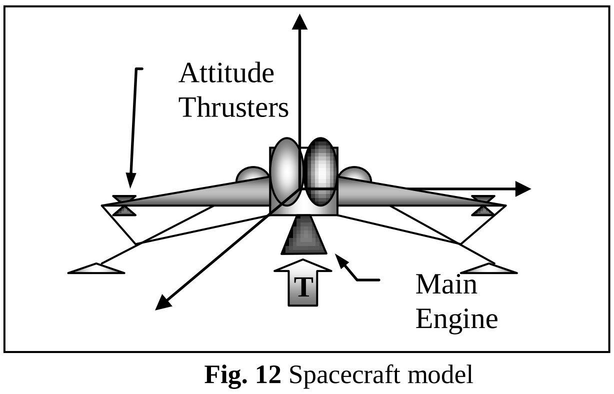

Fig. 12 Spacecraft model

Starting from the results for the control vector **u**, the inclinations, Θ and Ψ, of the thrust vector relative respectively to the x-y and to the z-x plane (see Fig. 11) have been computed.

The function U is the potential due to the gravity forces acting on the spacecraft, namely the gravity field of the Moon. In a selenocentric reference frame, the potential of the lunar gravity field is a sum of the potential of a sphere and the perturbation accounting for all the deviations of a real body from a sphere [11]:

$$U(\mathbf{s}) = \frac{\mu_M}{r} + R_M(\mathbf{s}) \qquad (43)$$

The perturbing function R_M is given as an expansion into spherical harmonics:

$$\begin{aligned} R_M(\mathbf{s}) &= \frac{\mu_M}{r} \sum_{l=2}^{+\infty} \left(\frac{r_M}{r}\right)^l \sum_{m=0}^{l} P_{lm}\left(\frac{s_Z}{r}\right) \\ &[\ C_{lm} \cos\left(m \operatorname{atan}\left(\frac{s_y}{s_x}\right) - m\theta\right) + \\ &S_{lm} \sin\left(m \operatorname{atan}\left(\frac{s_y}{s_x}\right) - m\theta\right)\ ] \end{aligned} \qquad (44)$$

where μ_M is the gravity parameter of the Moon, r_M is the mean equatorial radius and θ is the phase of the lunar rotation, namely the angle between some body fixed direction along the equator X_f and some inertial direction along the equator X (see Fig. 13). Parameters C_{lm} and S_{lm} are spherical harmonics coefficients while P_{lm} are Legendre spherical polynomials.

In this paper the perturbing function is developed up to order and degree 2, which is of course unrealistic to describe perturbations due to the gravity field of the Moon [12], but is enough to demonstrate the effectiveness of the optimisation approach in solving problems with highly nonlinear force functions. However, it has been demonstrated, in previews works [9], that for short arcs the perturbing function (44) does not produce relevant changes in the final optimal solution.

Notice that developing the analytical Jacobian of function (44) is quite cumbersome even using a dedicated software, thus in this case an indirect approach is less attractive than a direct one, especially for a preliminary study.

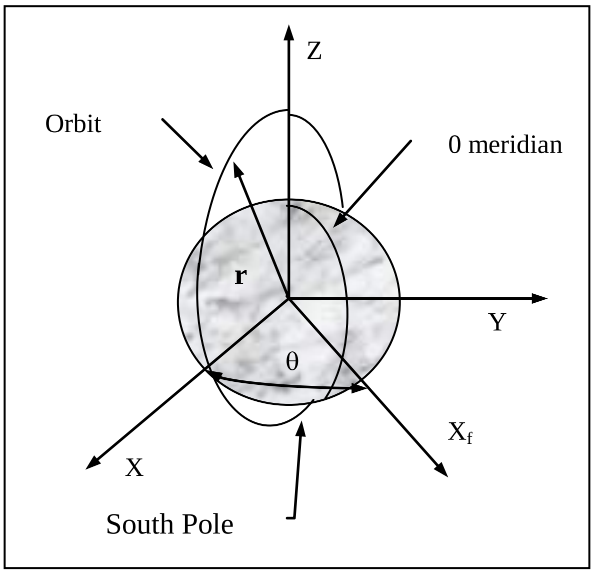

Fig. 13 Selenocentric and inertial reference frame

The initial point is constrained to lie on a given coasting orbit, thus initial position and velocity have to satisfy the following set of nonlinear algebraic equations, functions of the initial state vector and of the orbital parameters of the desired coasting orbit:

$$\psi(\mathbf{x},t)\ \Big|_{t_0} = \left\{ \begin{array}{l} v^2 + \frac{\mu_M}{a} - \frac{2\mu_M}{r} \\ r^2 - [a(1-e^2) - e(\chi\cos\omega + \zeta\sin\omega)]^2 \\ (s_y v_z - v_y s_z) - \sqrt{\mu_M a(1-e^2)}\sin i\cos\Omega \\ (s_x v_z - v_x s_z) + \sqrt{\mu_M a(1-e^2)}\sin i\sin\Omega \\ (s_x v_y - v_x s_y) - \sqrt{\mu_M a(1-e^2)}\cos i \end{array} \right\}\Bigg|_{t_0} = 0 \qquad (45)$$

where:

$$\chi = s_x\cos\Omega + s_y\sin\Omega$$
$$\zeta = \frac{(-s_x\sin\Omega + s_y\cos\Omega) + s_z}{\cos i + \sin i} \qquad (46)$$

Orbital parameters, for the coasting, orbit are summarised in table 3.

Final conditions on the states require having zero velocity at an altitude of 1 m over the ground. At this point the main engine is cut off. The landing site is located at 86° of latitude South and 0° of longitude and is slowly moving with the Moon. Therefore final condition are explicitly functions of time and of the Moon rotation velocity $\dot{\theta}$:

$$\psi(\mathbf{x},t)\ \Big|^{t_f} = \left\{ \begin{array}{l} r - 1738.001 \\ v_x \\ v_y \\ v_z \\ s_x - r_M\sin 86^\circ\cos\dot{\theta}t \\ s_y - r_M\sin 86^\circ\sin\dot{\theta}t \end{array} \right\}\Bigg|^{t_f} = 0 \qquad (47)$$

PARAMETER	VALUE
a	1798 km
e	0.0222
ω	270°
Ω	1°
i	90°

Table 3. Coasting orbit keplerian parameters

Final time t_f is free but bounded. At the beginning of the homing trajectory the performance index, J_1, is the initial mass ratio, i.e. the initial amount of propellant onboard:

$$\min\ J_1 = m_R(t_0 = 0) \qquad (48)$$

Below 10 km, at $t=t_3$, a new objective function, J_2, is introduced in order to maintain the velocity vector ground oriented. The new objective function is a combination of the previous merit function, and a weighted difference between the desired angle of descent and the actual one:

$$\min \quad J_2 = m_R(t_0) + \int_{t_3}^{t_f} w_v[\mathrm{abs}(v\cos 94^\circ - v_x) + \mathrm{abs}(v\sin 94^\circ - v_z)]dt \qquad (49)$$

Landing process is depicted in Fig. 14.

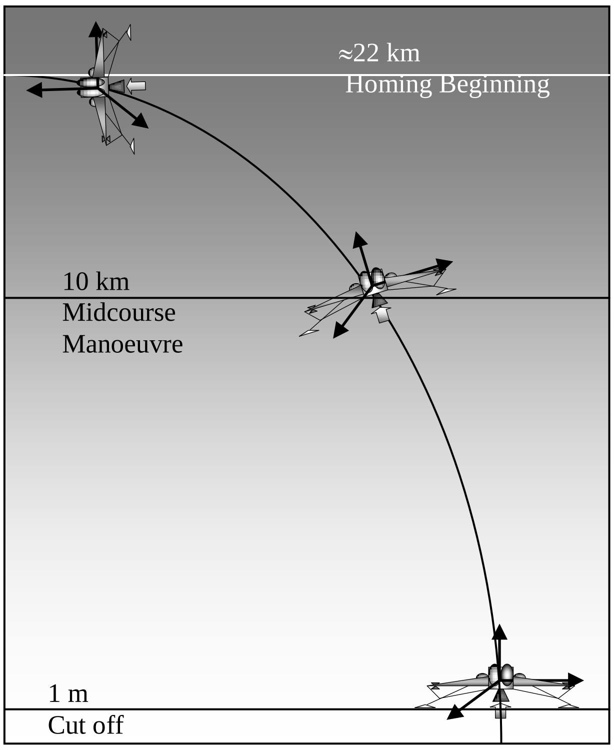

Fig. 14. Landing process.

The first solution has been computed using a uniform distribution of 8 elements with polynomials of the first order both for states and controls. After that all the static strategies have been applied in order to identify accurately the switching points. As a consequence the overall number of elements is risen up to 10 for the problem without manoeuvre and to 11 for the problem with manoeuvre, and the order for each element is risen up to 2.

Dynamic tuning has been applied both to the problem with and without the manoeuvre. However modified functional (49) makes the problem quite sensitive to mesh distribution and the present implementation of dynamic tuning has produced results less good than expected. Therefore only the output of static procedures has been reported here.

In Fig. 15 the history of the altitude is plotted against time both with and without the midcourse manoeuvre, while in Fig. 16 and 17 trajectories respectively in the x-z and x-y plane are reported. For states, solid line represents internal node solution while dotted line represents boundary node solution.

In Fig. 18,19 and 20 are reported the modulus of the velocity, its direction relative to the x-y plane and relative to the x-z plane respectively.

The effect of J_2 is quite evident in all the three plots.

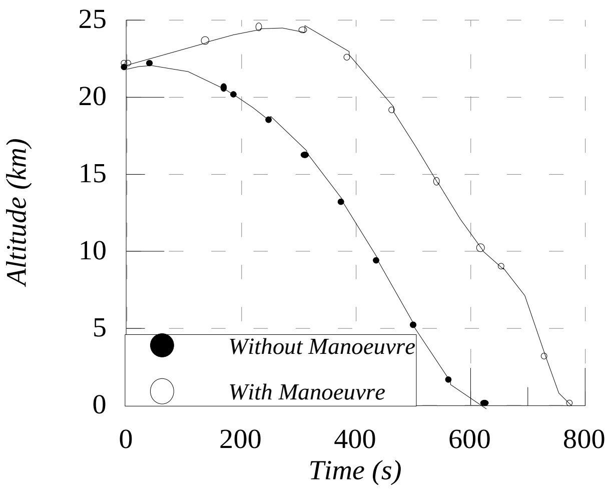

Fig. 15 The altitude vs. time.

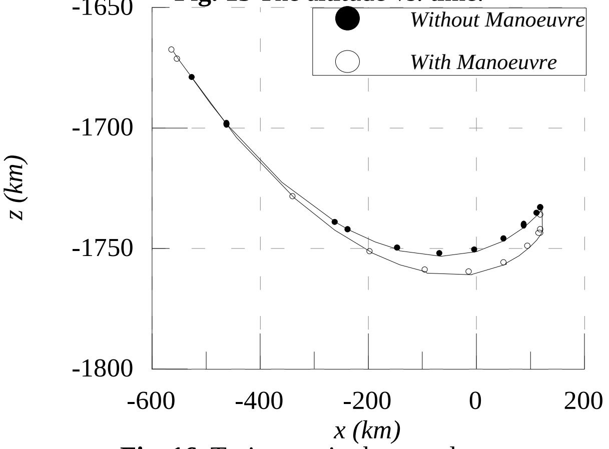

Fig. 16. Trajectory in the x-z plane.

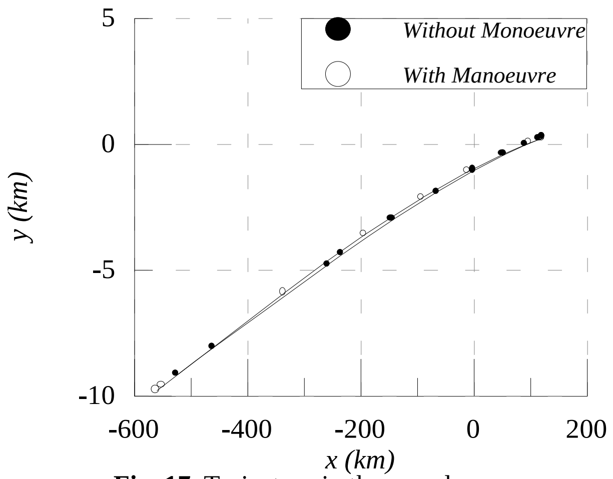

Fig. 17. Trajectory in the x-y plane.

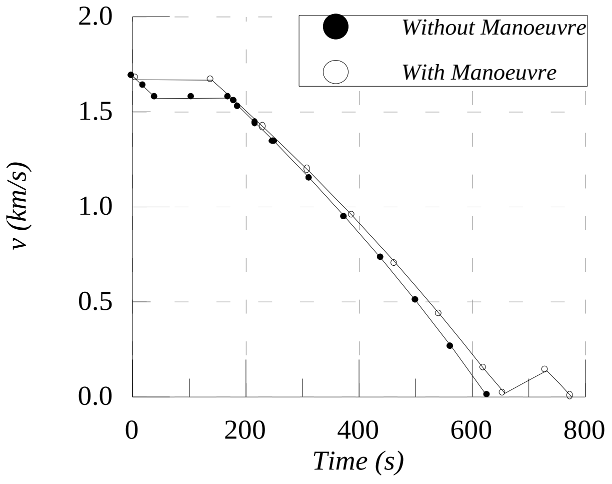

Fig. 18 Velocity Modulus vs. Time

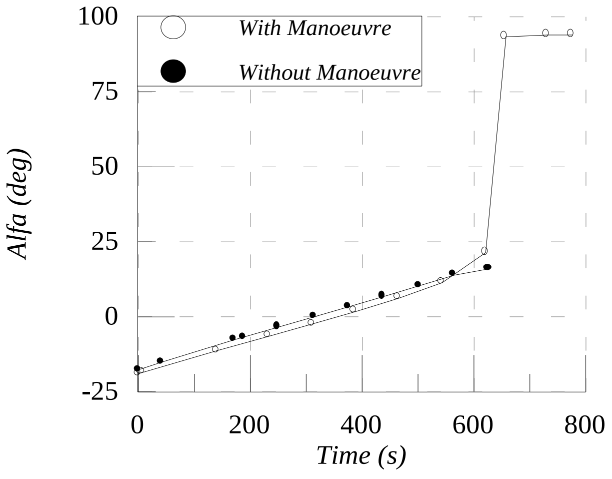

Fig. 19. Velocity direction relative to the x-y plane

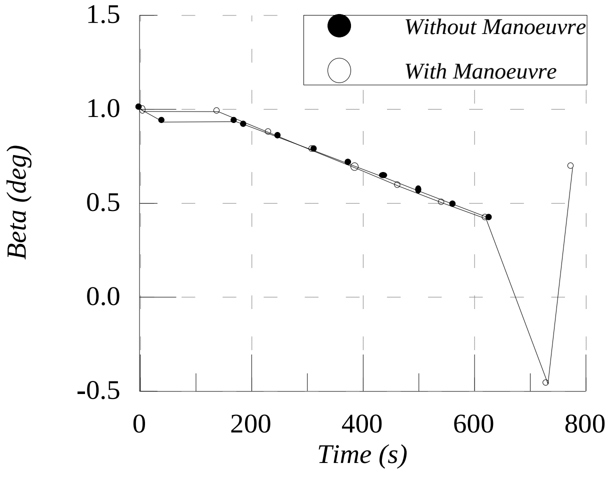

Fig. 20. Velocity direction relative to the x-z plane

The value chosen for parameter w_v is 0.02. A higher value produces a sharper manoeuvre while a smaller value yields a more gentle turn.

Thrust optimal program without and with manoeuvre is plotted respectively in Fig.21 and 22 , while thrust vector direction relative to the x-y and x-z plane is reported respectively in Fig. 23 and 24 (in this case only internal nodes are represented). Notice that, below 10 km, in accordance with the change in the velocity vector (see Fig.19), the thrust vector turns in the direction normal to lunar surface.

During the manoeuvre the engines are off (see Fig. 22) and there is a slight increase in the velocity modulus (see Fig.18). The switching structure for the problem with and without manoeuvre is presented in table 5 while results for the overall propellant consumption in both cases are summarised in table 4.

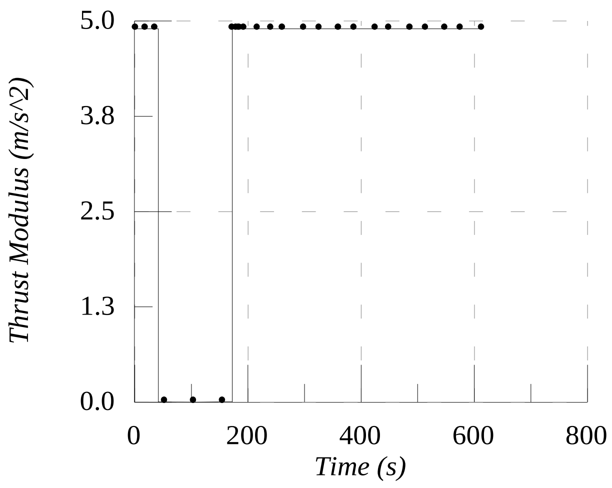

Fig. 21 Thrust modulus without manoeuvre.

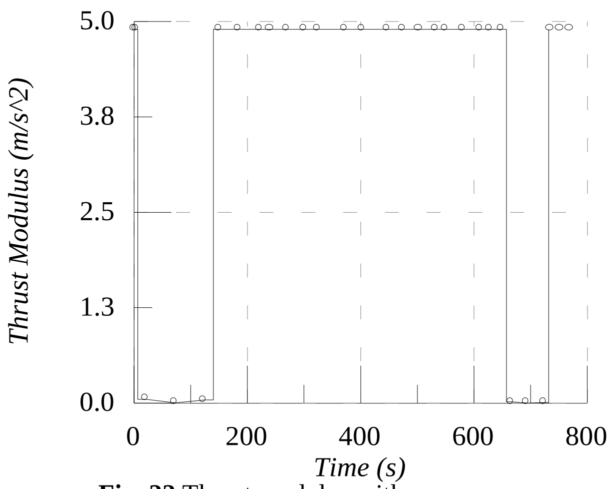

Fig. 22 Thrust modulus with manoeuvre

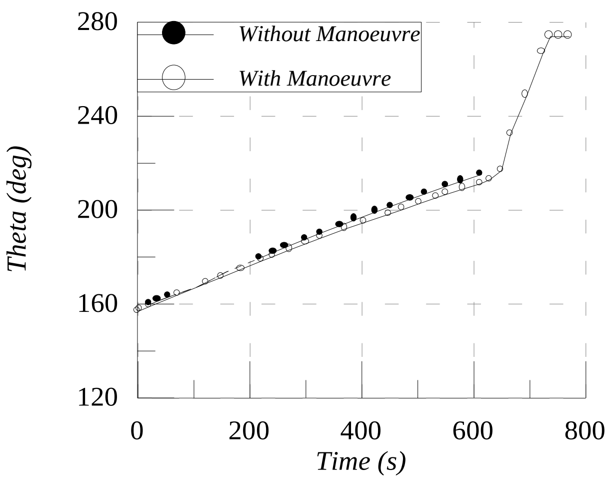

Fig.23 Thrust angle relative to the x-y plane

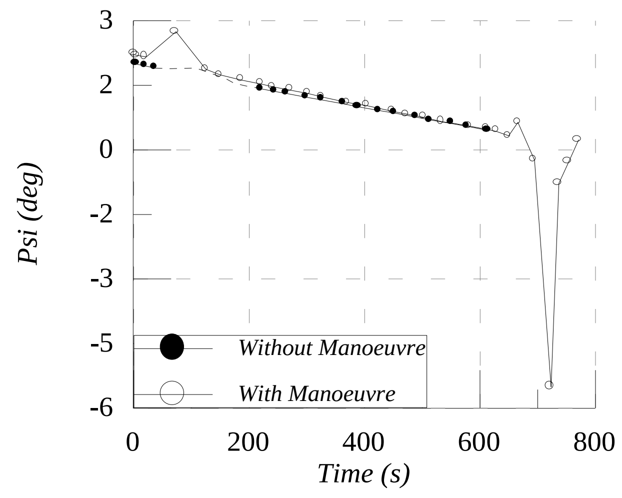

Fig. 24 Thrust angle relative to x-z plane

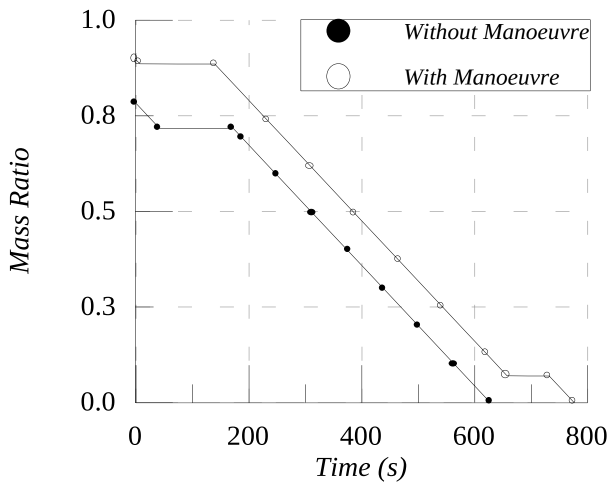

Fig..25 Mass Ratio vs. Time

	Without Manoeuvre	With Manoeuvre
Initial mass ratio	0.7829	0.8964
Final Time	627.447 s	775.806 s
T_{max}	4.9 m/s^2	4.9 m/s^2
I_{sp}	317 s	317 s

Table 4 Results for the landing manoeuvre

SWITCHING POINTS	WITHOUT MANOEUVRE	WITH MANOEUVRE
t_1	41.380 s	6.361 s
t_2	172.021 s	139.646 s
t_3	/	657.186 s
t_4	/	731.562 s

Table 5 Switching structure for the optimal landing problem

5 Conclusions

In this paper, a direct trajectory optimisation approach based on Finite Elements in Time has been employed to derive optimal trajectory and optimal control law to perform a soft landing on the South Pole of the Moon. The resulting NLP problem has been solved by a dense sequential quadratic programming algorithm. The novel approach exploits the ability of finite elements in representing both continuous and discontinuous functions. Discontinuities occur, especially on the controls, when constraints are alternately active and inactive along different arcs composing the solution. In this case the method is able to identify the switching structure yielding an accurate solution even without a further iteration by an indirect approach. To this end a mesh refinement strategy is implemented to better represent the solution with the right choice of elements for each sub-arc.

In addition, the present formulation provides a good estimation to the adjoint variables, which can be used, alternatively to the mesh adjustment, for further refinement by an indirect approach.

The effectiveness of the proposed technique has been proved by few selected problems presenting typical difficulties that can be encountered in space mission analysis and design.

Finally the optimal landing problem, related to future space missions aimed to the exploration of the Moon, has been afforded in order to confirm the effectiveness of the method even in difficult and realistic cases.

Although the reduced dimension of the NLP problem obtained by FET still allows the use of a dense SQP algorithm, at a relative low computational cost, a meaningful enhancement of the performances can be achieved exploiting the sparse structure of the Jacobian and Hessian matrixes. To this aim, at the moment, a sparse SQP algorithm is under development, whose benefits will be particularly welcome, in many huge space trajectory design problems that usually require a high computational cost.

References

[1] Betts J. T. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, Vol.21, No.2, pp.193-207, March-April 1998.

[2] Bryson A.E. Jr. and Yu-Chi H. *Applied Optimal Control,* Blaisdell Publishing Company, Waltham, Massachussetts, 1969.

[3] Von Stryk O. Numerical Solution of Optimal Control Problems by Direct Collocation. *Optimal Control Calculus of Variation, Optimal Control Theory and Numerical Methods*, 1993 Birkhauser Verlag.

[4] Von Stryk O., Schlemmer M. Optimal Control of Industrial Robot Manutec r3.*Computational Optimal Control, Internatioanl Series of Numerical Mathematics* 115 (Basel: Birkauser, 1994) 367-382.

[5] Hodges D. H., Bless R.R., Calise, A.J. and Leung M. Finite Element Method for Optimal Guidance of an Advanced Launch Vehicle. *Journal of Guidance, Control and Dynamics*, Vol. 15 No. 3, 1992,pp.664-671.

[6] Bless R. and Hodges D.H. Finite Element Solution of Optimal Control Problems with State-Control Inequality Constraints. *Journal of Guidance, Control, and Dynamics,* Vol. 15, No. 4, pp. 1029-1032, July-August 1992.

[7] Bless R., Hodges D. H., Seywald H. Finite Element Method for the Solution of State-Constrained Optimal Control Problems. *Journal of Guidance, Control, and Dynamics,* Vol. 18 No. 5, September-October 1995.

[8] Vasile M. Direct Trasription by FET for Optimal Space Trajectory Design. *Internal Report* DIA-SR 99-02, 1999.

[9] Vasile M., Floberghagen R. Optimal Trajectories for Lunar Landing Missions. NASA/CP-1998-206858, *Proc. of 13th International Symposium on Space Flight Dynamics,* Washington D.C.,Vol.1, pp. 243-257 AAS 98-321, 11-15 May 1998.

[10] Finzi A. E., Vasile M. Optimal Attitude Trajectory Manouvre for Moon Landing. *Proc. of the 49th IAF congrees*, Melbourne, Australia, 2-8 October 1998.

[11] Sansò F., Rummel R. *Theory of Satellite Geodesy and Gravity Field Determination.* Lecture Notes in Earth Sciences, Vol. 25, Springer-Verlag, 1988.

[12] Finzi A.E., Vasile M. Numerical Solution for Lunar Orbits. IAF-97-A.5.08,*Proc. of the 48th International Astronautical Congress*, Torino, Italy, October 6-10, 1997.