IAC-10-C1.9.8

INCREMENTAL SOLUTION OF LTMGA TRANSFERS TRANSCRIBED WITH AN ADVANCED SHAPING APPROACH

Daniel Novak
PhD Candidate, Department of Aerospace Engineering, University of Glasgow, UK
dnovak@ eng.gla.ac.uk

Massimiliano Vasile
Senior lecturer, Department of Aerospace Engineering, University of Glasgow, UK
mvasile@eng.gla.ac.uk

ABSTRACT

In the last decade the global optimisation of low-thrust multi-gravity assist transfers (LTMGA) has been tackled with different approaches. Some authors proposed to generate a first guess solution by building a multi-gravity assist transfer with impulsive manoeuvres and then using a direct or an indirect method to transcribe the multi-impulse arcs into low-thrust arcs. Other authors, notably Petropoulos et al. (2002), De Pascale et al. (2006), Wall et al. (2008) and Schütze et al. (2009), proposed the use of several forms of trajectory shaping to model low-thrust arcs. The disadvantage in all these studies is that the swingbys are powered and therefore suggest the use of high thrust propulsion along with the low thrust propulsion on board the spacecraft. The problem generally resides in the lack of flexibility of the low thrust trajectory models to satisfy a variety of boundary conditions. In this paper, a spherical shaping model is used whereby all encountered types of boundary constraints are satisfied analytically. Furthermore, a special incremental pruning of the search space is performed before employing a global optimiser. The process is conceptually equivalent to the approach proposed by Becerra et al. for the search space pruning of multi-gravity assist trajectories and exploits the decoupling of pairs of transfer arcs. Such decoupling removes the dependency of one arc from all those that are two or more before, and allows for pruning the search space in polynomial time. Numerical examples are presented for LTMGA transfers from Earth to asteroid Apollo and Earth to Jupiter.

INTRODUCTION

The development of electric propulsion for space missions has already given rise to successful missions, like Deep Space 1 [1] and SMART-1 [2]. The increased specific impulse can result in savings in propellant mass for a broad class of mission types. As of today, NASA's Dawn spacecraft is currently heading towards asteroid Vesta and is equipped with an ion thruster. The European Space Agency' is preparing to send a cornerstone mission called BepiColombo to Mercury. Solar electric propulsion will be used for that mission. The design of trajectories for these types of missions is more complex than for those employing chemical propulsion due to the need to optimize thrust profiles instead of impulses. This makes low thrust mission analysis mathematically and computationally a challenging task.

What makes the trajectory optimization problem even more complex for missions like Dawn and BepiColombo is the use of gravity assists. Each gravity assist adds dimensions to the global problem. Since these problems multitudes of local minima, global optimization is unavoidable to obtain the most interesting trajectories.

Before the eighties, multiple gravity assist (MGA) trajectories were computed with ad hoc methods. It was during the design of the Galileo mission in the eighties that the first codes to compute large sets of trajectories, using impulsive manoeuvres. These codes gave rise later to STOUR [3]. Williams and Longuski [4] automated the MGA search. STOUR was then used extensively by Petropoulos et al. [5] for assessing a large number of mission scenarios to Jupiter. They also applied STOUR with a model for low thrust transfers called exponential sinusoids [6]. With the development of the field of global optimisation, different approaches were tested in order to seduce the computational time to find interesting regions in the search space, for both high

thrust and low thrust transfers. These approaches included differential evolution [7], particle swarm optimization [7], evolutionary branching [8] and simulated annealing [9]. Evolutionary neurocontrol was also applied successfully by Carnelli et al. [10] to the low thrust MGA (LTMGA) problems.

However the disadvantage in the current techniques of solving LTMGA problems is that gravity assists involve impulsive manoeuvres, such that both a low thrust and a high thrust propulsion system is assumed on board. There is therefore no method today to compute LTMGA trajectories without employing high thrust manoeuvres, unless local constrained optimizations are performed inside the global optimization [9]. The problem resides in the lack of flexibility of the low-thrust trajectory models. The exponential sinusoids [11] have the disadvantage of being a planar model and one cannot impose boundary constraints on velocity and time of flight together. Pseudo-equinoctial elements, proposed by Vasile et al. [12] can provide first guess trajectories satisfying boundary constraints, time of flight constraint and thrust constraints. However, the satisfaction of the boundary constraints relies on the convergence of a Newton loop, due to the fact that pseudo-equinoctial elements are not osculating. Novak and Vasile [13] formulated a flexible shaping method based on spherical coordinates for which combination of boundary constraints on position and velocity can be solved analytically. This paper applies the latter shaping to the LTMGA problem in order to compute transfers employing only low thrust manoeuvres. Avoiding powered swingbys has a further the advantage of reducing the dimension of the search space.

Moreover, to further reduce the size of the search space, an incremental pruning technique is employed in this study. Incremental pruning has been proposed by Becerra et al. [14] on MGA missions and is based on the construction of sets of MGA trajectories, one leg at a time, and removing subsets that do not satisfy a given criterion, e.g. Δv of the leg too high. The approach exploits the decoupling of the transfer arcs offered by the powered swing-by model. Such decoupling removes the dependency of one arc from the preceding ones, and allows for pruning the search space in polynomial time. The final pruned space can then be explored with a global optimizer. Vasile, Schütze et al. used the exponential sinusoid model [15] to apply incremental pruning to LTMGA problems with powered swingbys.

The present study's aim is to avoid powered swingbys. In that case each leg cannot be completely decoupled from the others. The issue is solved by adapting the incremental pruning whereby the decoupled entities are pairs of legs. The pruning remains therefore of polynomial complexity with respect to the number of legs.

Numerical examples are provided for LTMGA transfers from Earth to Jupiter and Earth to Mercury.

1 TRAJECTORY SHAPING

This section explains the shaping method used to generate the trajectories for each leg of multiple gravity-assist trajectories. More detailed derivations and discussions can be found in [13] A trajectory model based on shaping the spherical coordinates is described. The last subsection describes the method used to satisfy time of flight constraints.

1.1 Shaping the Spherical Coordinates

In the proposed model, the spacecraft's position is expressed in the spherical coordinates $(r,\theta,\varphi)\in \mathbb{R}_+^* \times \mathbb{R}/2\pi\mathbb{Z} \times \left(-\pi/2 + \mathbb{R}/\pi\mathbb{Z}\right)$, where r is the distance from the central body, θ is the azimuthal angle and is φ the elevation angle. If the trajectory was parametrized by the time, the state vector would be $\left(r,\theta,\varphi,\dot{r},\dot{\theta},\dot{\varphi}\right)^T$. Here we assume that the trajectory can be parametrized by θ, i.e. $r = R(\theta)$, $\varphi = \Phi(\theta)$ and $t = T(\theta)$, so that the azimuthal angle swaps role with the time t. This can be done if there is a smooth mapping between t and θ, which implies that θ is strictly monotonous with respect to time. The state vector becomes thus $x = \left(r,t,\varphi,r',t',\varphi'\right)^T$ where the prime represents a derivative with respect to θ. Due to this parametrization, the poles need to be excluded from the set of admissible positions and we have to allow θ to account for the n_r revolutions of the trajectory. The configuration space W is defined hence as $W = \mathbb{R}_+^* \times \left[\theta_i;\theta_f + 2n_r\pi\right] \times \left(-\pi/2;\pi/2\right)$.

The equations of motion in an inertial reference frame satisfied by the spacecraft are given by

$$\frac{d^2\boldsymbol{r}}{dt^2} = -\mu\frac{\boldsymbol{r}}{r^3} + \boldsymbol{u} \qquad (1)$$

where the position vector is $\boldsymbol{r} = \left(r\cos\theta\cos\varphi, r\sin\theta\cos\varphi, r\sin\varphi\right)^T$. If the position vector is parametrized by θ the equations of motion become:

$$\dot{\theta}^2 \frac{d^2\boldsymbol{r}}{d\theta^2} + \ddot{\theta}\frac{d\boldsymbol{r}}{d\theta} = -\mu\frac{\boldsymbol{r}}{r^3} + \boldsymbol{u} \tag{2}$$

Note that $\dot{\theta} = 1/t'$ and $\ddot{\theta} = -t''/t'^3$. The control vector is obtained straight from equation (2), after having inserted the expression of $\boldsymbol{r}$ as a function of the spherical coordinates.

At this point, one can provide *R*, *Φ* and *T*, i.e. one can "shape" *r*, *φ* and *t*, and the corresponding control profile can be obtained, along with the *Δv* and the propellant consumption if the spacecraft's initial mass and specific impulse are provided. *R* and *Φ* model the pure geometry of the trajectory, while *T* shapes the dynamics along the trajectory. It is assumed that shaping functions *R, Φ* and *T* belong to a set of admissible functions S_R, S_Φ and S_T that are continuously differentiable twice.

We define $\tilde{\boldsymbol{v}} = d\boldsymbol{r}/d\theta = \begin{pmatrix}\tilde{v}_r & \tilde{v}_\theta & \tilde{v}_\varphi\end{pmatrix}$, $\tilde{\boldsymbol{h}} = \boldsymbol{r} \wedge \tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{a}} = d\tilde{\boldsymbol{v}}/d\theta = \begin{pmatrix}\tilde{a}_r & \tilde{a}_\theta & \tilde{a}_\varphi\end{pmatrix}$ and the flight path angle is denoted here by *γ*.

It can be shown that *t* satisfies

$$\frac{D}{T'^2} = \frac{\mu}{R^2} + \frac{u_n}{\cos\gamma} \tag{3}$$

where *D* is given by

$$D = \frac{\tilde{\boldsymbol{a}} \cdot \left(\tilde{\boldsymbol{h}} \wedge \tilde{\boldsymbol{v}}\right)}{R\left(\tilde{v}_\theta{}^2 + \tilde{v}_\varphi{}^2\right)} = \frac{\boldsymbol{a} \cdot \left(\boldsymbol{h} \wedge \boldsymbol{v}\right)}{R\dot{\theta}^2\left(v_\theta{}^2 + v_\varphi{}^2\right)} \tag{4}$$

So if we define the function T_0 by

$$T_0'^2 = \frac{DR^2}{\mu} \tag{5}$$

And assume that the time evolution *T* is shaped by T_0, then the control vector corresponding to the geometrical trajectory defined by *R* and *Φ* will have no component out of the tangential plane to the trajectory ($u_n = 0$). There is a restriction on the shape of the trajectory for which this control strategy is allowed, it is expressed by $D > 0$. Physically speaking, the plane generated by tangential and out-of-plane vectors (or all the allowed control vectors) divides the space in two, and the centre of curvature of the trajectory at every point must be on the same side of the plane as the central body. This makes sense, because if the acceleration points outwards from the central body with respect to the plane of allowed controls, then a control component outside of the latter plane is required to compensate the gravitational pull of the central body and therefore $u_n \neq 0$. In the particular case of a two dimensional trajectory, we obtain

$$D = \frac{cR^2}{\cos^3\gamma} \tag{6}$$

where *c* is the algebraic curvature of the trajectory. Thus $D > 0$ if the curvature is positive. The time of flight and the *Δv* corresponding to the trajectory are obtained by integration over $\left[\theta_i; \theta_f + 2n_r\pi\right]$ of T' and $\left|\boldsymbol{u}\right|T'$ respectively.

The disadvantage of fixing $T = T_0$ is that *R* and *Φ* define completely T' and the time of flight $T\left(\theta_f\right) - T\left(\theta_i\right)$. This can be problematic when a constraint on the time of flight exists. The advantage is that it is difficult to shape a priori *T* in a way to obtain a control that is not too far from optimal, i.e. interesting in practice, therefore using such an expression for T' will result, for certain transfers (see section 3), in reasonable thrust profiles and *Δv*. By imposing $u_n = 0$, the in-plane motion is controlled only by the tangential component of the control, which is the most efficient way to vary the energy of the osculating orbit. Such shaping for *T* was chosen in this study; only S_R and S_Φ need to be defined in this way.

R and *Φ* can be in any function space such that $R > 0$ and $-\pi/2 < \Phi < \pi/2$, but it is judicious to choose expressions for which the boundary constraints on the position and velocity can be solved analytically. The boundary conditions are written in equations (7).

$$\begin{cases} R\left(\theta_i\right) = R_i & R\left(\theta_f + 2n_r\pi\right) = R_f \\ \Phi\left(\theta_i\right) = \Phi_i & \Phi\left(\theta_f + 2n_r\pi\right) = \Phi_f \\ T'\left(\theta_i\right) = \dfrac{R_i\cos\Phi_i}{v_{\theta i}} & T'\left(\theta_f + 2n_r\pi\right) = \dfrac{R_f\cos\Phi_f}{v_{\theta f}} \\ R'\left(\theta_i\right) = v_{ri} & R'\left(\theta_f + 2n_r\pi\right) = v_{rf} \\ \Phi'\left(\theta_i\right) = \dfrac{v_{\varphi i}}{R_i} & \Phi'\left(\theta_f + 2n_r\pi\right) = \dfrac{v_{\varphi f}}{R_f} \end{cases} \tag{7}$$

Note that the boundary conditions on T' turn into boundary conditions on R'' and Φ'', as shown in (8).

$$\begin{cases} R''\left(t_i\right) + \alpha_i\Phi''\left(t_i\right) = C_i \\ R''\left(t_f\right) + \alpha_f\Phi''\left(t_f\right) = C_f \end{cases} \tag{8}$$

There are hence 10 boundary conditions, so unless there are more than 10 parameters that can be set in R and Φ combined, the time of flight is uniquely defined.
In the examples in this paper, S_R and S_Φ are the set of functions that take the form:

$$\begin{cases} R = \dfrac{1}{a_0 + a_1\theta + a_2\theta^2 + (a_3 + a_4\theta)\cos\theta + (a_5 + a_6\theta)\sin\theta} \\ \Phi = (b_0 + b_1\theta)\cos\theta + (b_2 + b_3\theta)\sin\theta \end{cases} \quad (9)$$

Expressing the radius and the elevation angle in these forms has the advantage of covering the Keplerian unperturbed arc, provided that the trajectory is by the spherical coordinates derived from the rotated inertial frame whose (x-y) plane is defined by the initial position and velocity vectors.
11 coefficients are to be determined by the boundary conditions and a constraint on the time of flight. Subsection 1.2 explains the method used to solve time of flight constraints.
It must be mentioned that due to the ability to solve analytically the boundary conditions, one can apply any subset of constraints from the set in (7), after adjusting the number of undetermined coefficients in (9). For example, one can impose the boundary conditions on the position at both tips and on the velocity at arrival only. In that case, the number of constraints is three less, and two coefficients are set to zero in the expression of R and on in the expression of Φ. As will be explained, this is what happens in the last leg of the pruning phase.

1.2 Satisfaction of the time of flight constraint

In this subsection it is explained how the time of flight constraint is satisfied exactly.
The expressions for the shaping functions in (9) involve a number of coefficients that can be used for satisfying several constraints. The boundary constraints account for ten of them. The time of flight constraint sets the value of an additional coefficient or parameter. Since the time of flight depends non-linearly with respect to the additional unknown, a Newton loop is applied to solve the constraint.
The coefficient a_2 in (9) is initially set to zero and within each iteration of the Newton loop the other 10 coefficients are computed analytically from (7) and (8).
Using a Newton loop for satisfying time of flight constraints in the described manner preserves the assumptions made on the trajectory by assigning expressions to T' such as in (5). However due to the non-linear nature of the time of flight constraint, the Newton loop can fail to reduce the time of flight violation T_{viol} to zero.
If that scenario happens another method is applied after the Newton loop. The time evolution T is modified in such a way that the time of flight is satisfied exactly. At the exit of the Newton loop, one reshapes T by assigning it the new expression $T = T_0 - T_{viol}\chi$ where

$$\chi(\theta) = -\frac{6}{(\theta_f - \theta_i)}\int_{\theta_i}^{\theta}(u - \theta_i)(u - \theta_f)\,du \quad (10)$$

It can be verified that $\chi'(\theta_i) = \chi'(\theta_f) = 0$ and therefore T' takes the same values at the boundaries as T_0', so the boundary constraints are not affected. Moreover, because $\chi(\theta_f) - \chi(\theta_i) = 1$, the time of flight $T(\theta_f) - T(\theta_i)$ will be the exact desired time of flight.
Reshaping T in such a way can alter the control profile considerably and can potentially result in higher Δvs, if the value of $|T_{viol}|$ becomes high. Furthermore, for higher values of $|T_{viol}|$, T' can become zero along the trajectory. If that happens, the control profile will exhibit a singularity, since $\dot{\theta} = 1/t'$ and $\ddot{\theta} = -t''/t'^3$.

2 LINEAR QUADRATIC CONTROLLER

In this section the method used to improve in a fast way low-thrust transfers is explained. In practice, this approach is applied to the transfers generated by the shaping methods. The objective is to cope with the arbitrariness of the choice to model the state vector **x**. The rationale behind this approach is that there may be an easily computable trajectory in the vicinity of the shaped trajectories that is better in terms of the given objective function. It is not intended at this stage to come up with an optimal trajectory, but to obtain improved trajectories that cannot be modelled by the shaping functions that were used at the previous step of the trajectory design. A more detailed derivation exists in [16].

Let's assume that a spacecraft, whose position is **r** and velocity is **v**, is subject to the gravitational pull of a central body. Additionally the spacecraft has an onboard controllable propulsion system that contributes with an acceleration **u** to the motion of

the spacecraft. If we define the state vector $\mathbf{x}$ as $\begin{pmatrix}\mathbf{r}^T & \mathbf{v}^T\end{pmatrix}^T$ then the equations of motion can be written in the following as $\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}) + \mathbf{B}\mathbf{u}$.

The equations of motion are then linearized in $\mathbf{x}$ and $\mathbf{u}$ around the reference trajectory, at all points of the time interval I, as in (11), which can be rewritten as in (12). The control vector of the linearized equations of motion are denoted $\mathbf{u}_l$.

$$\begin{cases}\boldsymbol{\xi}(t_i) = (0\,0\,0\,0\,0\,0)^T \\ \dot{\boldsymbol{\xi}} = \mathbf{A}^*(t)\boldsymbol{\xi} + \mathbf{B}\upsilon = \mathbf{A}^*(t)\boldsymbol{\xi} + \mathbf{B}(\mathbf{u}_l - \mathbf{u}^*(t))\end{cases} \quad (11)$$

$$\begin{cases}\boldsymbol{\xi}_1(t_i) = (0\,0\,0\,0\,0\,0\,1)^T \\ \dot{\boldsymbol{\xi}}_1 = \mathbf{A}_1(t)\boldsymbol{\xi}_1 + \mathbf{B}_1\mathbf{u}_l\end{cases} \quad (12)$$

The optimal control, corresponding to an unconstrained optimization problem, where the dynamics are defined by (12) and where the objective function is of quadratic form as in (13), can be put in a feedback form expressed in equation (14) .

$$J = \boldsymbol{\xi}_1{}^T(t_f)\mathbf{Q}\boldsymbol{\xi}_1(t_f) + \int_I \mathbf{u}_l{}^2 dt \quad (13)$$

$$\mathbf{u}_l = \mathbf{B}_1^T\mathbf{E}\boldsymbol{\xi}_1 \quad (14)$$

$\mathbf{E}$ is computed by integrating backwards the Riccati differential equation in (15).

$$\begin{cases}\mathbf{E}(t_f) = -\mathbf{Q} \\ \dot{\mathbf{E}} = -\mathbf{A}_1{}^T\mathbf{E} - \mathbf{E}\mathbf{A}_1 - \mathbf{E}\mathbf{B}_1\mathbf{B}_1{}^T\mathbf{E}, \forall t \in I\end{cases} \quad (15)$$

The first term in the objective function will make $\boldsymbol{\xi}_I$ tend towards 0, which is what is required: the perturbations on the trajectory should not affect the boundaries. The fact that the last component of $\boldsymbol{\xi}_I$ is always 1 is not problematic because the choice of $\mathbf{Q}$ is made such that it does not influence the convergence of the other components of $\boldsymbol{\xi}_I$ towards 0. The expression in (16) is chosen for Q.

$$\mathbf{Q} = \begin{pmatrix} q_r\mathbf{I}_3 & 0 & 0 \\ 0 & q_v\mathbf{I}_3 & 0 \\ 0 & 0 & 0 \end{pmatrix}, q_r > 0, q_v > 0 \quad (16)$$

q_r is a weight on the final position vector to satisfy the final boundary constraint, and q_v has the same role but for the velocity. The values for the two weights were set to 1 in order to satisfy all the time the boundary conditions at arrival up to relative violations of 10^{-6}.

Therefore, the optimization requires the integration of a 7 by 7 matrix differential equation backwards in time, followed by the forward integration of the linearised equations of motion using the previously computed expression of $\mathbf{E}$. The first integration can be made faster by noting that $\mathbf{E}$ is a symmetrical matrix, hence it is sufficient to compute 28 variables instead of 49. In order to do that, Matlab's 'ode45' 4th-5th order Runge Kutta integrator was used with relative and absolute tolerances of 10^{-9}.

Once the optimized linearised trajectory *(*$\mathbf{x}^*+\boldsymbol{\xi}$*)* is obtained, the corresponding control law needs to be updated since it verifies the linearised equations of motion and not the real ones. The real control law corresponding to the physical trajectory is calculated from:

$$\mathbf{u}_{real} = \ddot{\mathbf{r}}_l + \frac{\mu}{r_l^3}\mathbf{r}_l \quad (17)$$

where $\mathbf{r}_l$ is the position component of $\mathbf{x}_0+\boldsymbol{\xi}$. Note that the approach of keeping the linearised control law and calculating the corresponding state vectors by propagation would not only be more computationally intensive but would not guarantee that the trajectory would end at the target state vector.

The error on the control law due to the linearisation is approximated in [16] to be

$$\begin{cases}\Delta u_x = \boldsymbol{\xi}_\mathbf{r}^T\,\mathbf{H}_\mathbf{A}^1\big|_{r_0}\,\boldsymbol{\xi}_\mathbf{r} + O\left(\|\boldsymbol{\xi}_\mathbf{r}\|^3\right) \\ \Delta u_y = \boldsymbol{\xi}_\mathbf{r}^T\,\mathbf{H}_\mathbf{A}^2\big|_{r_0}\,\boldsymbol{\xi}_\mathbf{r} + O\left(\|\boldsymbol{\xi}_\mathbf{r}\|^3\right) \\ \Delta u_z = \boldsymbol{\xi}_\mathbf{r}^T\,\mathbf{H}_\mathbf{A}^3\big|_{r_0}\,\boldsymbol{\xi}_\mathbf{r} + O\left(\|\boldsymbol{\xi}_\mathbf{r}\|^3\right)\end{cases} \quad (18)$$

where $\left[\Delta u_x, \Delta u_y, \Delta u_z\right]^T = \mathbf{u}_l - \mathbf{u}_{real}$ and $\boldsymbol{\xi}_\mathbf{r}$ as the first three components of $\boldsymbol{\xi}$, i.e., the change of position resulting from the LQ controller. One also has that all of $\mathbf{H}_\mathbf{A}^i$ depends only on $\mathbf{r}_0$ and that $\left\|\mathbf{H}_\mathbf{A}^i\right\| = O\left(\|\mathbf{r}_0\|^{-4}\right)$.

A theorem has been proven on the behaviour of the LQ controller when applied to an optimal initial trajectory. It states that the LQ controller leaves the initial trajectory unchanged if and only if the initial trajectory is already optimal. So in a certain sense, the result of the LQ controller is a measure of the optimality of the initial trajectory.

Finally, the total Δv can be calculated by an integration of u_{opt} over I. No assumptions are made

on the physical properties of the spacecraft and its propulsion system, in particular on the initial mass and the specific impulse, because it was chosen to model the trajectory in the simplest way. Introducing the spacecraft's mass as a variable would raise the number of differential equations to compute backwards from 28 to 36 and forwards from 6 to 7. This would have repercussions on computational time without necessarily improving the results since optimizing the L_2 norm of the thrust vector does not yield necessarily better results than optimizing the L_2 norm of the acceleration control vector.

3 INCREMENTAL PRUNING

3.1 Background

Incremental pruning is a technique first proposed by Becerra et al. [14] for finding globally optimum multiple gravity-assist trajectories. The idea is that if one can construct the legs of the MGA transfers independently, then it is possible to prune out whole sets of transfers if one of the legs does not satisfy some required criteria. By constructing and assessing legs one after the other, the space of acceptable transfers is pruned out incrementally. Once the final pruned search space for the full problem is obtained, a global optimisation can be performed on it. It has been shown that applying such a pruning can increase the chance of finding the most promising trajectories.

As introduced by Becerra et al. in their code called GASP (Gravity Assist Space Pruning), the pruning relies on a systematic search on the discretised search space. The problem is formulated in such a way that the different legs can be constructed independently, but can also be linked together to form complete MGA trajectories. In the first form of pruning introduced by Becerra et al. the legs were Lambert arcs linked together by powered swingbys. The search space consisted of a grid of departure dates, encounter dates for the gravity assist and arrival dates. All the possible Lambert arcs were constructed for the first leg and a pruning of the departure and first gravity-assist dates was performed, based on the magnitude of the initial relative velocity. If there are launch dates for which no Lambert arc is acceptable, then that launch date is pruned out for the problem. In a similar manner, if no Lambert arc is acceptable for a given date for the first gravity-assist, then that date is not considered as starting date for the second leg. In the next step, all possible second legs are constructed except for the dates of first gravity-assist that were pruned out in the previous step. Criteria to prune out the initial and final dates of the second leg are based on the maximum thrust constraints and angular constraints on the incoming and outgoing relative velocities at the first gravity assist.

The following legs are constructed and pruned out similarly to the second leg. A constraint on the relative arrival velocity is imposed during the pruning of the final leg's departure and arrival dates.

After the computation of each leg and the pruning of departure and arrival dates of it, an additional forward and backward pruning is performed on the complete space, based on the consideration that if no Lambert arc arrives on a given date for a gravity assist, then that date is not considered as departure date for the next leg, and if no Lambert arc can depart on a given date, then all Lambert arcs of the previous leg and arriving on that date are pruned out.

Once the grid space of the complete problem is pruned, one recovers the acceptable combinations of intervals for the initial and arrival dates of each leg. One obtains hence so-called boxes, and a global optimisation is performed on each. Becerra et al. applied differential evolution.

The algorithms applied by GASP have been used with trajectory models that are more complex than Lamberts arcs. GASP has been tested successfully when deep space manoeuvres (DSM) is inserted in each leg [20]. DSMs increase the flexibility of each leg and represent more realistic missions, to the expense of an increased dimension of the search space. The objective in that case is to minimise the sum of the DSMs' Δv and the gravity assists' Δv. Schütze et al. applied GASP with exponential sinusoids [15][21] as trajectory models for the problem of optimizing low thrust MGA transfers. The inconvenience with the latter approach, however, is that one needs to employ powered swingbys, with impulsive Δvs, and suggests therefore both a chemical and a low-thrust propulsion system on board the spacecraft.

3.2 Overview of the method

This lack of realism is addressed here, by eliminating the need of a manoeuvre at the swingby. The issue arising then is that successive legs cannot be computed independently, because the outgoing relative velocity at a gravity assist must be reachable from the incoming relative velocity obtained in the previous leg, while setting a limit on the pericenter radius. For example, the magnitudes of the two relative velocities must be equal.

Therefore the GASP algorithm has been modified such that legs are independent in pairs instead of on their own. The low-thrust trajectory model is such that legs departing and arriving on given dates always arrive with the same velocity, but can depart with different velocities depending on the arrival velocity of the previous leg.

For the first leg, only Lambert arcs are considered. For each additional leg, the Lambert arcs are first computed, and the initial relative velocity is compared with the previous leg's arrival relative velocity. If the two can be matched, then the Lambert arc is kept, otherwise the initial relative velocity is modified such that no impulse is required at the gravity assist. The leg is then recomputed with a low-thrust trajectory model which can accommodate boundary constraints on velocity and time of flight. Hence the departure date t_{i-1} of the previous leg defines the incoming relative velocity at the swingby at date t_i, so the outgoing relative velocity too, and the arrival date t_{i+1} defines the arrival velocity.

3.3 Gravity assist model

Most often the outgoing relative velocity $\mathbf{v}_{i,rel}^{(i)}$ at the beginning of a given leg *i* cannot be matched with the incoming one $\mathbf{v}_{f,rel}^{(i-1)}$, obtained from the previous leg, without going below the minimum pericenter radius specified by the user. Then a transformation is applied to the outgoing relative velocity. The new relative velocity $\mathbf{v}_{i,rel}'^{(i)}$ is such that $\left\|\mathbf{v}_{i,rel}'^{(i)} - \mathbf{v}_{i,rel}^{(i)}\right\|$ is minimal, while keeping the new velocity achievable without any manoeuvre during the gravity assist.

It can be noted that $\mathbf{v}_{i,rel}'^{(i)}$ is always in the plane defined by $\mathbf{v}_{f,rel}^{(i-1)}$ and $\mathbf{v}_{i,rel}^{(i)}$. The angle δ between $\mathbf{v}_{f,rel}^{(i-1)}$ and $\mathbf{v}_{i,rel}^{(i)}$ is computed, and if it is greater than the maximum bending angle allowed by the gravity assist [13]

$$\delta_{\max} = 2\arcsin\frac{1}{1+\frac{r_{p,\min}\left\|\mathbf{v}_{f,rel}^{(i-1)}\right\|^2}{\mu}}$$

then $\mathbf{v}_{i,rel}'^{(i)}$ is defined such that its angle with $\mathbf{v}_{f,rel}^{(i-1)}$ is $\delta_{\max}$. If the line carrying $\mathbf{v}_{f,rel}^{(i-1)}$ separates the plane defined by $\mathbf{v}_{f,rel}^{(i-1)}$ and $\mathbf{v}_{i,rel}^{(i)}$ in two, $\mathbf{v}_{i,rel}'^{(i)}$ points towards the same half-plane as $\mathbf{v}_{i,rel}^{(i)}$. $\mathbf{v}_{i,rel}'^{(i)}$ is then fully defined by assigning it the same magnitude as $\mathbf{v}_{f,rel}^{(i-1)}$. Fig. 1 illustrates this transformation.

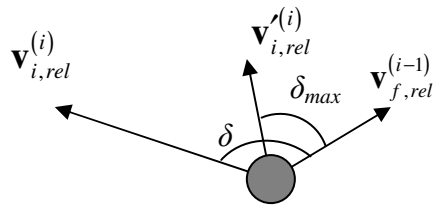

Fig. 1: Illustration of the transformation applied to the initial relative velocity of leg I if it cannot be obtained by a gravity-assist with incoming relative velocity $\mathbf{v}_{f,rel}^{(i-1)}$.

If $\delta \le \delta_{\max}$ then only the magnitude of $\mathbf{v}_{i,rel}^{(i)}$ is adjusted to $\left\|\mathbf{v}_{f,rel}^{(i-1)}\right\|$, if necessary.

3.4 Detailed description of the method

The inputs to the problem are the sequence of *N* planets to be encountered, the launch window I_0 and the range of times of flight $(I_i)_{1\le i\le N}$ allowed for each leg of the transfer. The total search space is therefore $I = I_0 \times I_1 \times \cdots \times I_N$. One also provides *N+1* integers greater than 1, representing the number of points on the grid to discretize each I_i. The discretized I_i are denoted I_i^d. One can then construct recursively *N* new sets $\left(I_i^{d*}\right)_{1\le i\le N}$ initializing from $I_0{}^d$. Then I_i^{d*} is the set of possible encounter dates of planet *i* such that the spacecraft left planet *i-1* at a time in I_{i-1}^{d*} and the time of flight of leg *i* is in I_i^d. From there on the pruning then works on the sets of encounter dates $\left(I_i^{d*}\right)_{1\le i\le N}$. A maximum magnitude of launch and arrival relative velocities can be specified, as well as maximum values of Δv for each leg, according to which the pruning will be performed. Moreover minimum pericenter radii have to be specified for each gravity-assist.

The pruning algorithm starts by generating all possible first legs. Three Lambert arcs are computed for each element of $I_0^d \times I_1^{d*}$, one with no revolution, and two with one revolution (the long arc and the short arc). Out of the three Lambert arcs, only the one whose initial relative velocity is lowest is retained. If one of the Lambert arcs has a zero initial relative velocity, which can happen if the first gravity assist planet is identical to the launch planet,

then that Lambert arc is directly discarded. The retained Lambert arc is then analyzed and the element in $I_0^d \times I_1^{d*}$ to which it corresponds can be invalidated if the launch velocity is above the specified threshold. At the end of this step, a set of valid points in $I_0^d \times I_1^{d*}$ is obtained. If there are dates in I_0^d or in I_i^{d*} for which all Lambert arcs were pruned out, i.e. there are lines or columns in $I_0^d \times I_1^{d*}$ full with invalid transfers, then those dates are pruned out and one obtains the new valid sets $V^{(1)}\left(I_0{}^d\right)$ and $V^{(1)}\left(I_1{}^{d*}\right)$.

From the second leg onward, five Lambert arcs are generated for each element of $I_1^{d*} \times I_2^{d*}$: one for zero revolutions and the short and long arc for both the one revolution case and the two revolutions case. Like for the previous step, only the Lambert with the smallest non-zero relative velocity at start is retained. For that Lambert arc, the initial relative velocity $\mathbf{v}_{i,rel}^{(2)}$ is compared to all the incoming relative velocity vectors $\mathbf{v}_{f,rel}^{(1)}$ of the first leg. For each case, $\mathbf{v}_{i,rel}^{\prime(2)}$ is constructed, if necessary, with the procedure described in subsection 3.3 and then a shaped trajectory is generated to replace the Lambert arc, such that the initial relative velocity is $\mathbf{v}_{i,rel}^{\prime(2)}$ instead of the Lambert arc's $\mathbf{v}_{i,rel}^{(2)}$. All other boundary conditions are kept the same as the Lambert arc's. Thanks to the nature of the shaping, the shaping covers the Keplerian motion so the closer $\mathbf{v}_{i,rel}^{\prime(2)}$ is to $\mathbf{v}_{i,rel}^{(2)}$, the closer the shaped trajectory will be to the Lambert arc.

Potentially one computes then a shaped trajectory for every node of $V^{(1)}\left(I_0{}^d\right) \times V^{(1)}\left(I_1{}^{d*}\right) \times I_2{}^{d*}$. For each shaped trajectory one verifies if the total Δv is smaller than what the user specifies and if not that element of $V^{(1)}\left(I_0{}^d\right) \times V^{(1)}\left(I_1{}^{d*}\right) \times I_2{}^{d*}$ is marked invalid. One can perform then the same forward and backward pruning as for the first leg. If there are dates of encounter in $V^{(1)}\left(I_1{}^{d*}\right)$ and $V^{(1)}\left(I_2{}^{d*}\right)$ such that all legs arriving on those dates are invalid, then those dates are pruned out for the rest of the computations. In an analogous way, if all possible departing legs are invalid for a given date, then that date is pruned out. One obtains new sets of valid dates $V^{(2)}\left(I_0{}^d\right)$, $V^{(2)}\left(I_1{}^{d*}\right)$ and $V^{(1)}\left(I_2{}^{d*}\right)$.

The trajectories of the following legs are generated and assessed in the same way as those of the second one, and each time the set of valid dates is updated with forward and backward pruning.

The last leg is special if the aim is to rendezvous the final planet. Indeed, in that case, the Lambert arc is not a good way to generate the first trajectory leg, because the arrival velocity can be far from the planet's velocity. In that case, a shaped trajectory is generated instead of the Lambert arc, where the initial velocity is not constrained, and the final velocity is that of the planet. The coefficients a_4, a_6 and b_3 are set to zero in the expressions of R and Φ in (9).

At this point, one has to analyze the distribution of the dates defining the acceptable pairs of consecutive legs. An acceptable pair of legs $(j, j+1)$ will be defined by a triplet of dates $\left(t_{j-1}, t_j, t_{j+1}\right)$ that is an element of $V^{(i_{j-1})}\left(I_{j-1}{}^{d*}\right) \times V^{(i_j)}\left(I_j{}^{d*}\right) \times V^{(i_{j+1})}\left(I_{j+1}{}^{d*}\right)$. These acceptable triplets of dates form a subset $V_{j,j+1} \subset V^{(i_{j-1})}\left(I_{j-1}{}^{d*}\right) \times V^{(i_j)}\left(I_j{}^{d*}\right) \times V^{(i_{j+1})}\left(I_{j+1}{}^{d*}\right)$. One can then proceed to reconstruct the continuous search spaces from the pruned discretized ones. For this, the connected components $V_{j,j+1}^k$ inside every $V_{j,j+1}$ are identified and boxes $BV_{j,j+1}^k$ are created around each one of them. Finally, hyperboxes BV^l inside I are identified such that an element inside it has each of its components belonging to one of the $BV_{j,j+1}^k$.

The hyperboxes are disconnected subsets of I, and one can then apply a global optimisation algorithm on each one of them. In this study, a differential evolution algorithm was applied. When applying it, the LQ controller was called after the shaping in order to improve the transfers.

3.5 Computational effort

During the pruning, the part that takes by far the most time is the generation of the shaped trajectory. The trajectory shaping is called from the second leg onward. If one discretizes the initial and arrival dates of the leg into k points, as well as the previous leg's initial dates, then the shapes are called potentially k^3 times. This is the case for all the legs except the first and the last, which adds up to $(N-1)k^3$. In the final leg there are two calls to the trajectory shaping for each node, adding up to $8k^3$ calls. So in total, the shaping is called at most $(N+7)k^3$ times.

4 TEST CASES

This section presents the test cases to which the shaping and the new incremental pruning was applied to. Rendezvous missions are presented from Earth to asteroid Apollo and Jupiter. The computations were performed on Intel Core 2 Duo running at 3 GHz. The codes were written in Matlab and were run in a Linux environment. The trajectories with the lowest Δv were optimized with DITAN with the objective of minimising the propellant mass with the constraint of a peak thrust equal to the peak thrust of the initial guess.

4.1 Earth-Earth-Apollo rendezvous

This mission scenario has four leg. Apollo's orbital elements are reported in Table 1.

Table 1 Orbital elements of Asteroid Apollo

Semi-major axis	1.471 AU
Eccentricity	0.56
Inclination	6.4°
Ascending node	25.9
Argument of periapsis	285.7

The launch window I_0 was set to be the interval between 1st January 2010 and 1st January 2015. I_0 was discretized into 240 equidistant dates, i.e. in average three launch dates per month. The first leg's range of times of flight was set to [200 d ; 800 d] and discretized into 250 points, the second one spanned [200 d ; 1000 d] and split into 300 values. The initial relative velocity was allowed to be 5 km/s maximum and a limit was set on the second leg's total Δv to 10 km/s. The minimal altitude allowed for the gravity-assist at Earth was 200 km. The pruned pairs of legs are plotted in Fig. 2.

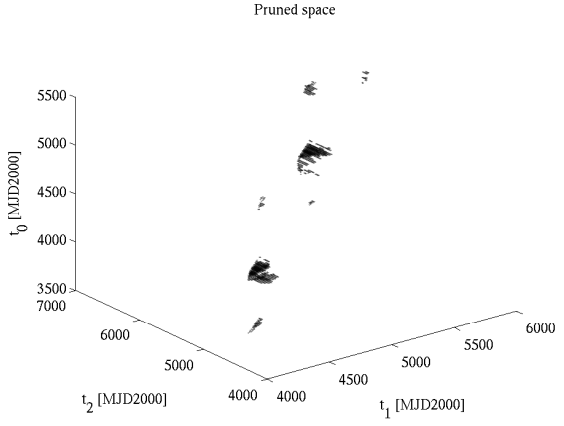

Fig. 2: plotted are the triplets of dates, corresponding to the pair of Earth-Earth-Apollo legs, that were not pruned

In total, 29 separate hyperboxes were obtained by the pruning after 8.5 hours of computation, and a differential evolution algorithm was run on both of them, in order to locate the global minimum, taking 8.9 minutes each.

The trajectory with the lowest Δv obtained from the DE turned out to be of 5.32 km/s with an initial launch velocity of 4.93 km/s. One should note that part of the Δv of the low thrust transfer includes gravity-loss. The total time of flight is 3.17 years. The local optimization with DITAN resulted in a transfer also lasting 3.17 years and requiring 4.49 km/s of Δv and an initial launch velocity of 5.00 km/s.

Table 2 provides the dates of encounter of each planet for the best trajectory.

Fig. 3 and Fig. 4 show that the optimized transfer is close to the initial guess one.

Table 2: Dates at each planet for the LTMGA trajectory with the lowest Δv.

	DE	DITAN
Launch from Earth	9/2/2011	19/3/2011
Earth GA	27/7/2012	4/9/2012
Rdv at Apollo	12/4/2014	20/5/2014

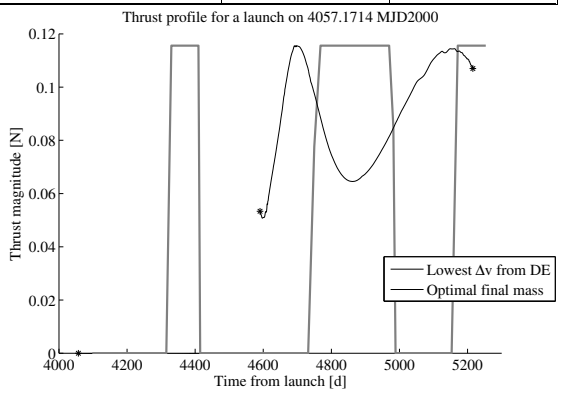

Fig. 3 Thrust profile of the result from DE with the lowest Δv and its optimized solution from DITAN

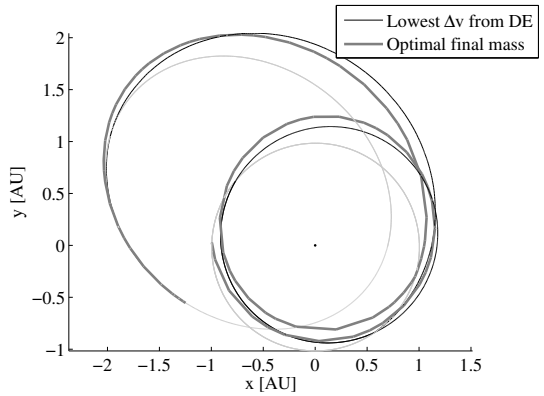

Fig. 4 Trajectory plo of the result from DE with the lowest Δv and its optimized solution from DITAN

4.2 Earth-Venus-Earth-Earth-Jupiter rendezvous

This LTMGA transfer has four legs. The launch window I_0 was set to be the interval between 1st January 2010 and 1st January 2020. I_0 was discretized into 240 equidistant dates, i.e. in average two launch dates per month. The first leg's range of times of flight was set to [50 d ; 500 d], the second one's to [50 d ; 700 d], the third one to [100 d ; 1000 d] and the fourth to [500 d ; 2000 d]. The ranges of time of flight were discretized into respectively 45, 65, 90 and 75 points. The initial relative velocity was allowed to be 5 km/s at most and limits were set on the last three leg's total Δv to 10 km/s, 10 km/s and 15 km/s respectively. The minimal altitude allowed for all gravity-assists was 200 km. The pruned pairs of legs are plotted in Fig. 5 to Fig. 7.

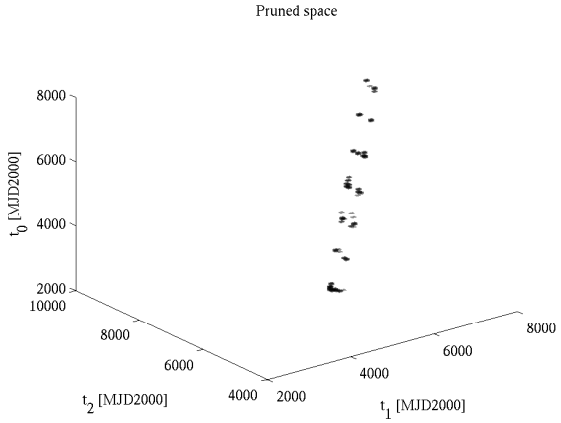

Fig. 5: plotted are the triplets of dates, corresponding to the pair of Earth-Venus-Earth legs, that were not pruned

Table 3: Dates at Each planet for the LTMGA trajectory with the lowest Δv.

	DE	DITAN
Launch from Earth	15/1/2017	3/2/2017
Venus GA	20/4/2017	20/4/2017
Earth GA 1	28/8/2018	1/9/2018
Earth GA 2	25/3/2021	25/3/2021
Rendezvous at Jupiter	8/9/2024	27/12/2024

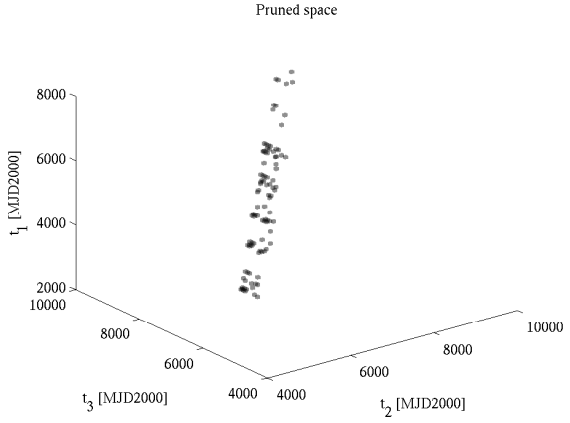

Fig. 6: plotted are the triplets of dates, corresponding to the pair of Venus-Earth-Earth legs, that were not pruned not pruned for the pair of first and second legs.

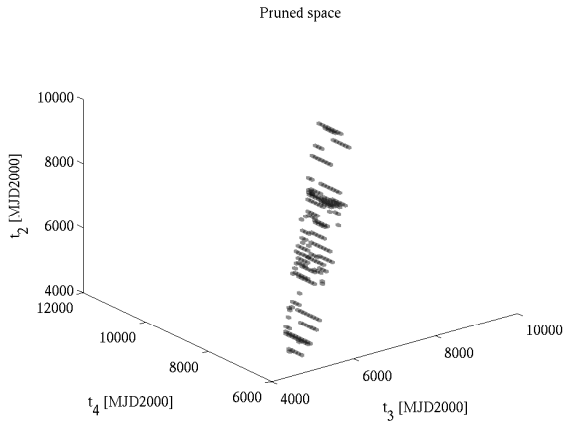

Fig. 7: plotted are the triplets of dates, corresponding to the pair of Earth-Earth-Jupiter legs, that were not pruned

In total, 5 separate hyperboxes were obtained by the pruning after 26 minutes of computation, and a differential evolution (DE) algorithm was run on all

hyperboxes, in order to locate the global minimum. In average, each minimisation ran for 30 minutes in average.

The lowest Δv obtained from the DE was 7.66 km/s with an initial launch velocity of 3.58 km/s. The Δv includes the gravity loss due to the low and long nature of the thrust arcs. This result is to be compared with the Hohmann transfer's total Δv of 14.44 km/s. The total time of flight amounted to 7.65 years. The optimization with DITAN resulted in a transfer of 7.89 years of duration with a Δv of 6.64 km/s and with an initial launch velocity of 5 km/s.

Fig. 8 and Fig. 9
Table 3Table 3 provides the dates of encounter of each planet for the best trajectory.

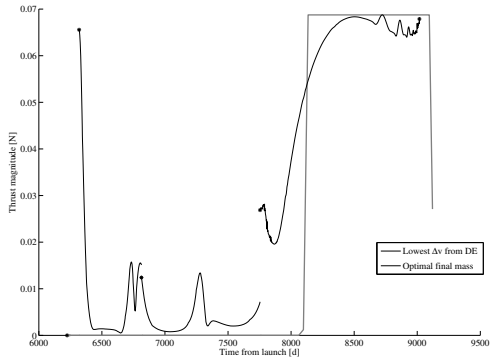

Fig. 8 Thrust profile of the result from DE with the lowest Δv and its optimized solution from DITAN

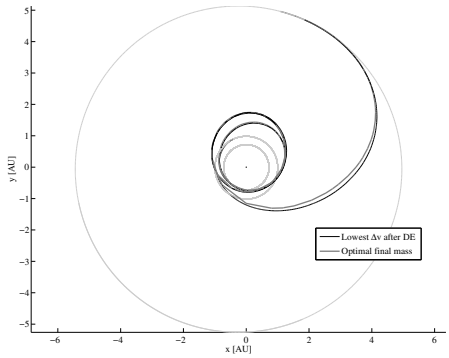

Fig. 9 Trajectory plo of the result from DE with the lowest Δv and its optimized solution from DITAN

CONCLUSION

A method to construct and globally optimize low thrust multiple gravity assist trajectories was presented in this paper. The advantage is that no powered swingby is necessary. A new incremental pruning scheme is also proposed in order to reduce the search space of the global optimization. The numerical results are promising and will be further studied and analyzed for further improvement of the algorithms.

ACKNOWLEDGMENTS

The work presented in this paper was partially supported by Qinetiq through an ICASE EPSRC grant.

REFERENCES

[1] Rayman, M.D., Varghese, P., Lehman, D.H. and Livesay, L.L.: Results from the Deep Space 1 Technology Validation mission, *Acta Astronautica*, Issues 2-9, July-November 2000, pp. 475-487

[2] Haros, D. and Schoenmakers, J.:Post-Launch Optimisation of the Smart 1 Low-Thrust Trajectory to the Moon, *18th International Symposium on Spaceflight Dynamics*, Munich, Germany, 15th-22nd Oct. 2004

[3] Rinderle, E. A.: Galileo User's Guide, Mission Design System, Satellite Tour Analysis and Design Subsystem, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, JPL D-263, 1986

[4] Williams, S.N. and Longuski, J.M.: Automated design of multiple encounter gravity-assist trajectories, Master's Thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 1990

[5] Petropoulos, A.E. and Longuski, J.M. and Bonfiglio, E.P.: Trajectories to Jupiter via gravity assists from Venus, Earth, and Mars, *Journal of Spacecraft and Rockets*, Vol. 17, No. 6, November-December 2000, pp. 776-783

[6] Petropoulos, A. E. and Longuski, J. M.: Automated design of low-thrust gravity-assist trajectories, AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper, Vol. 4033, pp.157-166, 2000

[7] Izzo, D., Becerra, V. M., Myatt, D. R., Nasuto, S. J. and Bishop, J. M.: Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories, *Journal of Global Optimization*, Springer, Vol. 38, No. 2, 2007, pp. 283-296

[8] De Pascale, P. and Vasile, M. : Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories, *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006, pp. 1065

[9] Yam, C. H., Di Lorenzo, D. and Izzo, D.: Constrained Global Optimization of Low-Thrust Interplanetary Trajectories, ESA internal

[10] Carnelli, I., Dachwald, B. and Vasile, M.: Evolutionary Neurocontrol: A Novel Method for Low-Thrust Gravity-Assist Trajectory Optimization, *Journal of guidance, control, and dynamics*, Vol. 32, No. 2, March-April 2009

[11] Petropoulos, A.E. and Longuski, J.M.: Shape-Based Algolrithm for the Automated Design of Low-Thrust, Gravity-Assist Trajectories, *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, September-October 2004, pp. 787-796

[12] Vasile, M., Bernelli-Zazzera F.: Optimizing Low-Thrust and Gravity Assist Maneuvres to Design Interplanetary Trajectories, *The Journal of the Astronautical Sciences*. Vol. 51, No. 1, January-March 2003

[13] Novak, D. M., Vasile, M.: An Improved Approach to Preliminary Mission Design Using Fast Linear Quadratic Intermediate Optimisations, Proceedings of the 60th International Astronautical Congress, IAC09-C1.10.5, Daejeon, Korea, 2009

[14] Becerra, V. M., Myatt, D. R., Nasuto, S. J., Bishop, J. M. and Izzo, D.: An efficient pruning technique for the global optimisation of multiple gravity assist trajectories, *Acta Futura*, Vol. 2005, p. 35, 2003

[15] Vasile, M., Schuetze, O., Junge, O., Radice, G., Dellnitz, M., and Izzo, D.: Spiral trajectories in global optimisation of interplanetary and orbital transfers, Technical report, Ariadna Study Report AO4919 05/4106, Contract Number 19699/NL/HE, European Space Agency, 2006

[16] Novak, D. M., Vasile, M.: An Improved Shaping Approach to the Preliminary Design of Low-thrust Trajectories, submitted to the *Journal of Guidance, Control and Dynamics* in May 2010.

[17] Battin, R. H., *An introduction to the mathematics and methods of astrodynamics*, AIAA Education Series, 1999, pp. 490-493

[18] Efroimsky, M.: Gauge Freedom in Astrodynamics, *Modern Astrodynamics*, edited by P. Gurfil, Butterworth-Heinemann, 2006, pp. 23-52.

[19] Vasile, M., De Pascale, P. and Casotto, S. : On the optimality of a shape-based approach based on pseudo-equinoctial elements, *Acta Astronautica*, Volume 61, Issues 1-6, June-August 2007, pp. 286-297

[20] Izzo, D.: Advances in global optimisation for space trajectory design, Proceedings of the Interbational Symposium on Space Technology and Science, Vol. 25, p. 563, 2006

[21] Schuetze, O., Vasile, M., Junge, O., Dellnitz, M., and Izzo, D.: Designing optimal low-thrust gravity-assist trajectories using space pruning and a multi-objective approach, *Engineering Optimization*, Vol. 41, No. 2, 2009