DESIGNING OPTIMAL MULTI-GRAVITY-ASSIST TRAJECTORIES WITH FREE NUMBER OF IMPULSES

Joris T. Olympio$^{(1)}$

$^{(1)}$ *ESA, Advanced Concepts Team, ESTEC, Keplerlaan 1, Postbus 299, 2200 AG, Noordwijk*
E-mail address: joris.olympio@esa.int

Abstract

The problem of optimising multi-gravity assist trajectories with deep space manoeuvres is considered. A new algorithm is presented to tackle the optimisation of the number of impulses as well as their respective locations. The new method uses the primer vector theory, which is extended to multi-gravity assist trajectories using a variational approach. Using the interaction prediction principle, the multi-gravity assist problem is then decomposed into sub-problems by introducing and relaxing boundary conditions. All these sub-problems can then be easily solved in parallel. The algorithm is illustrated on a simple Earth-Venus-Mars transfer problem, and more complicated problems of transfer to Mercury and Saturn with multiple gravity assists.

INTRODUCTION

The design of interplanetary trajectories using gravity assists is a challenging problem. Gravity assists are often used by mission designers to reduce the fuel consumption of a mission, but also to satisfy specific mission constraints such as small rendezvous manoeuvre. The resulting optimisation problem can be quite complex. With an impulsive thrust model, the number of impulses (Deep Space Manoeuvres) per leg of the overall trajectory has also to be optimised to reduce both the propellant consumption and the launch and arrival conditions. If the number of impulses is too low, the objective function value is not as low as possible, which can result in an expensive trajectory and thus very interesting launch opportunities might not be considered.

A popular approach to optimise this type of problem is to fix the number of impulses, with often one impulse maximum per leg, and then to optimise their locations[1][2]. However, this approach is not optimal, and it leads to sub-optimal solutions as a different number of DSM might be required. Another approach is to reinterpret the result of the optimisation. Using an appropriate numerical formulation[3], the numerical method can indicate whether we need or not DSM by forcing the velocity change of unused DSM to 0. For instance, if a trajectory can have an optimal number of DSM varying between 0 and n, n DSM must be coded, and 0-ΔV DSM would be re-interpreted as no DSM. This approach is however not efficient because the number of optimisation variables has to be important to be sure that the number of DSM is actually not constrained.

Some past works partially addressed this issue. In [4, 5, 6], the primer vector theory is introduced for a single leg transfer to minimise the characteristic velocity (sum of impulses). In [7], the authors propose an algorithm for optimising N-impulse transfers, using the primer vector theory. In both papers, the number of impulses is free, but no gravity assist is allowed. In [8], an attempt to optimise multi-gravity assist trajectories is done using the primer vector theory. The author considers an overall transition matrix to calculate the complete primer vector history along the multi-gravity assist trajectory. A grid search procedure is then used to optimise the trajectory with powered swing-bys. Obviously, the search space dimension becomes very important for long trajectory sequences.

Deep space manoeuvres have also been considered in a number of recent papers and studies dealing with global optimisation strategies[1]. In [9], the authors use a genetic algorithm to optimise multiple impulse transfers between orbits (results with no more than one DSM are presented). In [10], the authors investigate impulsive trajectories using heuristic algorithms for a fixed DSM structure.

In this work multi-gravity assist trajectories are computed with a free number of deep space manoeuvres (MGA DSM)[11]. The approach is based on the primer vector theory, to find the optimal number of DSM on each leg of a multiple gravity assist trajectory and optimise at the same time their locations in space and time.

PROBLEM STATEMENT

1. PROBLEM DESCRIPTION

Consider the problem of transferring a spacecraft from an initial orbital state, defined by the position and velocity $[\mathbf{r_0}, \mathbf{v_0}]$ in a given frame, to a final orbital state $[\mathbf{r_f}, \mathbf{v_f}]$ in a given time of flight $T = t_f - t_0$. The dynamics are described by:

$$f(\mathbf{X},\mathbf{u},q;t)=\frac{d}{dt}\begin{bmatrix}\mathbf{r}\\ \mathbf{v}\\ m\end{bmatrix}=\begin{bmatrix}\mathbf{v}\\ -\frac{\mu}{r^3}\mathbf{r}+\frac{V_e}{m}q\mathbf{u}\\ -q\end{bmatrix} \tag{1}$$

Constants μ and V_e define respectively the central body gravitational constant (e.g. the Sun) and the exhaust velocity of the chemical thruster. Variable m is the mass of the spacecraft, which evolves with the mass flow rate q when the spacecraft is thrusting. The mass flow rate is physically bounded: $0 < q < q_{max}$. Vector $\mathbf{u}$ defines the thrust direction.

The control variables are thus constrained with:

$$q(q_{max}-q)\geq 0 \tag{2}$$

$$\mathbf{u}^T\mathbf{u}=1 \tag{3}$$

To reduce the cost of a transfer, it is sometimes useful to use the dynamics of the planets. Under specific conditions a spacecraft can modify its velocity by exchanging momentum with a massive body[12][13] (e.g. planet), resulting in a free manoeuvre also called gravity assist or swing-by.

In the present study, intermediate non-powered gravity assists are considered, where no thrust is necessary during the planetary encounter. In addition, the patched-conic approximation[12] is used. This means that the gravitational influence of the planets is not directly taken into account in the dynamics, and the gravity assist is modelled as an instantaneous change in the velocity vector. The position and velocity of the planet are used to formulate intermediate constraints. Noting t_p the date of the instantaneous manoeuvre at the date of encounter with the planet (p), the trajectory constraint at the gravity assist is then:

$$\psi(\mathbf{r},\mathbf{v};t_p)=\begin{bmatrix}\psi_{R1}\\ \psi_{R2}\\ \psi_{V_\infty}\end{bmatrix}=\begin{bmatrix}\mathbf{r}(t_p^+)-\mathbf{r}_p(t_p^+)\\ \mathbf{r}(t_p^+)-\mathbf{r}(t_p^-)\\ V_\infty^- - V_\infty^+\end{bmatrix} \tag{4}$$

The first two constraints express the continuity of the position vector and the encounter with the planet. The last constraint ensures the energetic conservation of the non-powered gravity assist: the incoming and outgoing hyperbolic excess velocities are equal in module. Hyperbolic excess velocities are defined as the difference between the gravity assist planet velocity $\mathbf{V}_p$ and the spacecraft velocity immediately before and after the gravity assist manoeuvre: $\mathbf{V}_\infty^- = \mathbf{v}(t_p^-)-\mathbf{V}_p(t_p)$ and $\mathbf{V}_\infty^+ = \mathbf{v}(t_p^+)-\mathbf{V}_p(t_p)$.

In the patched-conic approximation, the velocity vector discontinuity for a non-powered gravity assist, owing to the altitude of the periapse, is given by:

$$\psi_d(\mathbf{v},r_p;t_p)=\frac{\mathbf{V}_\infty^{-T}\mathbf{V}_\infty^+}{\left\|\mathbf{V}_\infty^-\right\|^2}-\cos(\beta(r_p,V_\infty)) \tag{5}$$

The deviation angle β is a function of the hyperbolic excess velocities V_∞ and the gravity assist hyperbolae periapse radius r_p for a non-powered gravity assist[12]. The hyperbolae periapse radius r_p must be higher than the planet radius for the gravity assist to be considered feasible.

The periapse radius r_p is usually a variable of the optimisation problem. In the present study, it will not be considered, although, the constraint Eq. 5 shall be used at the end of the optimisation process to provide a solution with feasible gravity assists.

The objective function is to minimise the total propellant consumption. The constraint on the control and the type of thruster considered (reasonably $q_{max}\to\infty$ for chemical thrusters) result in impulsive manoeuvres. Thus, minimizing the propellant consumption is equivalent to minimise the total characteristic velocity (Toltoiski) which is the sum of the launch manoeuvre $\Delta\mathbf{V}_0$, the rendezvous manoeuvre $\Delta\mathbf{V}_f$ (braking manoeuvre) and any $n\geq 0$ intermediate impulses $\Delta\mathbf{V}_i$ (n being an unknown of the problem). Impulsive velocity changes $\Delta\mathbf{V}_i=\mathbf{v}(t_i^+)-\mathbf{v}(t_i^-)$ are applied along the trajectory at unknown dates t_i.

The objective function is thus:

$$J=\|\mathbf{\Delta V_0}\|+\|\mathbf{\Delta V_f}\|+\sum_{i=1}^{n}\|\mathbf{\Delta V_i}\| \tag{6}$$

Where $\Delta\mathbf{V}_0=\mathbf{V}(t_0)-\mathbf{v}_p(t_0)$, and $\Delta\mathbf{V}_f=\mathbf{v_p}(t_f)-\mathbf{V}(t_f)$.

In the next sections, the primer vector theory is introduced and extended to MGA trajectories. This allows optimising the MGADSM problem and find the optimal number of impulses. A specific algorithm shall be introduced to solve efficiently the problem. Then, the theory is applied to a simple case considering only one single gravity assist. Eventually, the developments discussed in this paper are illustrated with more complex examples.

EXTENSION OF THE PRIMER VECTOR THEORY TO MGA TRANSFERS

1. OPTIMAL CONTROL PROBLEM

To address the optimal control problem (OCP), a co-state vector $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_R, \boldsymbol{\lambda}_V]$ is assigned to the spacecraft state vector $\mathbf{X} = [\mathbf{r}, \mathbf{v}]$. The primer vector, noted $\boldsymbol{\lambda}_\mathbf{V}$, is the co-state vector of the velocity $\mathbf{v}$. Because of the impulses and the minimisation of the characteristic velocity, the mass does not need to be included in the formulation, nor its dual variable.

Using Eqs. (1), (4), (5) and (6), for one single gravity assist, the Lagrangian merit function of the OCP is given by:

$$\begin{aligned} L(\mathbf{z}, \mathbf{X}, \boldsymbol{\Lambda}, \boldsymbol{\nu}) = J(\mathbf{z}) & \\ & + \int_{t_0}^{t_f} \boldsymbol{\Lambda}^T F(\mathbf{X}, \mathbf{u}; t) dt \\ & + \boldsymbol{\nu}_{R1}^T \psi_{R1} + \boldsymbol{\nu}_{R2}^T \psi_{R2} + \nu_{V_\infty}^T \psi_{V_\infty} \end{aligned} \tag{7}$$

With:

$$F(\mathbf{X}, \mathbf{u}; t) = \frac{d\mathbf{X}}{dt} - f(\mathbf{X}, \mathbf{u}; t) \tag{8}$$

The variables $\boldsymbol{\nu}_{R_1}$, $\boldsymbol{\nu}_{R_2}$ and ν_{V_∞} are the constant Lagrange multipliers associated to the constraints (Eq. 4). At this stage the constraint (Eq. 5) is not used.

2. BOUNDARY AND TRANSVERSALITY CONDITIONS AT SWING-BY

The Primer vector theory is a direct result of calculus of variations theory. By computing appropriate transversality conditions, for their respective constraints, the theory can be extended to multi-gravity assist trajectories (MGA). The transversality conditions provide the boundary conditions to compute the primer vector history $\lambda(t)$ along a multiple leg trajectory and across non-powered instantaneous swing-bys and DSM, in order to assess the optimality of a MGA DSM trajectory.

The necessary conditions of optimality are given by the stationary condition on the Lagrangian. Each coefficient of the differentiated variables must vanish. For instance, the dynamics of adjoint equations, or Euler-Lagrange equations, can be written:

$$\frac{d\boldsymbol{\Lambda}}{dt} = -\begin{bmatrix} 0 & \mathbf{G} \\ I & 0 \end{bmatrix} \boldsymbol{\Lambda} \tag{9}$$

with I the 3x3 identity matrix and $\mathbf{G}(\mathbf{r}; t)$ the gravity gradient matrix[12, 6].

In addition, the stationarity condition gives boundary conditions, and transversality conditions at the date of the swing-by date. Using Eq. 4, these boundary conditions can be written as jump conditions at an un-powered swing-by:

$$\boldsymbol{\lambda}_V(t_p^+) = -\frac{\mathbf{V}_\infty^+}{\|\mathbf{V}_\infty^+\|} \|\boldsymbol{\lambda}_\mathbf{V}(\mathbf{t_p^-})\| \tag{10}$$

$$\boldsymbol{\lambda}_R(t_p^+) = \boldsymbol{\lambda}_R(t_p^-) - \boldsymbol{\nu}_{R_2} \tag{11}$$

3. CONDITIONS FOR A PRIMER VECTOR OPTIMAL TRAJECTORY

3.1 without swing-bys

Primer vector theory provides a mean to evaluate the optimality of impulsive trajectories. The theory gives four necessary conditions[4, 5, 14, 15] for a ballistic trajectory to be optimal, which can be summarised as follow:

1. The primer vector and its derivative are continuous.
2. If there is an impulse, the primer vector is aligned with the impulse, and its module is 1.
3. The primer vector module should not exceed 1.
4. The derivative of the primer vector module at all impulses is zero.

Indeed, the primer vector history dictates that to reduce the cost an impulse must be added or modified. The primer vector history $\lambda_V(t)$ has thus to be computed for the current impulsive trajectory. If the derivative $\dot{\lambda_V}(t)$ is zero at the time of the impulses, the impulses are optimal. In addition, if $\lambda_V(t)$ does not exceed 1, the trajectory is indeed optimal. On the contrary, had the primer vector history $\lambda_V(t)$ exceeds one at any time on the trajectory, the trajectory is not optimal and DSM must be added. As a first guess estimation[4], the new impulse can be added at the date when $\lambda_V(t)$ is maximum and above one, and at the place that corresponds to the unperturbed spacecraft position at this estimated date.

3.2 with swing-bys

Equations (10) and (11) and the above developments provide 3 additional conditions for a MGADSM trajectory to be optimal at a swing-by:

1. The primer vector module is continuous for a non powered swing-by. (Eq. (10)).
2. The primer vector derivative is not continuous, but a jump condition can be provided. (Eq. (11)).
3. If the swing-by is free, the primer vector module verifies $\|\mathbf{bm}\lambda_\mathbf{V}\| < 1$. (Eq. (10) and III.A.3)

These conditions together with the primer vector theory allow constructing a Lawden optimal multi-gravity assist trajectory with deep space manoeuvres (MGA DSM) and non-powered gravity assists.

PARALLEL ALGORITHM

1. ABOUT THE OPTIMAL CONTROL PROBLEM

The primer vector theory gives information on the impulses to add such as a first order estimation of their place[4], but it does not provide any information on the amplitude of the impulses to add. Because the thrust is unbounded, the impulses can have any amplitude and the variational equations do not permit to guess them easily. Consequently, it is not appropriate to solve the OCP directly using the boundary conditions and formulating a Two Point Boundary Value Problem (TPBVP).

As an alternative, the equivalent parameter optimal problem (POP) is solved, where the number of impulses is given by the primer vector analysis, and the control $\mathbf{u}$ is traded with a decision vector $\mathbf{z}$ describing the impulses (the description of $\mathbf{z}$ will be explained later). The purpose of introducing the POP is indeed to find the amplitude of the impulses. With a solution $\mathbf{z}^*$ of the POP, the primer vector history can be computed to see if it is equivalent to an optimal solution $\mathbf{u}^*$ of the OCP. The optimisation procedure is started again with a new POP as long as the primer vector theory dictates the need of additional impulses to get an optimal solution $\mathbf{u}^*$.

Note that the OCP and the POP are different in nature. With the OCP, the number of impulse is free and eventually optimal, while when solving the POP the number of impulses is constrained. Consequently, an optimal solution of the POP is an optimal solution of the OCP only if the number of impulses in the POP is optimal.

2. PROBLEM

It is difficult to solve the TPBVP (or a Multi Point Boundary Value Problems) (Eqs. 4, 6, 7, 10, 11) for MGADSM problems with usual methods. The reasons are threefold. First, the method would be very sensitive as any slight changes on the first leg modify the input conditions of each subsequent swing-by and leg, and eventually change the overall MGADSM trajectory cost by a huge amount. Second, any addition or modifications of an impulse need the computation of the primer vector history and state transition matrices over the entire trajectory, which is computationally expensive. Third, as explained earlier, as we have no information of the amplitude of the impulses the implementation is difficult.

Our approach tackles these issues. We use the interaction prediction principle [16, 17] that is based on a decomposition principle and a coordination strategy. The method performs the problem decomposition by breaking the original trajectory at points of swing-by (interfaces) into separate legs. Each leg is then independently optimised with a POP using specific boundary conditions on the state and the co-state variables (see Eqs. 4, 10, 11). Lastly, a coordination strategy update the interfaces according to an optimisation policy.

This method has been widely used for large scale problems [18] and distributed environments. A similar approach has been used on low-thrust interplanetary transfer problems[19].

The general method follows 3 steps:

1. Splitting and formulating the original problem into sub-problems of lesser size.
2. Optimizing each sub-problem individually.
3. Setting up exchange processes between the sub problems in order to satisfy the original problem optimality conditions.

The trajectory decomposition permits to remove the sensibility of the problem by solving simpler sub problems. The Euler-Lagrange equations are integrated on smaller time intervals thus reducing the sensibility to small changes over the legs. One of the major advantages of the method is also its ability to be easily parallelised.

3. SOLUTION METHOD

3.1 Optimisation Problem Formulation

Each body to body transfer problem is considered as a sub-problem. As stated earlier, we solve a POP. Thus, for a body to body transfer, the leg is described by the initial and final positions, a time of flight, and includes m DSMs. We choose to describe the DSM with 4 variables, which are the Cartesian position and the date of the impulse. The decision vector used once the number m of deep space manoeuvres is known is:

$$\mathbf{z} = [t_1, \mathbf{r}_1, ..., t_n, \mathbf{r}_n] \tag{12}$$

with n the total number of DSM. To avoid overlapping DSM in time, variables t_i actually describe the time duration between two successive events (impulses or swing-bys). Thus: $t_i > 0$ and $\sum_i t_i < t_f - t_0$.

Describing impulses with their positions in space is appealing as the Lambert's problem solver can be used to construct the entire trajectory. The Lambert's problems are restricted to be less than a revolution to ensure the existence and uniqueness of the solution found [12]. This restriction does not limit the number of DSM-leg revolution, and encourage the solver to add revolutions when increasing the number of DSM.

The problem to solve can be formulated as follow:

$$(P0) \quad \begin{cases} \min\limits_{\mathbf{z}} & J(\mathbf{z}) \\ \quad s.t. & \psi(\mathbf{r}, \mathbf{v}; t_p) = 0 \end{cases} \tag{13}$$

For simplicity, we explain the process of solving a multi-DSM transfer problem P0 with 2 phases (i.e. 1 gravity assist). The launch date t_0, the arrival date t_f and the planet encounter date t_p define the scenario. Integers m_1 and m_2 are respectively the optimal number of DSM for the 1st and 2nd phase.

3.2 Interaction Prediction Principle and Decomposition

The decomposition is done at each intermediate body (where a swing-by occurs), and the decision vector can be written.

$$\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2] \tag{14}$$

Where $\mathbf{z}_k$ is the decision vector of the k^{th} independent sub-problem (or phase).

$$\mathbf{z}_i = [t_1, \mathbf{r}_1, ..., t_{m_i}, \mathbf{r}_{m_i}] \tag{15}$$

The objective function is separable because of the non-powered swing-by assumption and the constraint formulation. Defining the cost of each phase $i \in \{1, 2\}$ with the value function:

$$J_i(\mathbf{z_i}) = \sum_{j=1}^{m_i} \|\mathbf{\Delta V_{j,i}}\| \tag{16}$$

where the impulse $\mathbf{\Delta V_{j,i}}$ stands for the j^{th} impulse of the i^{th} phase.

Using the interaction prediction principle[17], the 1st and 2nd sub problems are respectively:

$$(SP1) \quad \begin{cases} \min\limits_{\mathbf{z}_1} & \Delta V_0 + J_1(\mathbf{z}_1) \\ \quad s.t. & V_\infty^- - c_\infty = 0 \end{cases} \tag{17}$$

$$(SP2) \quad \begin{cases} \min\limits_{\mathbf{z}_2} & J_2(\mathbf{z}_2) + \Delta V_f - c_\nu V_\infty^+ \end{cases} \tag{18}$$

Note, that as m_i is the optimal number of impulses (in Eq. 16), several minimisation of these problems along with computations of the primer vector history might be required.

To solve the sub-problems independently while optimising the overall problem P0, the coordination variable c_ν has been introduced. A scalar Lagrange variable ν_1 shall be assigned to the constraint of problem SP1.

Each coordination variable represents the missing information that each sub-problem requires to comply with the overall problem optimality conditions. For instance, the 2nd sub-problem uses a coordination variable $\mathbf{c}_\nu$ associated to Lagrange variable ν_1. The 1st sub-problem uses the coordination variables $\mathbf{c}_\infty$ associated to the outgoing hyperbolic excess velocity. Both coordination variables account for the coupling relations resulting from the swing-by conditions.

Obviously, solutions of sub-problem SP1 (Eq. 17) and of sub-problem SP2 (Eq. 18) are not solutions of the minimum characteristic velocity problem described by the objective functions on Eq. 16. They however describes the solution of the overall problem P0 (Eq. 13).

Using Eq. 7 and comparing the OCP with the POP for the initial sub-problem at $t = t_0$ gives:

$$\lambda_V(t_0) = \frac{\mathbf{\Delta V}_0}{\|\mathbf{\Delta V_0}\|} \tag{19}$$

$$\lambda_V(t_p^-) = -\nu_1 \frac{\mathbf{V}_\infty^-}{\|\mathbf{V}_\infty^-\|} \tag{20}$$

For the final sub-problem:

$$\lambda_V(t_p^+) = c_\nu \frac{\mathbf{V}_\infty^+}{\|\mathbf{V}_\infty^+\|} \tag{21}$$

$$\lambda_V(t_f) = \frac{\mathbf{\Delta V_f}}{\|\mathbf{\Delta V_f}\|} \tag{22}$$

The boundary conditions allow actually communicating between the sub-problems. At each iteration we get new boundary conditions for each leg, which provide a new DSM structure. Eventually, when a stationary point is found, we have reached an optimum of the overall problem.

3.3 Coordination Strategy

Considering an iterative algorithm, at each iteration i of the algorithm the coordination step updates the coordination variables c_∞ and c_ν.

Considering the sub-problems SP1 and SP2, and the overall problem P0, the Lagrangian can be written:

$$\mathcal{L}(\mathbf{z}_1, \mathbf{z}_2, \nu_1, c_\infty, c_\nu) = \Delta V_0 + J_1(\mathbf{z}_1) + J_2(\mathbf{z}_2) + \Delta V_f + \nu_1 \left(V_\infty^- - c_\infty\right) + c_\nu \left(c_\infty - V_\infty^+\right) \tag{23}$$

This Lagrangian should not be mistaken by the one described in Eq. 7, which describes the OCP, as both Lagrangians describe different problem formulations.

From the stationary conditions of the Lagrangian (Eq. 23), we have an implicit Arrow-Hurwicz algorithm[20] given by:

$$c_\infty^{i+1} = c_\infty^i + \frac{\epsilon_\psi}{1 + \epsilon_\nu \epsilon_\psi} \left(\epsilon_\nu \left(\|\mathbf{V}_\infty^+\| - c_\infty^i\right) + \left(\nu_1 - c_\nu^i\right)\right) \tag{24}$$

$$c_\nu^{i+1} = c_\nu^i + \frac{\epsilon_\nu}{1 + \epsilon_\nu \epsilon_\psi} \left(\left(c_\infty^i - \|\mathbf{V}_\infty^+\|\right) + \epsilon_\psi \left(\nu_1 - c_\nu^i\right)\right) \tag{25}$$

where we can use:

$$c_\infty^0 = \|\mathbf{V}_\infty^-\| \quad and \quad c_\nu^0 = 1$$

The coordination variables $\{c_\nu, \mathbf{c}_\infty\}$ and the boundary states $\{\nu_1, \mathbf{V}_\infty^-\}$ allow respecting the swing-by constraint, and force the sub-problems solutions toward the overall problem solution. The coordination equations show indeed that c_ν is driving the coordination variables to their optimum value.

Scalars ϵ_ν and ϵ_ψ are relaxation variables that control the convergence.

According to Eqs. 20 and 21, and the extended primer vector theory, there is the condition on c_ν for $\|\lambda_\mathbf{V}\| \leq 1$. If this condition is not respected, the solution should be either to add impulses for changing the hyperbolic excess velocity at the swing-by, or to change the swing-by date.

Interestingly, adding impulses permits to lower the Lagrange variable module $\|\nu\|$. This can be explained by the Hamilton-Jacobi theory [21]. Lagrange variables represent the sensitivity of the constraint with respect to the decision vector. Thus, the more impulses there are, the more robust is the trajectory. This can also be interpreted considering that the final constraint relies on many intermediate impulses with small contributions, rather than on few impulses with high contribution.

3.4 General Parallel Algorithm

The overall algorithm is briefly described in table 1. After the initialisation step, sub-problems SP1 and SP2 (Eqs. 17 and 18) are solved in parallel. The convergence of each sub-problem is very important to the general behavior of the algorithm, and each problem optimisation must return valuable Lagrange multipliers and cost.

Because of the low dimensionality of the sub-problems, it is a good idea to first use a global optimising method to get, or improve, the initial guess. This has the advantage, over getting a good initial point for the DSM, to considerably reduce the influence of the initial guess for the coordination variables c_ν. In addition, if one can find the global minimiser of each sub-problem, it is very likely that the overall solution is close to the global optimum of the overall problem, if the problem is convex. Obviously, to account for the primer vector theory and the constraints, a local optimiser is always necessary to get the Lagrange coefficients. Because the constraints are not implicit when evaluating the objective function, this approach is often referred to the Infeasible Iterative Approach[15].

A Broyden-Fletcher-Goldfard-Shanno (BFGS) algorithm[22, 14] is then used to solve the POP problem, when no constraints are provided (e.g. sub-problem SP2). An additional solver[23, 24] is used when there is a constraint. It is very unlikely that the sub-problems are infeasible, indeed, owing to the nature of the problem. It is always possible to find an impulse that would help satisfy the constraints. But, to increase the convergence, all the gradients are provided analytically, in terms of the primer vector derivatives and state transition matrices. In addition, DSM are optimised one at a time when the theory dictates that more than one impulse is needed.

Step 2 compares the new computed solution to previous best one. The cost that is compared is the one defined by Eq. 6 including discrepancies at swing-bys (post-swing-by correction manoeuvres). The cost then always refers to a consistent trajectory.

In Step 3 the relaxation variables ϵ_ϕ and ϵ_ν are used to control the convergence of the fixed-point based coordination step, in case the iteration does not result in an improved solution. With sufficiently small updates on the coordination, the convergence of the algorithm to a fixed point should almost always happen.

However, in practice, as the problem is not strongly convex, Step 5 patches the sub-problems solutions, constructs an initial guess for the overall problem and optimises it. Indeed, the sub-problems solutions should to be "near" the basin of attraction of the overall problem, and this step could ensure the satisfaction of second order optimality conditions for a minimum.

Convergence of the algorithm is reached when updates on the coordination variables are smaller than a given tolerance, or when the algorithm does not manage to reduce the cost anymore.

3.5 Discussion and Critical Cases

As the primer vector theory predicts, it is not always necessary to insert DSM on a leg. If the first or any intermediate legs do not need DSM, the Lambert's problem solution imposes the initial and final hyperbolic excess velocity $\mathbf{V}_\infty^-$ and $\mathbf{V}_\infty^+$ for each leg, as the dates are fixed. Consequently, as there is no DSM on the affected leg, there is no variable to satisfy the free swing-by constraint as in Eq. 4 and thus it is not possible, with any solver, to get a Lagrange variable as it is required by Eqs. 20 and 21. For instance, in the 2 leg problem, 2 cases can appear.

Case 1. The first leg does not need any DSM. The overall cost only depends on the last leg, and Eq. 18 corresponds to the overall cost if $c_\nu = -1$. In this case, a post swing-by manoeuvre is considered with an impulse (that should be included in Eq. 16) during the swing-by, as in Eq. 26:

$$\mathbf{\Delta V} = \left(\mathbf{V}_p(t_p) - \mathbf{V}^+(t_p)\right)\left(1 - \frac{V_\infty^-}{\|\mathbf{V_p(t_p)} - \mathbf{V^+(t_p)}\|}\right) \tag{26}$$

Obviously, if the post swing-by manoeuvre is null, c_ν can take practically any value in the range $]-1, 1[$, but the algorithm finds the proper one.

Case 2. The second leg does not need any impulse. In this case, problem SP1 (Eq. 17) corresponds to the overall cost if $c_\infty = V_\infty^+$. The gravity assist is free, and the Lagrange variable is not needed.

These results are easily extended to multi-leg problems, bearing in mind that the cost is additive. In this case an intermediate leg would be the combination of sub-problems SP1 and SP2, and thus would have an extended objective function and a terminal constraint. The coordination strategy would be exactly the same as it applies to swing-bys.

Table 1: Parallel Algorithm for the n-Gravity Assist Trajectory Problem

```
Step 0.  Initialization of coordination variables: Choose $\{c^0_{\nu,k}, \mathbf{c}^0_{\infty,k}\}$ for all $k \in \{1, ..., n\}$
         Compute the trajectory cost without DSM: $J^0$
         Store boundary states $S^0 = \{V_\infty, \mu_1\}$
         Set $\epsilon^1_\phi = \epsilon^0_\phi$, $\epsilon^1_\nu = \epsilon^0_\nu$ and $i = 1$

Step 1.  Resolution in parallel of the problems
           the initial problem with $\mathbf{c}^i_{\infty,1}$ to get $\mathbf{z}_1$.
           intermediate problems $k$ with $\{c^i_{\nu,k}, \mathbf{c}^i_{\infty,k}\}$ to get $\mathbf{z}_k$.
           the final problem with $\{c^i_{\nu,n}, \mathbf{c}^i_{\infty,n}\}$ to get $\mathbf{z}_n$.
         Compute the cost of the new trajectory: $J^i$

Step 2.  If Step 1. provides an improvement on the overall cost ($J^i < J^{i-1}$), goto Step 4.
Step 3.  $\epsilon^i_\phi \leftarrow \epsilon^i_\phi/2$ and $\epsilon^i_\nu \leftarrow \epsilon^i_\nu/2$
         Update coordination variables $\{c^i_{\nu,k}, \mathbf{c}^i_{\infty,k}\}$ using former boundary state $S^{i-1}$
         Goto Step 1.

Step 4.  Update coordination variables $\{c^{i+1}_{\nu,k}, \mathbf{c}^{i+1}_{\infty,k}\}$.
         Store boundary states $S^i = \{V_\infty, \mu_1\}$
         Set $\epsilon^{i+1}_\phi = \epsilon^0_\phi$, $\epsilon^{i+1}_\nu = \epsilon^0_\nu$ and $i \leftarrow i + 1$

Step 5.  Resolution of global problem with $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n]$ as initial guess

Step 6.  Test of optimality
         Goto Step 1 if not optimal.
```

NUMERICAL EXAMPLES

1. A SIMPLE EVM TRANSFER

1.1 Illustration of the Algorithm

To illustrate the algorithm described above, a simple Earth-Venus-Mars problem (EVM) is proposed.

Figure 1 illustrates the algorithm behavior. For the sake of simplicity, we keep the DSM structure constant along the iterations to observe the convergence. The approach is obviously not efficient when the number of DSM is kept fixed. As can be seen, the coordination variables and the states at the swing-by converge to each other, while the cost is decreasing. This is a direct consequence of the coordination step. When the decomposition step failed to follow the solution and improve the cost, an adaptive coordination coefficient method is used to make smaller changes during the coordination (i.e. ϵ is reduced). To have the plain behaviour of the algorithm no global problem optimisation were performed, although, it is a good idea to optimise the overall problem around iteration 20 to get the optimal MGA DSM solution. In particular, we can observe, that at the end, the coordination variable c_ν and the Lagrange variable ν converge very slowly to each other, while c_∞ and v_∞ match. Because of this gap, the algorithm has difficulty to make good progress quickly.

It can then be admitted that the decomposition-coordination algorithm, if properly initialised, will result in decreasing iterations. Now using the primer vector solver at each step, i.e. not keeping the number of DSM constant, should result in the same trend. Indeed, the primer vector based solver returns at each iteration a better solution, so the overall cost decreases at each iteration.

Finally, a great advantage of the method is that problems of very small dimension are solved in parallel. Because the algorithm follows a solution path (i.e. it uses the previously converged solution) the solver converges quickly at each iteration. For a MGADSM problem, the algorithm is as fast and robust as solving a simple DSM transfer problem (supposing we have enough CPU resources).

1.2 Global Trade-off Study

The complexity of the problem is of the order of the complexity of one sub-problem. The Earth Venus Mars problem is of small dimension, and rapid results can be obtained regarding the global optimisation problem.

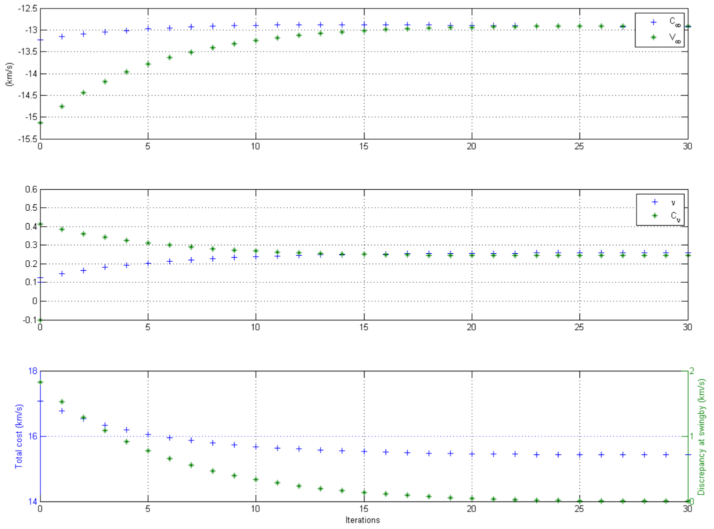

Figure 1: Iterations of the algorithms, for a fixed DSM structure.

Table 2: Trade-off configuration

	lower bound	upper bound
Earth Launch date T_0 (JD)	53164-365.25/2	53164+365.25/2
TOF of leg 1 (days)	150	350
TOF of leg 2 (days)	150	350
Number of random points	10000	

A global trade-off study is performed between the cost ΔV and the total time of flight tof, and the initial date and the time of flights are varied.

Opposite to global optimisation, we are seeking the optimums considering different options. The result can be described by a Pareto front. For any point on the Pareto front we cannot find a point that has a better cost and a better total time of flight. They are not dominated by any other points out of the Pareto front, and are optimal points. The approach is different from global optimisation, where most of the time only averaged good points can be sought because the time of flight cannot be optimised equally and at the same time as the cost without weightings in the objective function. A Pareto front allows to make optimal tradeoff decisions.

The code is run to solve a given number of problem with different time of flights and date of launch (see Table 2). The launch date T_0 and the times of flight are uniformly and randomly distributed. The uniformly generated random numbers make a parallel search very efficient to explore the state space. Corresponding problems are solved and costs ΔV are recorded. The Pareto points on the $Tof - \Delta V$ graph are the best points found for each time of flight that dominate the set. Then, the code accumulates enough data to span a wide range of possible mission opportunities, and identify the Pareto front. The result of the optimisations is plotted on figure 2. Figure 2 compares the nominal no-DSM trajectories with the optimised free-DSM, and clearly shows the algorithm always manages to improve the nominal trajectories. It also depicts the time of flight - cost Pareto front. All the solutions are plotted, even the non converged ones. Without further checks, it is although expected that all the solutions on the Pareto front are converged solutions. This shall demonstrate the major point of the approach: we do not need to take care of the control structure to assess the different options in the mission design process.

One solution is reported on Table 3. The trajectory is plotted on figure 3. The primer vector module history is displayed on figure 4, and shows that only one DSM is necessary. The value of the primer vector module at the date of the swing-by matches the value of the coordination variable c_ν.

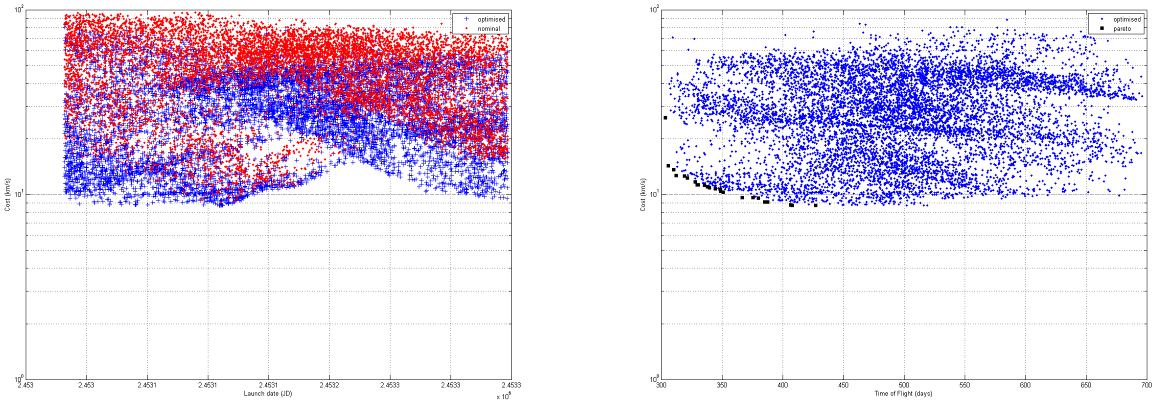

Figure 2: (left) Global optimisation of the Earth Venus Mars transfer, with free number of DSM. (right) Pareto analysis of the Earth Venus Mars transfer, with free number of DSM, for minimum time of flight and minimum propellant consumption.

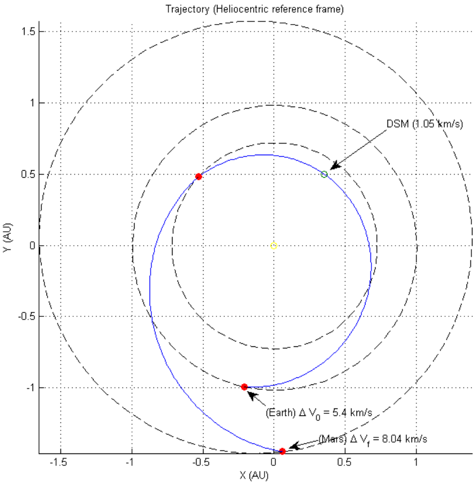

Figure 3: Example of an EVM optimal trajectory.

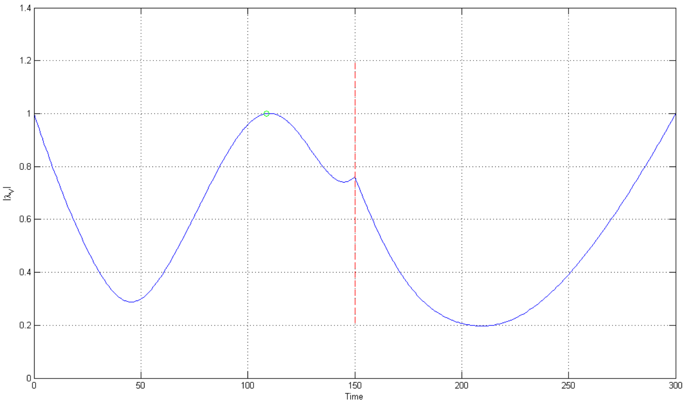

Figure 4: Example of a EVM Lawden optimal trajectory. The red vertical dash line is the swing-by location. The green dot is the DSM location.

Table 3: Algorithm Output

	without DSM	optimal DSM
Earth Launch date T_0	08/06/2004 (dd/mm/yyyy)	
Earth Departure $\|\mathbf{\Delta V_0}\|$	5.388 km/s	5.388 km/s
DSM date t_i	-	T_0 + 110.11 days
DSM impulse $\|\mathbf{\Delta V_i}\|$	-	1.055 km/s
Venus Swing-by date t_p	T_0 + 150 days	
Venus V_∞^-	8.854 km/s	6.946 km/s
Venus V_∞^+	6.946 km/s	6.946 km/s
EVM time of flight T	300 days	
Mars Rendezvous $\|\mathbf{\Delta V_f}\|$	8.042 km/s	8.042 km/s
Coordination variable c_∞	-	6.946 km/s
Coordination variable c_ν	-	-0.76199
Total cost J	15.024 km/s	14.486 km/s

Table 4: Best solutions reported for the ACT Trajectory Optimisation Problem database[1]

Problem Name	*Cassini 2*
Sequence	E V V E J S
Launch date T_0 (dd/mm/yyyy)	12/11/1997 (-779.63 MJD2000)
Time of Flights (days)	167.9, 424.0, 53.3, 589.8, 2200.0
Arrival date T_f	09/04/2007
Launch $\|\mathbf{\Delta V_0}\|$	3.265 km/s
DSM	472.8 m/s, 397.8 m/s, 0.0 m/s, 0 m/s, 0 m/s
Rendezvous $\|\mathbf{\Delta V_f}\|$	4.246 km/s
Final cost (km/s)	8.383 km/s
Problem Name	*Messenger easy*
Sequence	E E V V Y
Launch date T_0 (dd/mm/yyyy)	17/03/2003 (1171.0 MJD2000)
Time of Flights (days)	400, 178.4, 299.2, 180.5
Arrival date T_f	08/02/2006
Launch $\|\mathbf{\Delta V_0}\|$	1.409 km/s
DSM	910.8 m/s, 0.0 m/s, 263.3 m/s, 1415.4 m/s
Rendezvous $\|\mathbf{\Delta V_f}\|$	4.632 km/s
Final cost (km/s)	8.631 km/s

2. COMPARISON TO REFERENCE CASES IN THE GTOP DATABASE

Many MGADSM sequences have been studied and assessed in the past[1]. For the sake of the illustration of the algorithm, some sequences and results are compared to the output of the algorithm. This should demonstrate the performance of the algorithm for complex sequences.

The selected sequences, from the database[1] are described in table 4. Cassini-Huygens is a international mission that had been launched in 1997 to study the planet Saturn and its moons. MESSENGER is a NASA mission that will conduct an orbital study of the planet Mercury.

We took the best results found to date. These results were obtained with method based on global optimisation techniques. There is exactly one DSM on each leg, however, their amplitude can be zero as a result of the optimisation. Note again, that with the objective function Eq. 6 and analytical derivatives, it is not possible to find $0-\Delta V$ DSM because of the singularity that appears in the gradient. Most of the time, it is preferable to change the formulation[3], or to use finite differences[2] at the cost of having inaccurate optima.

The decomposition-coordination algorithm has been used to solve the problems described on table 4. The launch dates and the time of flight are kept constant. The algorithm only looked for the optimal DSM structure. To be consistent with the results of the GTOP database, the same ephemeris tool were used. If these solutions are locally optimal, in the sense of the OCP, the algorithm should find similar results.

Table 5: Optimal solutions

Problem Name	*Cassini 2*
Launch $\|\boldsymbol{\Delta}\mathbf{V_0}\|$	3.266 km/s
DSM Structure	1, 1, 0, 0, 0
DSM	469.780 m/s (first leg), 397.122 m/s (second leg)
Rendezvous $\|\boldsymbol{\Delta}\mathbf{V_f}\|$	4.246 km/s
Final cost (km/s)	8.387 km/s
Number of major iterations	6
Problem Name	*Messenger easy*
Launch $\|\boldsymbol{\Delta}\mathbf{V_0}\|$	1424.1 km/s
DSM Structure	2, 0, 1, 1
DSM	909.7 m/s, 16.6 m/s, 146.9 m/s, 1540.8 m/s
Rendezvous $\|\boldsymbol{\Delta}\mathbf{V_f}\|$	4.342 km/s
Final cost (km/s)	8.494 km/s
Number of major iterations	4

The results are reported on table 5. Figure 5 illustrates the results for the *Cassini 2* sequence, figure 6 illustrates the results for the *Messenger easy* sequence. On figure 6, the primer vector graph shows that the first three legs are optimal for the DSM structure to a given tolerance, although the last leg might need an additional DSM. Although, the algorithm concludes also that the last leg would need an impulse, the solver failed to converge. The issue might be due to the apparition of resonances and the inadequacy of the DSM model used. It is likely that a DSM description based on the date and the velocity provides a converged result.

Note that the cost of the sequence without DSM is more than 100 km/s. After one iteration, the algorithm manages to reduce the cost to 8.546 km/s with the same DSM sequence as for the solution reported in table 4 within only few seconds of computation. Obviously, this depends on the initialisation of the coordination variables. The method we describe in this study is a local optimisation method, and it is very unlikely it returns better results than global optimisation methods, unless these global optimisation methods disregard local optimality, which is generally not the case. Without too much analyses, the algorithm shows that for the same sequence and for the same dates, as in table 4 for the *Messenger easy* problem, a better solution exists. This might be because the solution has never been re-optimised properly with a local solver and using precise gradients[2]. During the next few iterations the algorithm changed the DSM structure. The final result is displayed on figure 6.

Although, the solution we found and reported is better than the one in table 4, we also found solutions with different DSM structures, multiple revolutions and lower rendezvous cost.

CONCLUSION

Using optimality conditions at swing by, the primer vector theory has been extended to multiple gravity assist trajectories. A method for simplifying the resolution of the free number of impulse problem has been proposed and a new algorithm has been provided. The algorithm proceeds in two steps. The first step decomposes the overall problem into sub-problems. All the sub-problems are solved in parallel. For each sub-problem, an optimum number of impulses is found using specific boundary conditions ensuring the validity of necessary condition for an optimum of the overall problem. The second step coordinates the sub-problems and optimises the overall problem with the optimum number of impulse of each phase.

The optimisation of multiple gravity assist with an optimal number of impulses under this algorithm proves to be robust and efficient. Solving the multiple gravity assist only depends on how well one can solve a direct transfer problem with multiple impulses.

It has been shown that global trade-off studies can be performed with a complexity, which only increases linearly with the number of phases. As the algorithm seeks automatically the optimal number of DSM, the intervention of the mission designer is reduced, and an assessment of the possible DSM structures is not necessary.

ACKNOWLEDGMENT

This work has been carried on as an ESA Ariadna study, during the prime author PhD studies. The prime author was sponsored by Thales Alenia Space (France), and the Centre National d'Etudes Spatiales (CNES,

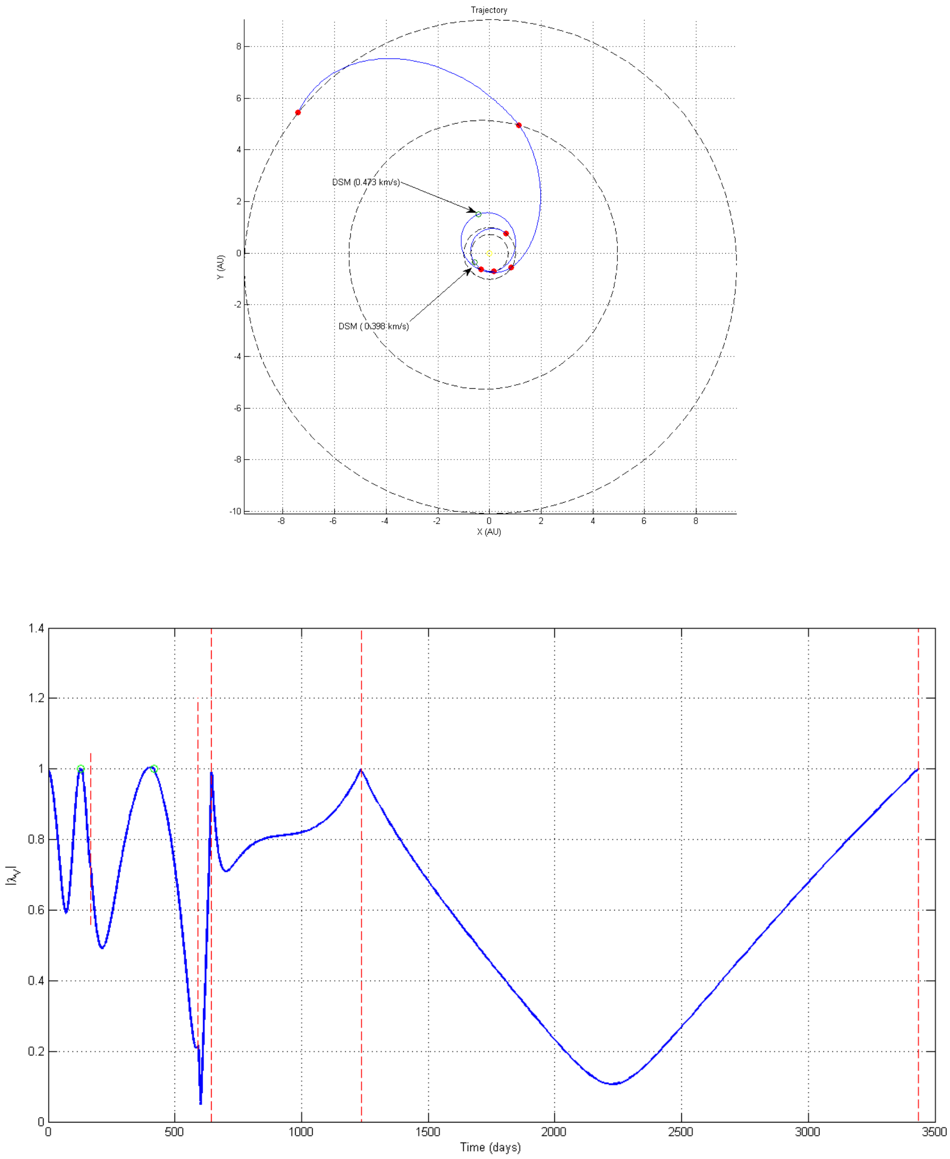

Figure 5: Example of an optimal EVVEJS trajectory (*Cassini*). Red dots and dash lines represent the swing-by locations and dates. Green dots represent the DSM locations.

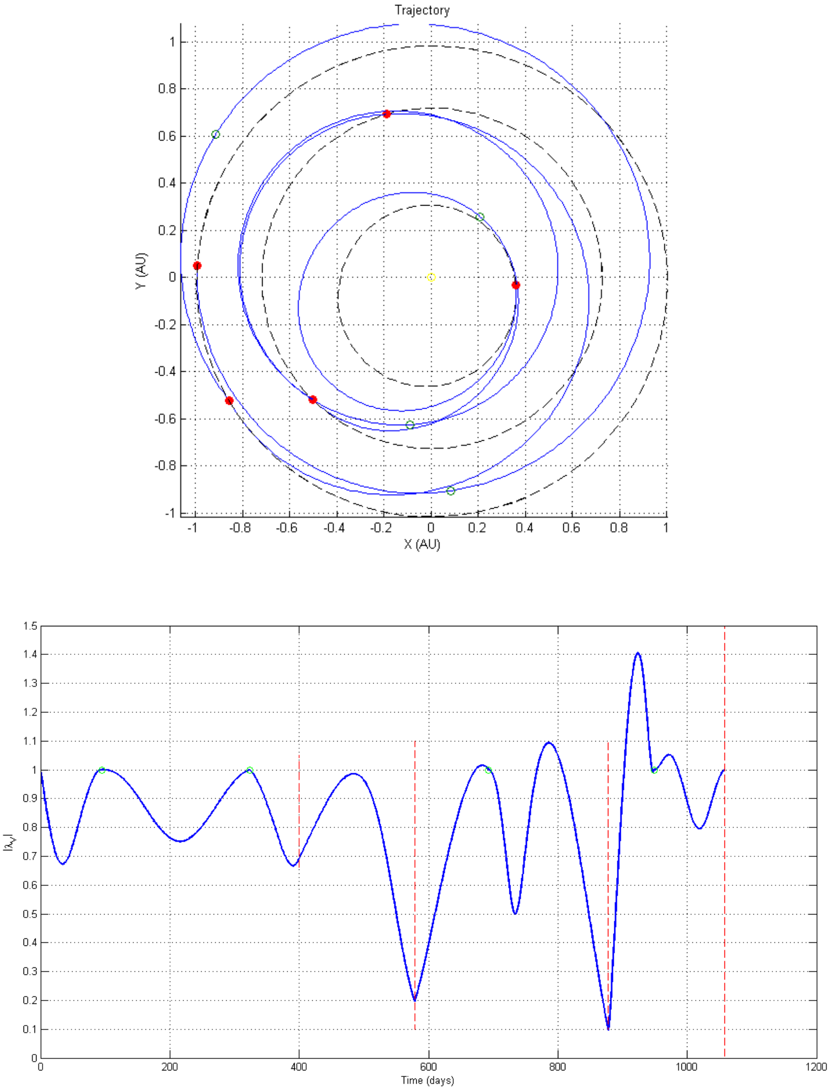

Figure 6: Example of an optimal EEVVY trajectory (*MESSENGER easy*). Red dot and dash lines represent the swing-by locations and dates. Green dots represent the DSM locations.

France) at this time.

References

[1] Gtop act trajectory database. http://www.esa.int/gsp/ACT/inf/op/globopt.htm, July 2009.

[2] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen. A global optimization method for the design of space trajectories. *Computational Optimization and Applications*, online, June 2009. doi:10.1007/s10589-009-9261-6.

[3] L.A. D'Amario, D.V. Byrnes, and R.H. Stanford. A new method for optimizing multiple-flyby trajectories. *Journal of Guidance and Control*, 4(6):591 – 596, 1981. doi:10.2514/3.56115.

[4] D.J. Jezewski. Primer vector theory and applications. NASA Technical report 454, NASA, 1975.

[5] D.F. Lawden. *Optimal Trajectories for Space Navigation*. Butterworths, London, 1963.

[6] J.P. Marec. *Optimal Space Trajectories*, volume 1 of *Studies in Astronautics*. Elsevier Scientific Publishing Company, 1979.

[7] D.J. Jezewski and H.L. Rozendhal. Efficient method for calculating optimal free space n-impulse trajectories. *AIAA Journal*, 6(11):2160–2165, 1968. doi:10.2514/3.4949.

[8] J. Navagh. Optimising interplanetary trajectories with deep space maneuvers. Master's thesis, George Washington Univ, 1993.

[9] O. Abdelkhalik and D. Mortari. N-impulse orbit transfer using genetic algorithms. *Journal of Spacecraft and Rockets*, 44(2), March 2007. doi:10.2514/1.24701.

[10] M. Vasile and P. DePascale. Preliminary design of multiple gravity assist trajectories. *Journal of Spacecraft and Rockets*, 43(4):794–805, 2006. doi:10.2514/1.17413.

[11] J. Olympio and J.P. Marmorat. Global trajectory optimisation: Can we prune the solution space considering deep space maneuvers ? Ariadna Final report 06/4101, ESA, 2007.

[12] R.H. Battin. *Introduction to the Mathematics and methods of Astrodynamics*. AIAA, New York, 1987.

[13] J.C. Niehoff. Gravity assisted trajectories to solar system targets. *Journal of Spacecraft*, 3(9):1351 – 1356, 1966. doi:10.2514/3.28659.

[14] L.A. Hiday-Johnston and K.C. Howell. Transfers between libration-point orbits in the elliptic restricted problem. *Celestial Mechanics and Dynamical Astronomy*, 58(4):317 – 337, 1994. doi:10.1007/BF00692008.

[15] S.P Hughes, L.M. Mailhe, and J.J. Guzman. A comparison of trajectory optimization methods for the impulsive minimum fuel rendezvous problem. *Advances in the Astronautical Sciences*, 113:85 – 103, 2003.

[16] M.D. Mesarovic, D. Macka, and Y. Takahara. *Theory of Hierarchical Multilevel Systems*. Academic Press. New-York, 1970.

[17] G. Cohen. Optimisation by decomposition and coordination: a unified approach. *IEEE transaction on automatic control*, 23(2), 1978.

[18] J. Batut and A. Renaud. Daily generation scheduling optimisation with transmission constraints: A new class of algorithms. *IEEE trans. On power Systems*, 7(3), 1992. doi:10.1109/59.207311.

[19] R. Bertrand. *Optimisation de trajectoires interplanétaires sous hypothèse de poussée faible*. PhD thesis, Université Paul Sabatier, Toulouse, France, 2001.

[20] G. Cohen and B. Miara. Optimisation with an auxiliary constraint and decomposition. *SIAM Journal of Control and Optimization*, 28(1):135 – 157, 1990. doi:10.1137/0328007.

[21] A. Bryson and Y.C. Ho. *Applied Optimal Control*. Hemisphere Publishing Corporation, New York, 1975.

[22] R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190 – 1208, 1995. doi:10.1137/0916069.

[23] P.E. Gill, W. Murray, and M.A. Saunders. Snopt: An sqp algorithm for large scale constrained optimisation. *SIAM Journal of optimisation*, 12(4), 2002. doi:10.1137/S1052623499350013.

[24] A. Wächter and L.T. Biegler. On the implementation of a primal-dual interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. doi:10.1007/s10107-004-0559-y.