23^{rd} Conference of the Italian Association of Aeronautics and Astronautics
AIDAA2015
Politecnico di Torino, 17-19 November 2015
E. Carrera, M. Cinefra, A. Pagani, M. Petrolo and E. Zappino (Editors)
www.aidaa2015.it

INVERSE DYNAMICS PARTICLE SWARM OPTIMIZATION FOR SPACECRAFT MINIMUM-TIME MANEUVERS WITH CONSTRAINTS

Dario Spiller[*], **Fabio Curti**[†], **Luigi Ansalone**[‡]

[*] Sapienza University of Rome
Via Eudossiana, 18
00184, Rome, Italy
e-mail: dario.spiller@uniroma1.it

[†] Sapienza University of Rome
Via Salaria, 851
00138, Rome, Italy
e-mail: fabio.curti@uniroma1.it

[‡] ASI Italian Space Agency
Via del Politecnico snc
00133, Rome, Italy
e-mail: luigi.ansalone@est.asi.it

Key words: Particle Swarm Optimization, Spacecraft Re-orientation Maneuvers.

Summary. *The problem of spacecraft time-optimal reorientation maneuvers under boundaries and path constraints is solved using the Particle Swarm Optimization technique. Keep-out constraints for an optical sensor are taken into account. A novel method based on the evolution of the kinematics and the successive obtainment of the control law is presented and named as Inverse Dynamics Particle Swarm Optimization. It is established that the computation of the minimum time maneuver with the proposed technique leads to near optimal solutions, which fully satisfy all the boundaries and path constraints.*

1. INTRODUCTION

The minimum time reorientation maneuver of a rigid spacecraft is a well-known problem: the first related work regarding a numerical approach dates back to the 90s [1] [2]. When introducing boundaries and path constraints, these maneuvers are difficult to compute because the solutions are related to complex nonlinear problems. The research still focuses on this problem with several approaches, for example, through homotopic approach algorithms [3], pseudospectral optimization analysis [4] [5] or with hybrid numerical techniques [6].

The problem of reorientation maneuvers with path constraints has been initially studied by McInnes in [7]. The path constraints may be determined by bright light sources (e.g. Sun and Moon) for optical sensors or by pointing boundaries for antennas to maintain communication [8]. Several numerical methods, such as the Randomized Motion Planning [9], the Logarithmic Barrier Potentials [10] or the Lie group variational integrator [11], have been used to obtain the optimal solution.

Metaheuristic algorithms (properly defined in [14]) have been recently proposed for the planning of slew maneuvers, as in [12] [13]. Metaheuristic algorithms may be used to find sub-optimal solutions: for instance, Melton proposed a hybrid technique where a metaheuristic solution was used as the best available initial guess for a pseudospectral optimizer [15]. The metaheuristic algorithms are being studied extensively and the high interest generated from their results is shown in the research performed by NASA [16]. With regards to the Particle Swarm Optimization (PSO), which is the primary interest of this paper, it has been used not only for the planning of attitude maneuvers as in [17] [18], but also for the trajectory planning [19] or for attitude determination [20].

From the previous works, the optimization of attitude maneuvers uses the PSO applied to the control. In [21] it has been shown how such an approach fails in satisfying the boundary constraints (i.e. final position and final velocity). In this work, a new approach is reported, where the PSO technique is applied to the kinematics rather than to the control. As a result, the final boundary constraints are straight satisfied. This approach is referred to as Inverse Method.

The paper is organized as follows: Sec. II reviews the PSO method. Sec. III shows the scenario in which the slew maneuver has to be accomplished. Sec. IV introduces the Inverse Method. Sec. V shows the numerical results. Sec. VI concludes the paper.

2. PSO Method

The PSO is an algorithm introduced by James Kennedy and Russel Eberhart in 1995 [22]. This method is based on cooperation between a fixed-size population of solutions [23]. This method presumes the evolution of a group of candidate solutions called *particles* that move through the set of all acceptable and meaningful solutions called *Feasible Research Space* FRS. Typical values of the number of particles $n_{particles}$ are between 30 and 50. During the evolution, each particle is evaluated according to a numerical value determined through a *fitness function* ϕ, which takes into account the goal of the optimization and the constraints. The aim is to find the minimum value of the fitness function.

The generic particle represents a possible optimal solution inside the FRS. The *position* of the generic particle $x_i^{(k)}$ is defined as the solution associated with the i^{th} particle at the k^{th} step of its evolution. With reference to Fig. 1, the particle is perturbed by a term called *velocity* evaluated as:

$$x_i^{(k+1)} = x_i^{(k)} + v_i^{(k+1)} \tag{1}$$

In each step the actual position $x_i^{(k)}$ is updated through the term $v_i^{(k+1)}$ reaching the new position $x_i^{(k+1)}$. Note that the term v_i is not a velocity in the traditional physical sense, i.e. $v_i \neq dx_i/dt$,

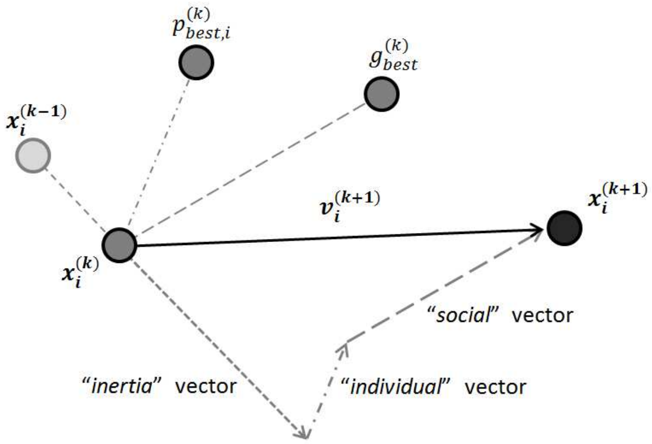

Figure 1. Representation of the displacement of a particle with respect to the different inputs.

but the velocity is the rate at which the position per generation changes. The *velocity* term dictates the direction of the evolution for the entire swarm.

In each step k, every i^{th} particle is assigned a performance index $J_i^{(k)}$ corresponding to the best value of the fitness function obtained by the particle up to the k^{th} step. The position of the particle related to this index is saved as *personal best* $p^{(k)}_{best,i}$.

The particle which has obtained the best value of the performance index up to the generic step k is saved as global best particle. Its perfomarce index and position are referred to as $J_g^{(k)}$ and $g^{(k)}_{best}$, respectively.

With respect to what is reported in Fig. 1, the displacement of the i^{th} particle is the sum of three different *velocity* vectors:

- The *"individual"* vector pointing toward the best position $p^{(k)}_{best,i}$.
- The *"social"* vector pointing toward the best position $g^{(k)}_{best}$ (*global version* of the PSO).
- The *"inertia"* vector which makes the i^{th} particle move in the direction in which the actual position has been reached. This means that the inertia vector is along the direction which goes from the positions $x^{(k-1)}$ to the position $x^{(k)}$ of the particle.

The individual vector and the social vector are multiplied by stochastic values uniformly ditributed between 0 and 1 in order to give some randomness to the search of the optimization goal.

In the *local version* [25] of the PSO the $g^{(k)}_{best}$ is substituted by a *local best* particle $l^{(k)}_{best,i}$ with performance index $J^{(k)}_{l,i}$: the i^{th} particle compares its fitness value only in a small neighborhood. In this manner, the whole search group tends to separate into subgroups thus enhancing the probability of finding the optimal solution when the problem has more than one suboptimal solution. The dimension of the neighborhood is a parameter of the algorithm that can influence the results. The *Unified Particle Swarm Optimization* UPSO strategy [24] combines the advantages of the *global* and *local* versions. A simple way to deal with UPSO is that, in the first part of the loop, the local best is considered to be more important than the global best. Hence, the swarm may compare different local minima (if various local minima are found) in the first part of the evolution. In the final part of the evolution, the global version is now privileged consequently allowing the swarm to converge quickly to the global minimum.

The *velocity* of the particles is described by the following formulation:

$$\begin{aligned} v_i^{(k+1)} = \; & r\left(w \cdot v_i^{(k)} + u_1 \cdot c_p \left(p_{best,i}^{(k)} - x_i^{(k)}\right) + \right. \\ & \left. + u_2 \cdot c_l \left(l_{best,i}^{(k)} - x_i^{(k)}\right) + u_3 \cdot c_g \left(g_{best}^{(k)} - x_i^{(k)}\right)\right) \end{aligned} \tag{2}$$

where u_1, u_2 and u_3 are different random numbers with uniform distribution between 0 and 1. The *inertia* vector is multiplied by a constant w, the *inertia weight*. The magnitude of the last three quantities depends on the *distance* of $p_{best,i}^{(k)}$, $l_{best}^{(k)}$ and $g_{best}^{(k)}$ from the actual position $x_i^{(k)}$. However, these terms are multiplied by user-defined constants (c_p, c_l and c_g respectively) and by the random number u_i which produces an amount of randomness to the displacement. The term r may be introduced as a scale factor usually decreasing during the evolution of the swarm.

The magnitude of the velocity and displacement terms must be limited in order to make the particle move only within the FRS. As a consequence, the following boundaries are introduced:

$$v_{min} < v < v_{max} \;, \quad x_{min} < x < x_{max} \tag{3}$$

where the subscript i has been removed for sake of simplicity. The maximum and minimum values of the velocity and the displacement are defined by the user: usually the maximum value of the velocity is set at about 10-20% of the dynamic range of the variable.

Convergence of the swarm towards a stable position with velocity tending to zero is demonstrated [23]. The time required for convergence is a function of the parameters in Eq. (2) and nothing guarantees that the point of convergence is the sought optimum. In our case, however, we will illustrate that the swarm meets in the neighborhood of the global optimum. The evolution of particles continues until a convergence criterion based on a user-defined tolerance is satisfied.

3. Problem statement

This study deals with the problem of a constrained reorientation slew maneuver in which a satellite must move from an initial attitude to a final attitude. The slewing motion must be constrained to prevent an optical sensor axis (we consider a star tracker) from entering into established “*keep-out*” zones. Such areas are cones that have central axes pointing to the Sun and the Moon and specified half-angles depending on the light magnitude, the distance from the satellite and the angular diameter of the source. The maneuver angle Θ_f and the initial and final attitudes are known. Moreover, the maneuver must be *rest-to-rest*, i.e. the angular velocity must be equal to zero for $t = t_0$ and $t = t_f$. In particular t_0 is fixed and equal to zero.

For every possible geometry of the keep-out zones we want to find the minimum-time maneuver. As a consequence, the performance index is equal to the maneuver overall time:

$$J = t_f - t_0 \tag{4}$$

Supposing that the time for completing the maneuver is negligible with respect to the time for completing an orbit, it is possible to approximate the velocity of the satellite center of mass $\boldsymbol{CM}$ to zero. As a consequence we can define an inertial reference frame fixed in the original position of the body-fixed frame at time $t = t_0$ that will be referred to as BRF$_0$. The positions of the keep-out cones defined in this inertial frame do not change during the maneuver. The angular velocity is defined in the body-fixed frame and it is denoted as $\boldsymbol{\omega} = [\omega_x \ \ \omega_y \ \ \omega_z]^T$.

The rigid-body motion is described by the Euler's equation expressed in the most general form as follows:

$$I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I\boldsymbol{\omega} = \mathrm{M} \tag{5}$$

where I is the inertia tensor and M the total torque vector. In particular we will consider three independent torques aligned with the axes of the body- fixed frame.

In the evolution between the initial and the final positions, the sensor axis must be kept at least at the minimum angular distance α_s from each light source. Using the notation from [12] we can define the so-called *keep-out constraints* as:

$$C_s(t) = \boldsymbol{\sigma}(t) \cdot \boldsymbol{\sigma}_s - \cos(\alpha_s) \leq 0 \qquad \forall t \in [t_0, t_f] \tag{6}$$

where $\boldsymbol{\sigma}(t)$ is the direction pointed to by the optical sensor and $\boldsymbol{\sigma}_s$ is vector placed in the center of mass of the satellite and pointing to the generic source of light, here represented by *s*. Introducing the new variable β defined as $\beta(t) = \cos^{-1}(\boldsymbol{\sigma}(t) \cdot \boldsymbol{\sigma}_s)$, the constraint in Eq. (6) may be re-written more easily as:

$$\beta(t) \geq \alpha_s \qquad \forall t \in [t_0, t_f] \tag{7}$$

The meaning of the constraint is shown in Fig. 2. On the left, a feasible configuration is reported: the sensor axis σ is outside the keep-out cone being $\beta > \alpha_s$. On the right, an unfeasible configuration is reported: in this case, the sensor axis σ is inside the keep-out cone since $\beta < \alpha_s$.

The cone defined for each source of light will be referred to as *keep-out cone*. The angle of exclusion α_s is determined by the sensor baffle and the source (considering its angular dimension and its intensity), and it is typically between 15 and 45 degrees depending on the sensitive optical sensor [26] [27].

4. Inverse Dynamics Particle Swarm Optimization

The traditional way of dealing with these kind of optimization problems is the direct integration of the control law. This approach shows some issues due to the high computational load

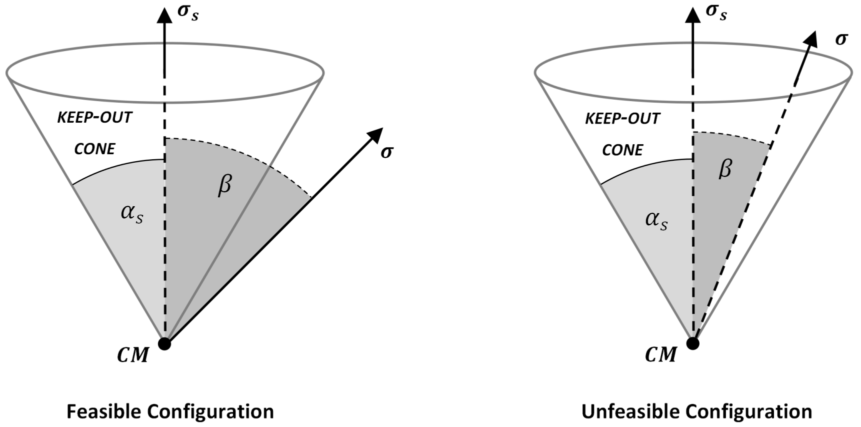

Figure 2. Graphical representation of the keep-out cone constraint.

required by the integration of the Euler equations and the difficulty of completely satisfy the boundary constraints.

A novel approach has been recently presented [21]: instead of applying the PSO to the control, we use the PSO approach to the kinematics. This technique may be referred to as Inverse Dynamics and the acronym "*i*PSO" will be used consequantly hereinafter. The novelty of the *i*PSO method lies in the fact that it may be used as a sub-optimal planner since both the initial and the final conditions are imposed *a priori* for each particle.

The *i*PSO problem may be summarized as follows:

$$\begin{aligned}
Find: &\quad \min t_f \\
\text{subjected to} & \\
& \\
dynamic\, constraints: &\quad \mathbf{M} = f(\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}) \\
initial\, conditions: &\quad \dot{\mathbf{p}}(t_0) = \mathbf{0} \\
&\quad \mathbf{p}(t_0) - \mathbf{p}_0 = \mathbf{0} \\
final\, conditions: &\quad \dot{\mathbf{p}}(t_f) = \mathbf{0} \\
&\quad \mathbf{p}(t_f) - \mathbf{p}_f = \mathbf{0} \\
path\, constraint: &\quad \boldsymbol{\sigma}(t) \cdot \boldsymbol{\sigma}_x - \cos(\alpha_x) \leq 0 \qquad \forall t \in [t_0, t_f] \\
control\, constraint: &\quad |M_\nu(t)| - M_{max} \leq \mathbf{0}\ ,\ \ \nu = 1, 2, 3 \qquad \forall t \in [t_0, t_f]
\end{aligned} \tag{8}$$

The fitness function is selected in the form of an *Exterior Penalty Function* (explained for example in [29], [30] and [31] with special regards to genetic algorithms, or more generally in [14]). Such a function is problem-dependent and must be built according to the characteristics of the constraints. The path and the control constraints are inequality constraints that must be taken into account in the fitness function. Consequently, the fitness function is chosen as follow:

$$\phi = t_f + \sum_{i=1}^{N_{ineq}} G_i + c\, N_{viol} + f \tag{9}$$

The first term of the fitness function is the time required for completing the maneuver.

The second term takes into account the penalty function G_i which is defined for the i^{th} inequality constraint as:

$$G_i = l_i \sum_{j=0}^{n} \nu_i(t_j) \tag{10}$$

where l_j is a user-defined constant and $\nu_i(t_i)$ takes the following forms depending on the type of constraint:

- Path constraint:

$$\nu_i(t_j) = \begin{cases} 0 & \text{if } \boldsymbol{\sigma}(t_j) \cdot \boldsymbol{\sigma}_x - \cos(\alpha_x) < \Delta_i \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

- Control constraint:

$$\nu_i(t_j) = \begin{cases} 0 & \text{if } |M_\nu(t_j)| - M_{max} < \Delta_i \ , \ \ \nu = 1, 2, 3 \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

The most important feature of the present formulation is how the constraints are considered. In fact, differently from other works in literature [12], here the constraints' tolerances Δ_j decrease during the swarm evolution. In this work, we have set a piecewise linear decreasing law: the more the solution converges the more the tolerance Δ_j decreases. The tolerances decrease according to the following law:

$$\Delta_{j+1} = \begin{cases} \Delta_j - k_{dec} \cdot m \cdot 10^{\xi - \mu} & \text{if } \Delta_j \geq 1e-5 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where μ is equal to 1, m and ξ are the mantissa and the exponent of Δ_j, i.e. $\Delta_j = m \cdot 10^{\xi}$. The parameter k_{dec} is a user-defined parameter with the following law:

$$k_{dec_{j+1}} = k_{dec_j} + K\, \frac{j-1}{N^*} \tag{14}$$

N^* is a user-defined parameter. The inequality constraints are fully satisfied only when $\Delta_j = 0$. Eq. (13) is taken into account only when the global best satisfies all the constraints.

The third term in eq. (9) is related to number of violated constraints N_{viol}, being c a user-defined constant. This term allow to reward the particles which satisfy more constraints than other. Every time the global best particle reaches the value of $N_{viol} = 0$ the precision is improved, and the tolerances decrease.

The last parameter f in eq. (9) takes into account the geometry of the problem. This term is set to 0 if the keep-out cones do not intersect. In this case a generic maneuver can pass through the keep-out cones, even though the best maneuver may lie outside this region. On the other hand, if the keep-out cones intersect, then $f = 1e10$ if the optical sensor axis $\boldsymbol{\sigma}_x$ passes between the cones and $f = 0$ otherwise. In this way we avoid to concentrate the swarm around solution that are in an unfeasible region since there is no passagge between the two keep-out cones.

The particles are structures containing:

- An array $t^{(i)}(k)$ containing the discretized time interval where i identifies the particle and $k = 1, 2, ..., m$. m is equal to the number of points used for the interpolation of the angular displacement. The array $t^{(i)}$ is monotonically increasing being $t^{(i)}(1) = t_0 = 0$ and $t^{(i)}(m) = t_f^{(i)}$.
- The angular displacement $\zeta_j^{(i)}(k)$, where $j = 1, 2, ..., n$. In particular n is the number of scalars required by the chosen angular representation. The m points are associated to time instants contained in the array $t^{(i)}$. The kinematics is obtained interpolating the m points $[t^{(i)}(k); \zeta_j^{(i)}(k)]$ with B-splines.

The novelty of the proposed approach lies in the fact that also the time array move according to the PSO strategy. This characteristic allow a better interpolation of the kinematic curves then the one reported in [21]. $\Delta\zeta_j^{(i)}(k)$ and $\Delta t^{(i)}(k)$ are the velocities associated to the kinematics and the time instants of each particle of the swarm. In this case Eq. (3) takes the following form:

$$\begin{gathered}
|\Delta\zeta_j^{(i)}(k)| \leq 0.02 \cdot \tan(\theta^*/4)\ ,\quad |\zeta_j^{(i)}(k)| \leq \tan(\theta^*/4) \\
|\Delta t^{(i)}(k)| < 0.01 \cdot (t_{max} - t_{min})\ ,\quad t_{min} < t_f^{(i)} < t_{max} \\
i = 1, ..., n_{particles}\ ,\quad j = 1, ..., m
\end{gathered} \tag{15}$$

The expression $\tan(\theta^*/4)$ is explained in Eq. (18), while θ^* is an angle which satisfies $\theta^* \geq \Theta_f$, being Θ_f the imposed angle of maneuver. The time constraints t_{max} and t_{min} may be defined by knowing the unconstrained solution. The values 0.01 and 0.02 have shown to give good results in the numerical experiments.

The initialization of the kinematics and the maneuver time of each particle is based on a uniform random distribution of the particles within the constraints of Eq. (15). The time arrays are initialized as sets of uniformly distributed points between t_0 and $t_f^{(i)}$.

In this approach, given the angular displacement of each particle, the control is directly evaluated by Eq. (5). As a consequence, $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are needed in order to obtain $\mathbf{M}$. The setting of the problem depends on:

1. Choice of the interpolation method.

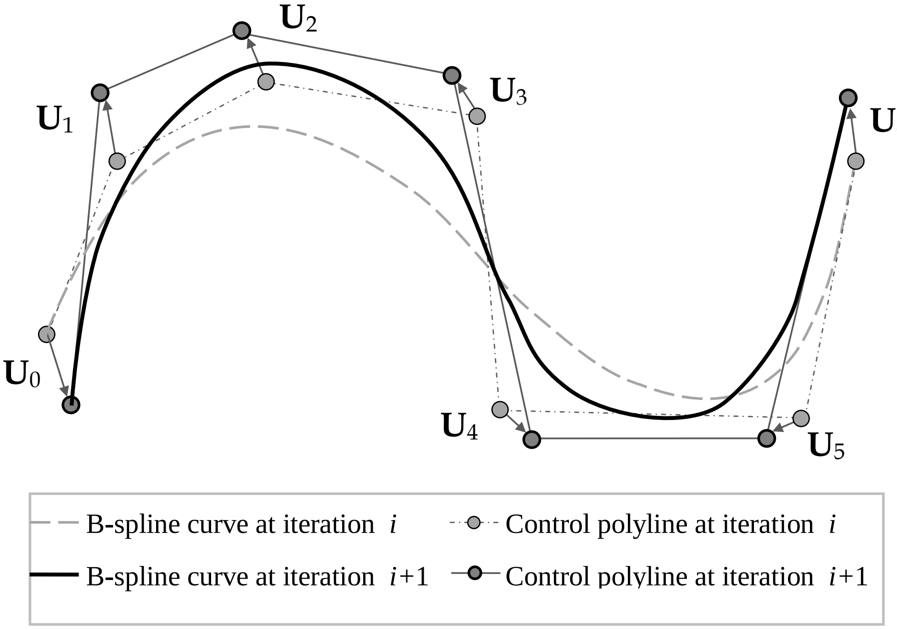
U2
U1
U3
U6
U0
U4
U5
B-spline curve at iteration i
Control polyline at iteration i
B-spline curve at iteration i+1
Control polyline at iteration i+1

Figure 3. Example of interpolation curve obtained with a clamped B-spline.

2. Choice of the parameters for the attitude kinematics.

The first point is related to the formulas reported in Eq. (24) to Eq. (27). In fact, the representation used for the description of the kinematics must easily allow the computation of the needed derivatives. The problem may be solved using the B-spline (*Basis-spline*) interpolation with the *Cox-de Boor recursion formula* [33]. B-splines are based on the fact that each segment of the interpolating curve is built upon several rather than only one polynomial. The clamped version of the B-spline is used here, where clamped means that the curve passes through the initial and final control points $\mathbf{U}_0$ and $\mathbf{U}_f$. The other control points $\mathbf{U}_1$, $\mathbf{U}_2$ ..., $\mathbf{U}_{f-1}$ define the shape of the whole curve. An example of clamped B-spline is reported in Fig. 3. As it can be seen from the figure, the control points may move in any direction since the particle may change both the time instants and the values of the kinematic displacement.

It is recommended to interpolate the angular displacement at least with curves based on functions with differentiability class C^4 in order to have smooth and continue second derivatives (i.e. the angular acceleration). Moreover, the first derivative of the angular displacement at $t = t_0$ and $t = t_f$ may be user-defined since the clamped B-spline curve is tangent to the first and last leg of the control polyline, as shown in Fig. 3. The curve is tangent to the first leg $\overline{\mathbf{U}_0\mathbf{U}_1}$ in $\mathbf{U}_0$ and tangent to the last leg $\overline{\mathbf{U}_5\mathbf{U}_6}$ in $\mathbf{U}_6$.

As a consequence, to obtain a *rest to rest* (i.e. with velocity equal to zero in $t = t_0$ and $t = t_f$) maneuver, it is only necessary to set the first two points $\mathbf{U}_0$ and $\mathbf{U}_1$ and the last two points $\mathbf{U}_{f-1}$ and $\mathbf{U}_f$ of each angular displacement curve at the same ordinate value. In fact, as stated in Eq. (26), when $\omega_x = \omega_y = \omega_z = 0$ also the Modified Rodrigues Parameters are equal to zero. Summarizing, four points for each of the three curves interpolating the MRP are defined. As a result, we can set $\dot{\mathbf{p}}(t_0) = 0$ and $\dot{\mathbf{p}}(t_f) = 0$ by imposing:

$$\mathbf{U}_0 = \mathbf{U}_1 = \mathbf{p}_0 \qquad \mathbf{U}_{f-1} = \mathbf{U}_f = \mathbf{p}_f \tag{16}$$

With regard to the attitude representation, an attitude description with only three parameters is essential to avoid the invertibility of the kinematics matrix. The Modified Rodrigues Parameters (MRPs, [28]) are suitable for this issue. Moreover, MRPs show no singularity during the maneuver because of the imposed Θ_f is below 2π. The mathematical formulation that describes the kinematics through the MRPs is reported below. A vector $\mathbf{p}$ is defined as follows:

$$\mathbf{p} = \frac{\boldsymbol{\eta}}{1 + \eta_4} \tag{17}$$

where $\boldsymbol{\eta}$ and η_4 are the vectorial and scalar components of the quaternion. Moreover $\mathbf{p}$ may be rewritten in terms of axis and angle of rotation as:

$$\mathbf{p}(\hat{\mathbf{n}}, \theta) = \tan(\theta/4)\hat{\mathbf{n}} \tag{18}$$

The rotation matrix using the Modified Rodrigues Parameters appears as:

$$R(\mathbf{p}) = I + \frac{4\,(1 - |\mathbf{p}|^2)}{(1 + |\mathbf{p}|^2)^2}\,[\tilde{\mathbf{p}}] + \frac{8}{(1 + |\mathbf{p}|^2)^2}\,[\tilde{\mathbf{p}}]^2 \tag{19}$$

where $[\tilde{\mathbf{p}}]$ is defined as:

$$[\tilde{\mathbf{p}}] = \begin{bmatrix} 0 & p_3 & -p_2 \\ -p_3 & 0 & p_1 \\ p_2 & -p_1 & 0 \end{bmatrix} \tag{20}$$

In particular:

$$\boldsymbol{\sigma}(t) = \mathrm{R}(\mathbf{p})^T\,\boldsymbol{\sigma}(t_0) \tag{21}$$

The derivative of the Modified Rodrigues Parameters are related to the angular velocity by the following equation:

$$\dot{\mathbf{p}} = \frac{1}{4}\Psi(\mathbf{p})\boldsymbol{\omega} \tag{22}$$

where the matrix $\Psi(\mathbf{p})$ is defined as:

$$\Psi(\mathbf{p}) = \left[\left(1 - \mathbf{p}^T\mathbf{p}\right)I + 2[\tilde{\mathbf{p}}] + 2\mathbf{p}\mathbf{p}^T\right] \tag{23}$$

For the following development of the inverse dynamics with PSO algorithm, it is necessary to find $\boldsymbol{\omega}(\mathbf{p}, \dot{\mathbf{p}})$ and $\dot{\boldsymbol{\omega}}(\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}})$. As far as the former vector is concerned, it is quite simple to obtain it from Eq. (22):

$$\boldsymbol{\omega} = 4\Psi^{-1}(\mathbf{p})\dot{\mathbf{p}} \tag{24}$$

Ψ^{-1} is defined as a near-orthogonal matrix since its inverse matrix is proportional to its transpose:

$$\Psi^{-1}(\mathbf{p}) = \frac{\Psi^T(\mathbf{p})}{(1+\mathbf{p}^T\mathbf{p})^2} \tag{25}$$

From Eq. (24) an important consequence may be drawn:

$$\boldsymbol{\omega} = \mathbf{0} \iff \dot{\mathbf{p}} = \mathbf{0} \tag{26}$$

From Eq. (24), $\dot{\boldsymbol{\omega}}(\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}})$ may be derived as:

$$\dot{\boldsymbol{\omega}} = 4\left(\dot{\Psi}^{-1}(\mathbf{p})\,\dot{\mathbf{p}} + \Psi^{-1}(\mathbf{p})\,\ddot{\mathbf{p}}\right) \tag{27}$$

where $\dot{\Psi}$ and $\dot{\Psi}^{-1}$ are evaluated as:

$$\begin{aligned} \dot{\Psi} &= \left[-(\dot{\mathbf{p}}^T\mathbf{p} + \mathbf{p}^T\dot{\mathbf{p}})I + 2[\tilde{\dot{\mathbf{p}}}] + 2(\dot{\mathbf{p}}\mathbf{p}^T + \mathbf{p}\dot{\mathbf{p}}^T)\right] \\ \dot{\Psi}^{-1} &= \frac{\dot{\Psi}^T}{(1+\mathbf{p}^T\mathbf{p})^2} - \frac{2\Psi^T}{(1+\mathbf{p}^T\mathbf{p})^3}(\dot{\mathbf{p}}^T\mathbf{p} + \mathbf{p}^T\dot{\mathbf{p}}) \end{aligned} \tag{28}$$

These equations fully describe the attitude kinematics through the Modified Rodrigues Parameters. The main feature of the above equations is that an analytical closed-form solution is found to compute $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$. Although the mathematical form of these equations is more complex than the mathematical form described in the attitude kinematics with the quaternions, the advantage is in dealing with square matrices . In order to summarize these results, placing Eq. (24) and (27) in Eq. (5) the following relation is obtained:

$$\mathbf{M} = f(\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}) \tag{29}$$

The main steps of the *i*PSO approach are reported in Algorithm 1. In the code, N_{int} represents the number of internal loops.

The *i*PSO approach may be used as:

1. Planner for near minimum-time maneuvers: the algorithm guarantees that all constraints are satisfied. Moreover, the numerical results will prove that we can obtain a maneuver time very close to that obtained with a Pseudospectral Optimal Control Software POCS [32] approach in small computational times.

2. Initial guess for a POCS approach, reducing the required computational time for the obtainement of the solution.

5. Numerical Results

A satellite for Earth observation in LEO is taken as test case. The nominal attitude is defined as:

Algorithm 1: *i*PSO algorithm

```
Initialization of constants, swarm and tolerances;
while $toll > 1e-8$ do
    upgrade $c_l$ and $c_g$;
    if $N^*_{viol} = 0$ and $i > 1$ then
        upgrade the tolerance;
        reset $J$, $J_l$ and $J_g$ if $g^{(i-1)}_{best}$ is not consistent with the new tolerances;
    end
    for $ii = 1 : N_{int}$ do
        upgrade $r$ and $w$;
        for $j = 1 : n_{particles}$ do
            interpolate $\mathbf{p}$, $\dot{\mathbf{p}}$ and $\ddot{\mathbf{p}}$;
            compute the sensor rotation from Eq. (21);
            compute of $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ through Eq. (24) and (27) and $\mathbf{M}$ from Eq. (29);
            evaluate penalty functions $G$, $N_{viol}$ and parameter $f$;
            compute the fitness function $\phi$, Eq. (9);
            if $\phi < J(j)$ then
                $J(j) = \phi$;
            end
            if $J(j) < J_g$ then
                $J_g = J(j)$;
                save $N^*_{viol}$;
            end
            if $J(j) < J_l(j)$ then
                $J_l(j) = J(j)$;
            end
        end
        for $j = 1 : n_{particles}$ do
            update the swarm velocity and position;
        end
    end
end
```

- The $\mathbf{Z}_b$ axis points in the nadir direction towards the Earth.
- The $\mathbf{X}_b$ axis is in the direction of the spacecraft velocity vector for circular orbits.
- The $\mathbf{Y}_b$ axis completes the right-handed coordinate system and it is perpendicular to the orbital plane in the negative orbit normal direction.

The inertial reference frame BRF$_0$ is associated to the coordinate system $\{\mathbf{X}_b,\ \mathbf{Y}_b,\ \mathbf{Z}_b\}$ at time $t = t_0$. In the following *roll rotations* around $\mathbf{X}_b$ and *pitch rotations* around $\mathbf{Y}_b$ will be considered. The rotation angle is always set to $\Theta_f = 60$ degree. At $t = t_0$ the satellite is in its nominal attitude. The inertia tensor is diagonal in the BRF and has the values $I_x = 3000$ kg·m^2, $I_x = 4500$ kg·m^2 and $I_x = 6000$ kg·m^2. We assume that the attitude maneuvers are

Table 1. Direction of Sun and Moon in BRF_0 for the proposed case studies.

	Sun			**Moon**			**Free Angle**
	x [-]	y [-]	z [-]	x [-]	y [-]	z [-]	[degree]
Case 1	-0.582	-0.083	-0.809	0.405	-0.132	-0.905	0.518
Case 2	-0.649	-0.649	-0.397	0.100	-0.292	-0.951	0.836
Case 2	-0.123	-0.975	-0.187	-0.187	-0.317	-0.930	0.596
Case 4	-0.882	-0.126	-0.454	-0.060	0.019	-0.998	0.779

obtained through three independent torques aligned with the BRF with same maximum value $M_{max} = 0.25$ Nm.

Taking an angle $\xi = 38$ degree, let us consider a star tracker sensor mounted on the $\mathbf{Y}_b\mathbf{Z}_b$ plane with the unit vector $\mathbf{ST}$ expressed in the in BRF as:

$$\mathbf{ST} = [0 \quad -sin\xi \quad -cos\xi]^T \tag{30}$$

The *i*PSO has been tested with $n_{particles} = 30$. With regards to Eq. (2), the inertia weight w, the local best constant c_l and the global best constant c_g linearly go from $w_0 = 1.2$, $c_{l_0} = 2$,

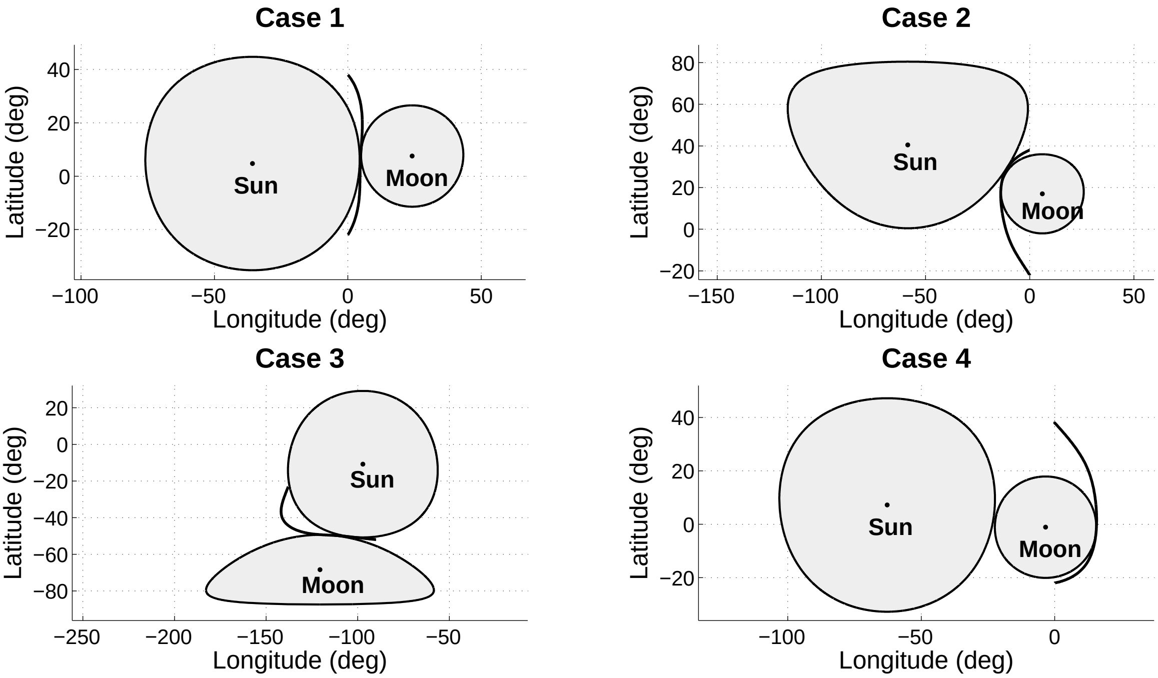

Figure 4. Geometries of the proposed case studies.

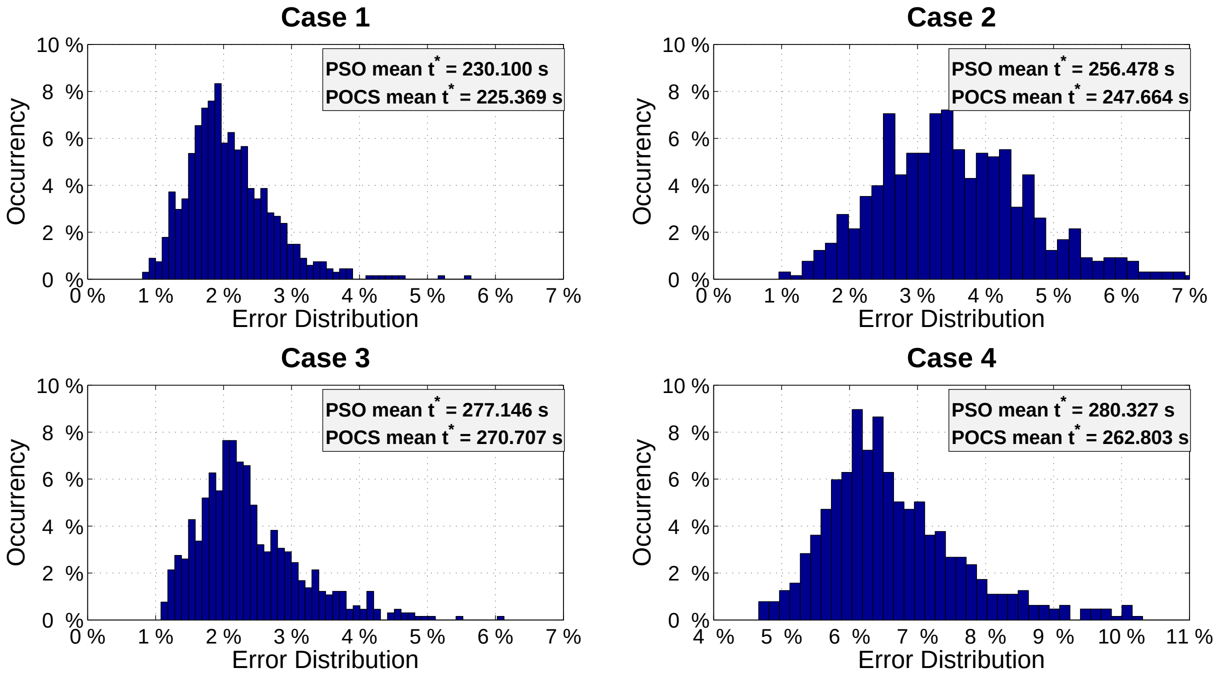

Figure 5. Monte Carlo results over 600 test cases.

$c_{g_0} = 0$ to $w_f = 0.6$, $c_{l_f} = 0$, $c_{g_f} = 2$. The number of iterations for the internal loop N_{int} has been set equal to 20. The scale factor r has been set to 0.9 and the cognitive constant c_p is equal to 1.5. With regard to the neighbourhood chosen for the $l_{best,i}^{(k)}$ term, the i^{th} particle compares its fitness value with the particles $i \pm j : j = 1, 2, 3$. The constants c in Eq. (9) and l_j in Eq. (10) have been set to 10.

The initialization values for k_{dec_j} is 0.2 and the value of K may be set equal to 1. The values of Δ are initialized as a function of the keep-out cones geometry. Considering the ideal minimum-time maneuver going from the initial attitude to the final attitude without considering the keep-out cones, we evaluate the minimum distance between the optical axis and the light source axes ζ_{sun} and ζ_{moon}. The parameter λ^* is chosen as $0.8 \cdot \min(\zeta_{sun}, \zeta_{moon})$. Introducing a parameter $\epsilon = 10$ degree, we impose for the generic light source s:

$$\Delta_{1,s} = \begin{cases} 5 \cdot 10^{-3} & \text{if } \zeta_s > \lambda_s \\ 1 - \cos(\lambda_s - \lambda^*) & \text{if } \zeta_s > \epsilon \text{ and } f = 0 \\ 1 - \cos(\lambda_s) & \text{otherwise} \end{cases} \tag{31}$$

$$\Delta_{1,control} = 0 \tag{32}$$

For numerical reasons, normalized units are considered: the control is divided by M_{max}, the inertia matrix by I_1 and the maneuver time by $\sqrt{I_1/M_{max}}$. The number of external iterations is

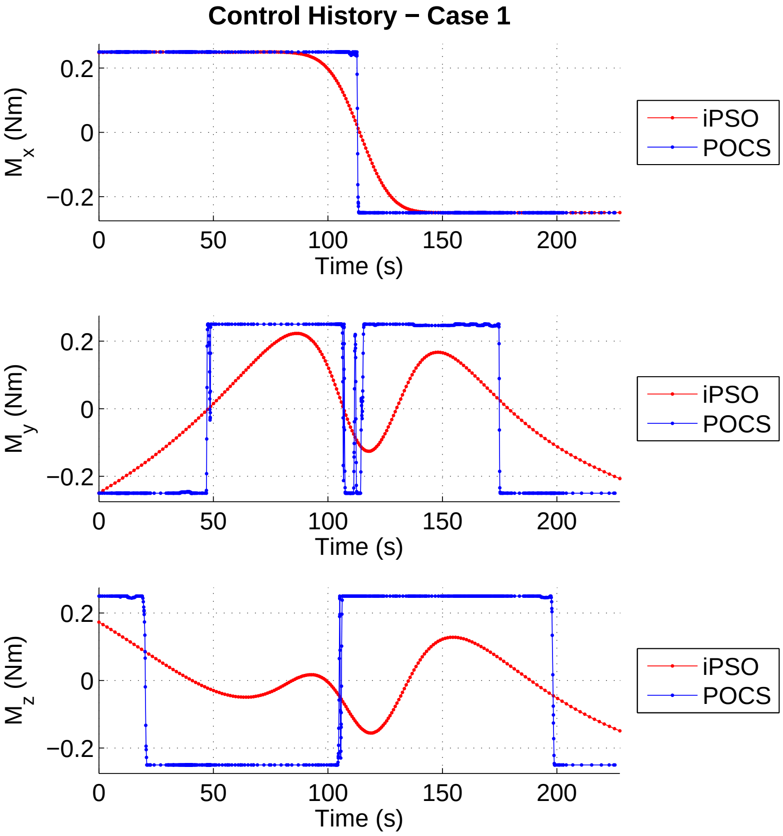

Figure 6. Comparison of the *i*PSO and POCS results for the otimal control.

a function of the tolerance criteria reported in Eq. (33).

$$\frac{1}{10}\sum_{i=1}^{10}(J_g^i - J_g^{i+1}) < 1e-8 \tag{33}$$

All results are obtained considering a PC with a processor Intel® Core$^{\text{TM}}$ i7-2670QM CPU @ 2.20GHz and with 6.00 GB of RAM.

Four different case studies are proposed, whose characteristics are shown in the Table 1 and reported in Fig. 4. The directions of the Moon and the Sun are referred in BRF$_0$. The free angle is the space between the two keep-out cones. The half-angles of the cones centered on the Sun and on the Moon are set to $\lambda_{sun} = 40$ deg and $\lambda_{moon} = 19$ deg, respectively. The first and the second cases are different roll rotations with the minimum-time maneuver between the keep-out cones. The third case is a pitch rotation and the last case is a roll rotation where the minimum-time maneuver is not beetween the two cones. The maneuvers reported in Fig. 4 have been obtained with the *i*PSO approach.

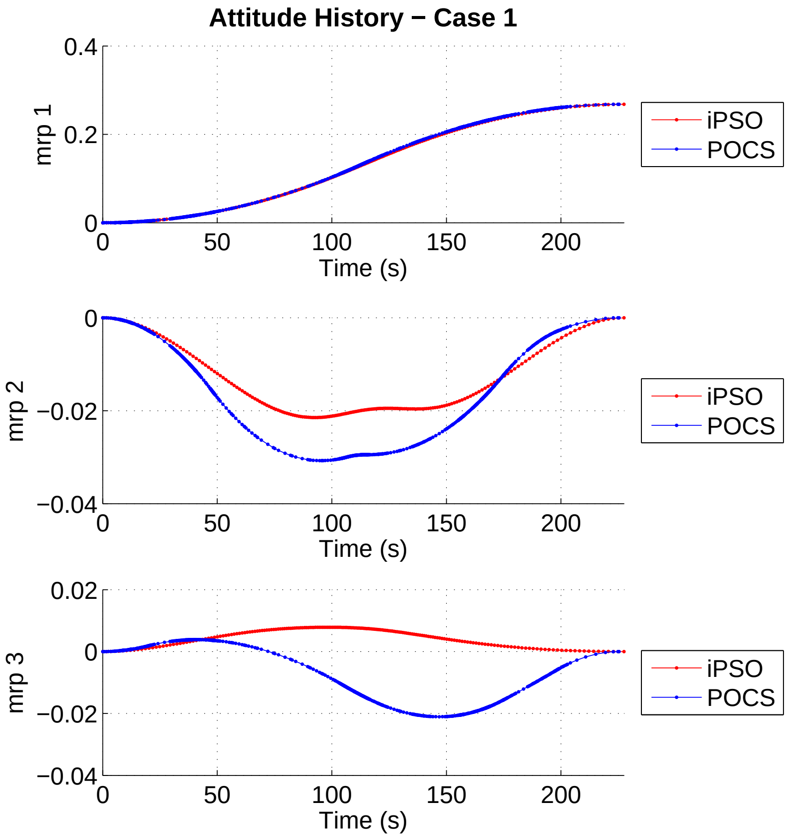

Figure 7. Comparison of the *i*PSO and POCS results for the attitude.

All the results obtained with the proposed approach have been compared with the results obtained with a POCS. In Fig. 5 the percentage error of the *i*PSO otimal time with respect to the POCS optimal time is reported after having carried out a Monte Carlo simulation of 600 experiments. As it can be seen, case 1 and case 3 have a mean error of about 2%, while in the other two cases we arrive at a maximum mean error of about 6%. The most important characteristic of the proposed approach is that the solution is always around the POCS solution, i.e. the problem of local minima associated with other possible trajectory around the esclusion cone is completely avoided. This is particularly important for cases 2 and 4, where a local minimum with final time close to the obtained minimum time exists on the opposite side of the reported maneuver (Fig. 4).

Detailed results have been reported for case 1 chosing one reference experiment. From Fig. 6, 7 and 8 we can see that the *i*PSO solution along the x axis is quite identical to the POCS solution. In this case, the maneuver is mainly along the x axis: this result means that the main characteristics of the maneuver have been caught from the *i*PSO solution. The y and z axes show *i*PSO trends that differ from the POCS ones. It must be noted that all the constraints are satified

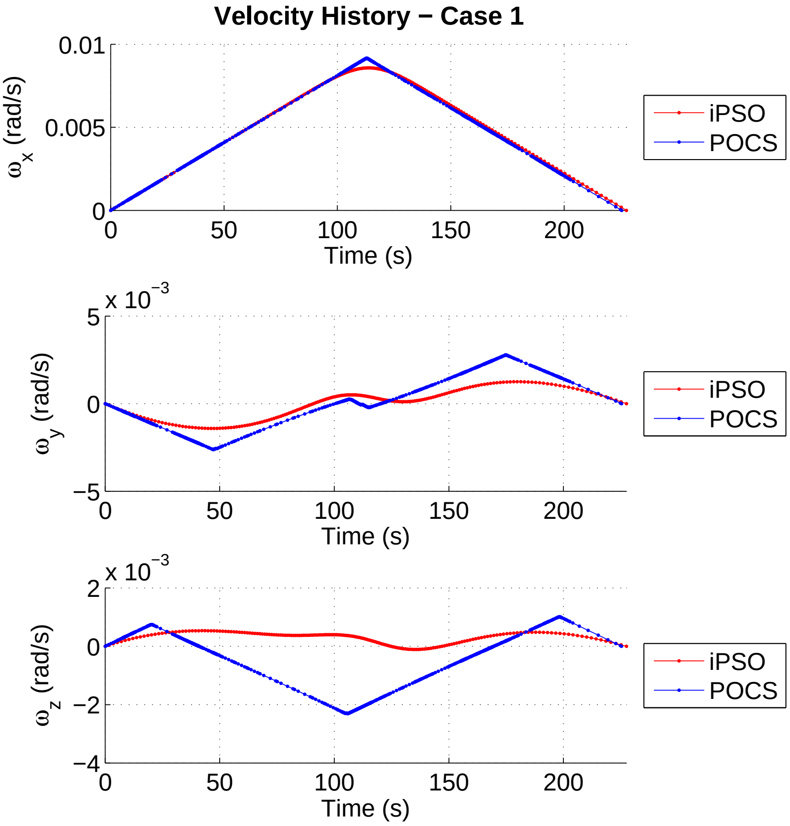

Figure 8. Comparison of the *i*PSO and POCS results for the angular velocity.

by both the *i*PSO and POCS solutions. All the proposed test cases shows these features: the future development of the proposed approach will take into account the possibility to improve the solution in order to obtain the same accuracy for all the three axes.

The mean computational times required by the proposed *i*PSO approach has been reduced with respect to the one reported in [21]: for the reported test cases about 50 seconds are required for the obtainement of the solution. This time does not depend on the particular geometry of the analysed cases. A further reduction of the computational time will be one of the goal of the future development of the algorithm.

Accordigly with previous work in literature [15] [21], it has been noted that, using the *i*PSO solution as best guess for the POCS, computational times may be considerably reduced. For example, solving case 2 with the *i*PSO guess requires about 70 seconds, while about 3800 seconds are required without best guess.

6. Conclusion

It has been demonstrated that the Particle Swarm Optimization may be used for planning sub-optimal constrained maneuvers. The proposed Inverse Method guarantees solutions that fully satisfy boundary and path constraints.

The final time obtained with the prposed method is greater than the time computed using a pseudospectral method. However, the introduced movement of the time instants leads to solutions very close to the ones obtained with a pseudospectral optimization software, with errors less than 1% in some reported test cases.

The low computational effort and the satisfaction of all the imposed constraints make the proposed approach suitable in the perspective of achieving fully autonomous satellites.

References

[1] Bilimoria, K.D., Wie, B., "Time-optimal three-axis reorientation of rigid spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 3, 1993, pp. 446-452. doi: 10.2514/3.21030

[2] Li, F., Bainum, P.M., "Numerical approach for solving rigid spacecraft minimum time attitude maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, pp. 38-45, 1990. doi: 10.2514/3.20515

[3] Li, J., Xi, X.,"Fuel-optimal low-thrust reconfiguration for formation-flying satellites via homotopic approach," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, pp. 1709–1717, 2012. doi: 10.2514/1.57354

[4] Li, J., Xi, X., "Time-optimal reorientation of the rigid spacecraft using a pseudospectral method integrated homotopic approach," *Optimal Control Applications and Methods*, 15 October 2014. doi: 10.1002/oca.2145

[5] Yutko, B. M., Melton, R. G., "Optimizing spacecraft reorientation maneuvers using a pseudospectral method," *Journal of Aerospace Engineering, Sciences and Applications*, Vol. 2, No. 1, pp. 1-14, January-April 2010.

[6] Bai, X., Junkins, J. L., "New Results for Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft," *Journal of Guidance, Control and Dynamics*, Vol. 32, No. 4, pp. 1071-1076, July-August 2009. doi: 10.2514/1.43097

[7] McInnes, C. R., "Large angle slew maneuvers with autonomous sun vector avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, pp. 875-877, 1994. doi: 10.2514/3.21283

[8] Hablani, H. B., "Attitude Commands Avoiding Bright Objects and Maintaining Communication with Ground Station," *Journal of Guidance, Control and Dynamics*, Vol.22, No.6, November-December1999, pp. 759-767. doi: 10.2514/2.4469

[9] Frazzoli, E., Dahleh, M. A., Feron, E., Kornfel, R. P., "A Randomized Attitude Slew Planning Algorithm for Autonomous Spacecraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, No. AIAA-2001-4155, pp. 1-8, Montreal, Quebec, Canada, August 6-9, 2001.

[10] Lee, U., Mesbahi, M., "Spacecraft Reorientation in Presence of Attitude Constraints via Logarithmic Barrier Potentials," *American Control Conference*, pp. 450 - 455, O'Farrell Street, San Francisco, CA, USA, June 29 - July 01, 2011. doi: 10.1109/ACC.2011.5991284

[11] Lee T., Leok, M., McClamroch, N.H., "Time optimal attitude control for a rigid body," *American Control Conference*, pp. 5210-5215, Westin Seattle Hotel, Seattle, Washington, USA, 11-13 June 2008. doi: 10.1109/ACC.2008.4587322

[12] Cui, P., Zhong, W., Cui, H., "Onboard Spacecraft Slew-Planning by Heuristic State-Space Search and Optimization," *International Conference on Mechatronics and Automation*, pp. 2115-2119, Harbin, China, August 5-8, 2007. doi: 10.1109/ICMA.2007.4303878

[13] Lai, L.-C., Yang, C.-C., Wu, C.-J., "Time-optimal maneuvering control of a rigid spacecraft," *Acta Astronautica*, Vol. 60, No. 10-11, pp. 791-800, May-June 2007. doi: 10.1016/j.actaastro.2006.09.039

[14] Yang, X.-S., "Engineering Optimization - An Introduction with Metaheuristic Applications," John Wiley & Sons, Inc., Hoboken, Chaps. 1,2,15. New Jersey, 2010.

[15] Melton, R. G., "Hybrid methods for determining time-optimal, constrained spacecraft reorientation maneuvers," *Acta Astronautica*, Vol. 94, No. 1, pp. 294-301, January-February 2014. doi:10.1016/j.actaastro.2013.05.007

[16] Komfeld, R. P., "On-board Autonomous Attitude Maneuver Planning for Planetary Spacecraft using Genetic Algorithms," *AIAA Guidance, Navigation and Control Conference*, August 11-14, 2003, Austin, Texas.

[17] Siliang, Y., Shijie, X., "Spacecraft attitude maneuver planning based on particle swarm optimization," *Journal of Beijing University of Aeronautics and Astronautics*, 2010-01, School of Astronautics, Beijing University of Aeronautics and Astronautics, Beijing 100191, China

[18] Showalter, D. J., Black, J. T., "Responsive Theater Maneuvers via Particle Swarm Optimization," *Journal of Spacecraft and Rockets*, Vol. 51, No. 6, pp. 1976-1985. November-December 2014. doi: 10.2514/1.A32989

[19] Huang, P., Liu, G., Yuan, J., Xu, Y., "Multi-Objective Optimal Trajectory Planning of Space Robot Using Particle Swarm Optimization," *Advances in Neural Networks - ISNN 2008*, Vol. 5264, pp. 171-179, 2008. doi: 10.1007/978-3-540-87734-9-20

[20] Xia, N., Han, D., Zhang, G. F., et al, "Study on attitude determination based on discrete particle swarm optimization," *Science China Technological Sciences*, Vol. 53, pp. 3397-3403, December 2010. doi: 10.1007/s11431-010-4148-4

[21] Spiller, D., Ansalone, L., Curti, F., "Particle Swarm Optimization for Time-optimal Spacecraft Reorientation with Keep-out Cones," *Journal of Guidance, Control, and Dynamics* , accessed September 22, 2015. doi: http://arc.aiaa.org/doi/abs/10.2514/1.G001228.

[22] Kennedy, J., Eberhart, R, "Particle Swarm Optimization," *Proceedings of the IEEE International Conference on Neural Networks (Perth, WA)*, Vol. 4, IEEE Service Center, pp. 1942-1948, 1995. doi: 10.1109/ICNN.1995.488968

[23] Clerc, M., "Particle Swarm Optimization," iSTE, Chaps. Introduction, 24 February 2006.

[24] Parsopoulos, K.E., Vrahatis, M.N., "Parameter selection and adaptation in Unified Particle Swarm Optimization," *Mathematical and Computer Modelling*, Vol. 46, No. 1-2, pp. 198-213, Jult 2007. doi: 10.1016/j.mcm.2006.12.019

[25] Kennedy, J., Eberhart, R, "A New Optimizer Using Particle Swarm Theory," *Micro Machine and Human Science, Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43, Nagoya, 4-6 October1995. doi: 10.1109/MHS.1995.494215

[26] Boldrini, F., Procopio, D., Airy, S. P., Giulicchi, L., “Miniaturised Star Tracker (AA-STR) ready to fly,” *Proceedings of the 4S Symposium: Small Satellites, Systems and Services (ESA SP-571)*, 20-24 September, La Rochelle, France. Editor: B. Warmbein. Published on CDROM., id.46.1.

[27] Schmidt, U., Fiksel, T., Kwiatkowski, A., Steinbach, I., Pradarutti, B., Michel, K., Benzi, E., “Autonomous star sensor ASTRO APS: flight experience on Alphasat,” *CEAS Space Journal*, url: http://dx.doi.org/10.1007/s12567-014-0071-z, pp. 1-10, 2015. doi: 10.1007/s12567-014-0071-z

[28] Shuster, M. D., “A Survey of Attitude Representations,” *The Journal of the Astronautical Sciences*, Vol. 41, No. 4, October - December 1993, pp. 439-517.

[29] Kuri-Morales A. F., Gutiérrez-Garcia, J., “Penalty Function Methods for Constrained Optimization with Genetic Algorithms: A Statistical Analysis,” *MICAI 2002: Advances in Artificial Intelligence*, Vol. 2313, pp. 108 - 117, 2002. doi: 10.1007/3-540-46016-0-12

[30] Gen, M., Cheng, R., “A Survey of Penalty Techniques in Genetic Algorithms,” *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Service Center, pp. 804 - 809, Nagoya, 20-22 May 1996. doi: 10.1109/ICEC.1996.542704

[31] Yeniay, Ö., “Penalty Function Methods for Constrained Optimization with Genetic Algorithms,” *Mathematical and Computational Applications*, Vol. 10, No. 1, 2005, pp. 45-56.

[32] Rao, A. V., Benson, D. A., Darby, C. L., Patterson, M. A., Francolin, C., Sanders, I., Huntington, G. T., “Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method,” *ACM Transactions on Mathematical Software*, Vol. 37, No. 2, Article 22, 39 pages, April-June 2010.

[33] De Boor, C., “A Practical Guide to Splines (Applied Mathematical Sciences),” *Springer*, December 2001, Chaps. 9-11. ISBN-10: 0387953663