Constrained Attitude Maneuvering via Modified-Rodrigues-Parameter-Based Motion Planning Algorithms

Riccardo Calaon* and Hanspeter Schaub†
University of Colorado Boulder, Boulder, Colorado 80303

https://doi.org/10.2514/1.A35294

The attitude dynamics and control of a spacecraft becomes complex in the presence of constraints in its orientation. Such constraints consist in not pointing sensitive payload toward bright objects in space or, vice versa, keeping the same bright objects within some instrument's field of view. This paper investigates the applicability of nonsingular attitude motion planning algorithms to navigate a three-dimensional grid in the Modified Rodrigues Parameter configuration space. The aim is to compute a constraint-compliant reference trajectory in Modified Rodrigues Parameter space that can be tracked to reorient a spacecraft from an initial attitude to a final attitude while also attempting at minimizing the required control effort. Nonzero initial and final angular rates are considered to yield a general solution. The path is constructed with B-spline curves that pass through each attitude waypoint. Further, a constant angular maneuver rate is imposed, which influences the speed at which the reference attitude traverses the B-spline solution. Simulations illustrate that constraint-compliant, control-torque-minimizing paths are found if the grid spacing is chosen to be fine enough.

I. Introduction

SPACECRAFT reorientation and maneuvering is a task that becomes nontrivial when combined with constraints in the spacecraft's orientation. Such constraints might be represented by the necessity of not pointing sensitive payload like a telescope or a star tracker toward bright objects like the sun, the moon, or Earth's albedo: these are identified as keep-out constraints. Furthermore, the spacecraft might be required to maneuver such that a certain celestial object remains within the field of view of a body-mounted instrument: this means, for example, to keep the sun direction within a certain angular distance from the boresight direction of sun sensor, or keep the incidence angle of sunlight on solar panels to a minimum to ensure continuous and sufficient power generation. This second type of constraints is classified as keep-in constraints.

The constrained attitude maneuvering problem has been investigated in the past, but an exhaustive solution has yet to be provided. Some authors proposed a solution that is based on Lyapunov functions coupled with barrier potential functions that can repel the spacecraft from the obstacles [1–5]. These implementations have a low computational load and provide smooth, differentiable trajectories. However, they can fail in the presence of nonconvex obstacles that can lure the trajectory into local minima, causing the spacecraft to remain stuck in the wrong configuration [6]. Moreover, these approaches do not attempt to minimize the required control effort or angular rates, which might in certain cases exceed admissible bounds. Other approaches found in literature are based on path-planning algorithms: these approaches do not suffer the problem of local minima. Frazzoli et al. [7] provide a solution based on probabilistic roadmaps, which build incremental trees based on a randomized search of the workspace. This approach provides smooth trajectories to the target and guarantees probabilistic completeness. However, the random nature of the algorithm does not guarantee to

always converge to a uniquely optimal trajectory. Kjellberg and Lightsey [8,9], Tanygin [10], and Tan et al. [11] propose instead path-planning algorithms that are based on discretizations of the workspace, which eliminate the random component in the planner. Tanygin's approach aims at achieving boresight alignment with a target, regardless of the twist component of the attitude around the boresight direction, reducing the problem to a two-dimensional (2D) path-planning query. In this 2D space, he computes a solution based on shortest angular displacement along the path. Kjellberg and Lightsey propose a solution to the full three-dimensional (3D) orientation of the spacecraft where the final path also aims at minimizing the total angular displacement and, at the same time, penalizing deviations of the spacecraft from a direction of motion. The trajectories that they compute are not smooth, but rather piecewise continuous as they connect waypoints in the attitude space. Ultimately, while an attempt is made at reducing the required control effort by discouraging deviations in the trajectory, no guarantee is provided that the computed trajectory is effort-optimal. A novel approach among the path-planning-based solution is the one proposed by Tan et al., where emphasis is given to the smoothing of the path obtained as a sequence of waypoints outputted by the path-planning routine. The smoothing technique is obtained by means of a smooth and differentiable interpolating function in the $SO(3)$ space, which provides a reference trajectory that can be tracked by the spacecraft.

The problem addressed by this paper is the reorientation of a spacecraft from an initial attitude and angular rate to a final attitude and angular rate. The objective is computing a reference frame trajectory that is also constraint-compliant in all of its parts. This is done navigating a graph consisting of a large number of nodes in 3D Modified Rodrigues Parameter (MRP) space, which represents the discrete space of admissible, constraint-compliant attitudes. The MRP shadow set properties are exploited to yield a globally nonsingular attitude path solution. Using MRPs is interesting for this application as the dual set of original and shadow MRP sets allows for a nondimensional attitude description using a minimal three-parameter coordinate set. As a result the search algorithm can traverse a regular 3D space but must account for the discontinuity where the MRP norm exceeds unity. The A* algorithm is applied to compute a valid trajectory that is optimal with respect to a cost function that is representative of the total control effort required to perform the slew maneuver. However, note that the general attitude path-planning methodology developed here is not tied to the A* method, and other search methods could be applied. Given a set of MRP waypoints, the novel aspect of this work is that a smooth attitude path is sought which minimizes the control effort required to connect two arbitrary orientations with arbitrary initial and final angular rates. The use of B-spline curves is explored to yield a twice differentiable MRP

*Ph.D. Student, Ann and H. J. Smead Department of Aerospace Engineering Sciences, 3775 Discovery Drive, 429 UCB-CCAR. Student Member AIAA.

†Professor, Glenn L. Murphy Chair in Engineering, Ann and H. J. Smead Department of Aerospace Engineering Sciences, 3775 Discovery Drive, 429 UCB-CCAR. Fellow AIAA.

trajectory where the control effort can be evaluated. The guidance goal is to have the spacecraft rotate at a nominally fixed angular rate to have a predictable velocity bound. This research explores how to modify the MRP B-spline solutions to yield such rotations that comply with the stated initial and final conditions while satisfying the conic inclusion and exclusion constraints. In this study, the assumption is made that the constraints' positions remain fixed with respect to the spacecraft's orbital frame. This assumption is acceptable in the context where the maneuver time is small enough compared to other time scales involved. The challenge posed by time-varying constraints will be addressed in future related work.

The paper is structured as follows: The first section describes MRPs as an attitude representation set, as well as all the related kinematic equations that are used in the paper, together with the discretization of the attitude space and obstacle representation in MRP configuration space. The A* algorithm is described as a graph search algorithm in its simplest form. The second section starts from a simple, distance-based implementation of the A* algorithm as a proof-of-concept, and proceeds describing an interpolation technique to navigate the space between the waypoints. Subsequently, a method is described to navigate the interpolated trajectory with a constant angular velocity magnitude. Lastly, the interpolated, constant-rate trajectory is incorporated to the A* algorithm to provide a technique to search the graph for the lowest-effort trajectory. The final section shows a variety of case scenarios and the optimal trajectories computed for them, with different sets of keep-out and keep-in constraints.

II. MRP Workspace Discretization

A. Modified Rodrigues Parameters

MRPs are a minimal, 3D set of parameters that represent the attitude of a rigid body rotating in the $SO(3)$ space. A single MRP set σ is obtained from a stereographic projection of the four-dimensional (4D) Euler parameter (quaternion) set β onto a 3D hyperplane [12]. Defining $(\hat{e}, \Phi)$ as the principal rotation vector and angle in a rigid body rotation, the MRP set is obtained as

$$\sigma = \frac{1}{1 + \cos\left(\frac{\Phi}{2}\right)} \begin{Bmatrix} e_1 \sin\left(\frac{\Phi}{2}\right) \\ e_2 \sin\left(\frac{\Phi}{2}\right) \\ e_3 \sin\left(\frac{\Phi}{2}\right) \end{Bmatrix} = \hat{e}\tan\left(\frac{\Phi}{4}\right) \quad (1)$$

The advantage of using MRPs over EPs is that the first are a 3D, constraint-free set, which means that any point in $\boldsymbol{R}^3$ represents a valid attitude. In contrast, EPs only represent a valid attitude when $\boldsymbol{\beta}$ lies on the surface of the 4D hypersphere $\|\boldsymbol{\beta}\| = 1$. On the other hand, reducing the dimension of the parameterization set from four to three causes a singularity to appear in the MRP formulation when the set describes a 360 deg rotation. However, the singularity in the MRP formulation can be avoided by choosing the appropriate MRP set for attitude representation. For every principal rotation set (e, Φ) there exists a shadow set (e, Φ') with $\Phi' = \Phi - 2\pi$ that represents the same attitude with respect to the origin. Such shadow set is mapped into the shadow-set MRP:

$$\sigma_i^S = \frac{e_i \sin\left(\frac{\Phi'}{2}\right)}{1 + \cos\left(\frac{\Phi'}{2}\right)} = \frac{e_i \sin\left(\frac{\Phi}{2} - \pi\right)}{1 + \cos\left(\frac{\Phi}{2} - \pi\right)} = \frac{-\sigma_i}{\sigma^2}$$

$$\text{for } i = 1, 2, 3 \quad (2)$$

where $\sigma^2 = \|\boldsymbol{\sigma}\|^2$. Therefore, any two MRP sets for which Eq. (2) holds true are indeed different MRP sets, which, however, represent the same attitude. Another interesting property that can be inferred from Eq. (2) is that any MRP set whose norm is higher than 1 ($\sigma > 1$)

has a corresponding shadow set whose norm is smaller than one ($\sigma^S < 1$): this means that for any MRP set represented by a point outside a unit 3D sphere centered in the origin, there exists a shadow MRP set within the same unit sphere that represents the same attitude. For points on the boundary of the unit sphere, such that $\sigma = 1$, it is $\sigma^S = -\sigma$. For all the above, the operational domain for the representation of the attitude of a rigid body can be restricted, without loss of generality, to the points in $\boldsymbol{R}^3$ contained in a sphere of radius 1, including boundary points.

The restriction of the workspace to the unit MRP sphere is advantageous for several reasons. First of all, although MRPs are a nonunique set, this choice de facto removes the nonuniqueness by always choosing to refer the short rotation set described by Eq. (1) with $|\Phi| \leq \pi$. This way, the issue of attitude unwinding, which arises with naive control laws based on nonsingular attitude representations [5,13], is bypassed. Moreover, the versatility of the MRPs relies on the fact that they do not have to satisfy any constraint, such as norm unity for EPs or matrix orthogonality for pure SO(3) representations. This property becomes useful in path-planning-based approaches, because any continuous curve that connects any two waypoints in MRP space represents a valid, continuous reference trajectory, allowing for easy interpolation between waypoints. The same cannot be said for EP-based [9] and SO(3)-based [11] attitude discretization strategies, where the corresponding constraints must be enforced between waypoints.

The following differential kinematic equations are used to switch between the time derivatives of the MRP sets $\dot{\boldsymbol{\sigma}}$ and $\ddot{\boldsymbol{\sigma}}$, and the angular rates and accelerations ${}^B\boldsymbol{\omega}_{BN}$ and ${}^B\dot{\boldsymbol{\omega}}_{BN}$ of the body-fixed frame with respect to the origin frame, in $\mathcal{B}$-frame components. The differential kinematic equations for the angular rates are

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}[B(\boldsymbol{\sigma})]^B\boldsymbol{\omega} \quad (3)$$

$$^B\boldsymbol{\omega} = \frac{4}{(1 + \sigma^2)^2}[B(\boldsymbol{\sigma})]^T\dot{\boldsymbol{\sigma}} \quad (4)$$

where the $[B(\boldsymbol{\sigma})]$ matrix is defined as

$$[B(\boldsymbol{\sigma})] = (1 - \sigma^2)[\boldsymbol{I}_{3\times3}] + 2[\tilde{\boldsymbol{\sigma}}] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^T \quad (5)$$

with $[\boldsymbol{I}_{3\times3}]$ being the identity matrix and $[\tilde{\boldsymbol{\sigma}}]$ the skew-symmetric matrix cross-product operator computed from $\boldsymbol{\sigma}$. To obtain the relation between $\ddot{\boldsymbol{\sigma}}$ and angular acceleration, Eqs. (3) and (4) must be differentiated, for which purpose it is useful to define the derivative of $[B(\boldsymbol{\sigma})]$:

$$[\dot{B}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}})] = (-2\boldsymbol{\sigma}^T\dot{\boldsymbol{\sigma}})[\boldsymbol{I}_{3\times3}] + 2[\dot{\tilde{\boldsymbol{\sigma}}}] + 2(\boldsymbol{\sigma}\dot{\boldsymbol{\sigma}}^T + \dot{\boldsymbol{\sigma}}\boldsymbol{\sigma}^T) \quad (6)$$

The differential kinematic equations for the angular accelerations are

$$\ddot{\boldsymbol{\sigma}} = \frac{1}{4}([B(\boldsymbol{\sigma})]^B\dot{\boldsymbol{\omega}} + [\dot{B}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}})]^B\boldsymbol{\omega}) \quad (7)$$

$$^B\dot{\boldsymbol{\omega}} = \frac{4}{(1 + \sigma^2)^2}[B(\boldsymbol{\sigma})]^T\left(\ddot{\boldsymbol{\sigma}} - \frac{1}{(1 + \sigma^2)^2}[\dot{B}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}})][B(\boldsymbol{\sigma})]^T\dot{\boldsymbol{\sigma}}\right) \quad (8)$$

For a more detailed derivation of MRPs and the related equations, the reader is redirected to Ref. [14].

B. Workspace Discretization

The workspace for the present analysis can be reduced, without loss of generality, to the points contained in a unit sphere centered in the origin. Such workspace is discretized using a 3D Cartesian grid consisting of equally spaced nodes. The fineness level N of the grid is defined as the number of equally spaced nodes contained along each principal Cartesian semi-axis. A visual representation of the Cartesian grid, with a fineness level $N = 6$, is reported in Fig. 1: such representation is 2D, for ease of visualization, and describes an
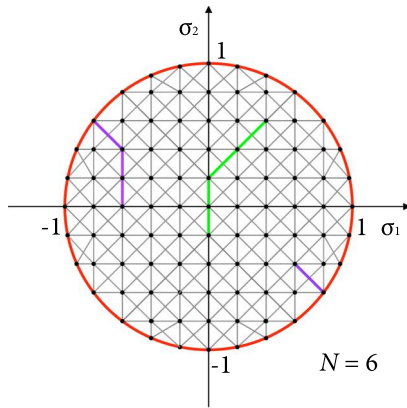
Fig. 1 MRP grid with two minimum-distance paths: i) green: no MRP switching; ii) purple: MRP switching.

obstacle-free workspace. As shown in Fig. 1, nonequally spaced nodes are added on the surface of the unit sphere, in substitution of the neighboring nodes that exceed the limit distance $\sigma = 1$ from the origin, and are therefore excluded. The gray lines connect each node with its immediate neighbors: the definition of neighboring nodes is vital for the pathfinding algorithm, as it defines the directions in which it is allowed to explore the workspace from the current location. In absence of obstacles, all internal nodes (i.e., far enough from the surface of the sphere) have 26 neighboring nodes. A scheme of an internal, 26-connected node is reported in Fig. 2: in purple is the internal node, in red are the neighboring nodes located at a distance of $1/(N - 1)$, in orange the neighbors at a distance of $\sqrt{2}/(N - 1)$, and in yellow the neighbors at a distance of $\sqrt{3}/(N - 1)$. Additionally, each node on the $\sigma = 1$ boundary surface is linked to its own shadow set node, i.e. the node on the opposite side of the sphere with respect to the origin. This allows the pathfinding algorithm to automatically switch to the shadow set when necessary, avoiding principal rotations larger than 180 deg.

This discretization is uniform in MRP space, but it is not uniform when the MRP sets are mapped to the respective principal rotation sets. Equation (1) shows the nonlinear relation between MRP and principal rotation set, for which any two pairs of nodes separated by the same distance in MRP space are not necessarily characterized by the same principal rotation angle Φ. However, since the workspace is bounded to the unit sphere, the principal rotation angle is also bounded in the domain $-\pi \leq \Phi \leq \pi$, for which Eq. (1) is fairly linear. The nonlinearity can be further reduced by using higher-order Rodrigues parameters [15], or higher-order map projections for attitude representation [16,17], at the expenses of having to perform multiple set switching to avoid the singularities introduced by more complex formulations. Tanygin [16] shows that MRPs have better linearity properties than other popular attitude representation sets. Strides have been made to sample the special orthogonal group $SO(3)$ of rigid-body rotations with grid structures that have a high
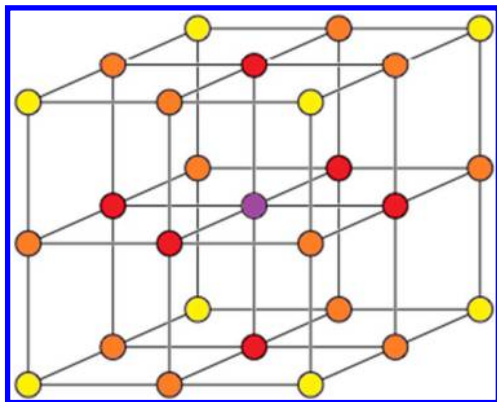
degree of uniformity and coverage of the space [18]. The following work accepts the small deviation from linearity introduced by MRPs. Such nonlinearity introduced by Eqs. (1) and (3) is handled later in this paper, where a constant-angular-rate magnitude $\|\boldsymbol{\omega}\|$ is imposed. This allows to track the reference trajectory with a constant rate of variation of attitude over time.

C. Obstacle Representation

To perform the path-planning query it is essential to map the constraints into the workspace. The constraints are known in terms of the direction of the celestial object $^{\mathcal{N}}\hat{\boldsymbol{s}}_i$, the body-fixed direction of the instrument $^{\mathcal{B}}\hat{\boldsymbol{b}}_i$, and the minimum/maximum angle between the two β_i, according to

$$^{\mathcal{B}}\hat{\boldsymbol{b}}_i \cdot [\mathcal{B}\mathcal{N}]^{\mathcal{N}}\hat{\boldsymbol{s}}_i \lesseqgtr \cos\beta_i \qquad (9)$$

where the $<$ sign is used for keep-out constraints, whereas the $\geq$ for keep-in constraints. Each of these constraints, in their simplest form, can be visualized as a conical region in the unit sphere. For keep-out constraints, the boresight vector of the sensitive instrument must avoid such conical regions as the spacecraft executes the maneuver. On the contrary, for keep-in constraints, the boresight should remain within the conical region during maneuvering.

For every node in the grid, compliance is verified with respect to all the constraints. The nodes that do not satisfy Eq. (9) are marked as not collision-free and removed from the grid. As a consequence, the path-finding algorithm is blocked from exploring nodes that are not constraint-compliant. It should be noted that, with this approach, the segment that connects two constraint-compliant nodes that are very close to an obstacle may itself not be constraint-compliant in all its parts. However, this paper only aims at analyzing simple, convex geometric constraints, for which this phenomenon can be avoided choosing a fineness level N that is sufficiently high.

It is interesting to show how such convex geometric constraints are mapped into obstacles in MRP space. Figure 3a shows the $\mathcal{B}$ frame aligned with the inertial frame $\mathcal{N}$, i.e., with the principal axes aligned with the inertial directions x, y, and z. The x body axis is shown in red, as it coincides with the boresight of the sensitive instrument with a field of view $\beta = 20$ deg. The directions of the three keep-out cones are contained in the (y, z) plane, at an angle of 120 deg from one another. Figure 3b shows the keep out cones mapped in MRP space, where the dot in the origin represents the current attitude $\boldsymbol{\sigma} = [0, 0, 0]$. Since the attitude is constraint-compliant, with the boresight vector outside of all the keep-out cones, so is the dot representing the initial configuration collision-free. It can be observed, for example, that the red keep-out cone blocks negative rotations around the body axis $^{\mathcal{B}}\hat{\boldsymbol{b}}_y = [0, 1, 0]$: in MRP space, this causes the appearance of an obstacle along the negative σ_2 direction. Similar considerations can be made for the other two keep-out cones. General keep-out cones almost always generate tubular structures as those in Fig. 3b: such structures are thicker or thinner according to the half-angle of the respective cone, and always go from one side of the unit sphere to the opposite, because opposite ends of the unit surface represent the same attitudes.

Some qualitatively different obstacles are reported in Fig. 4, where the keep-out cones are 0 deg (yellow) and 180 deg (blue) apart from the boresight direction. It can be observed, first of all, that the boresight vector is inside the yellow cone: as a consequence, the black dot representing the initial attitude in MRP space falls inside the yellow obstacle in Fig. 4b. Since any rotation about $^{\mathcal{B}}\hat{\boldsymbol{b}}_x = [1, 0, 0]$ maintains the boresight within the cone, the corresponding obstacle in MRP space encompasses the whole σ_1 Cartesian axis. As for the blue keep-out cone, any rotation of $\Phi \geq 160$ deg about any body axis perpendicular to $^{\mathcal{B}}\hat{\boldsymbol{b}}_x = [1, 0, 0]$ causes a constraint violation; for this reason the blue obstacle in MRP space takes the shape of a ring distributed around the intersection between the unit sphere and the (σ_2, σ_3) plane.

The results in Figs. 3 and 4 can be generalized to a problem with multiple sensitive instruments and/or sensors characterized by a keep-in constraint. For multiple sensitive instruments, there are multiple boresight vectors: this causes each keep-out cone to generate an
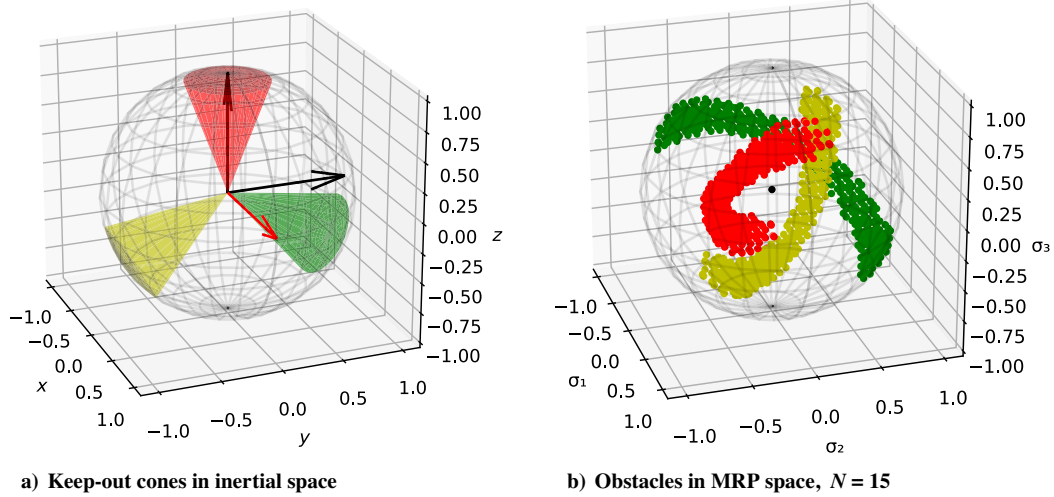

Fig. 2   Neighboring nodes.

a) **Keep-out cones in inertial space**   b) **Obstacles in MRP space, $N = 15$**

Fig. 3   Three general keep-out constraints, $\beta = 20$ deg.

a) **Keep-out cones in inertial space**   b) **Obstacles in MRP space, $N = 15$**
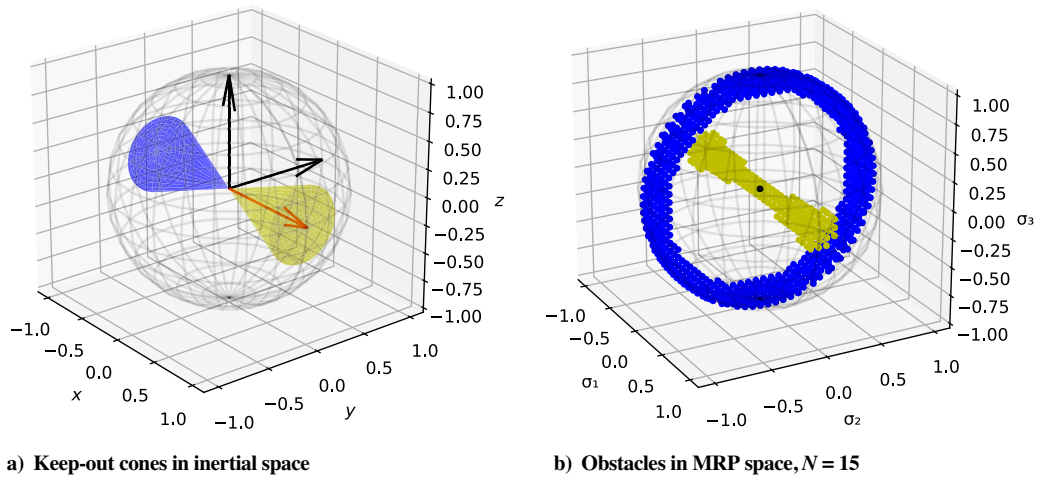
Fig. 4   Two constraints, 0 and 180 deg apart from boresight, $\beta = 20$ deg.

obstacle in MRP space for every sensitive instrument. Such obstacles can have different orientations and sizes, especially if the different instruments have different fields of view, and may overlap in points that represent attitudes for which multiple instruments are facing a keep-out direction. For keep-in constraints, the only difference is the sign in Eq. (9). For this reason, the results are analogous to those of Figs. 3b and 4b; only in this case the colored regions represent constraint-compliant zones, whereas empty regions are not constraint-compliant. Obviously, in the presence of combined keep-out and keep-in constraints, the compliant regions are those where both the constraints are satisfied simultaneously.

III.   Graph Search: The A* Algorithm

A.   Algorithm Review

The primary purpose of this paper is to maneuver a spacecraft from an initial orientation to a final, user-specified orientation. Given the grid discretization and the obstacle representation in MRP space described in the previous section, the maneuvering problem is now transformed into a path-finding query. The aim is to find a path in MRP space that connects the starting point, representing the initial attitude, with the final point, representing the target attitude, avoiding the obstacles that represent the geometric constraints. Such path must be searched in the directed graph that has the grid nodes as graph nodes, and the segments connecting each nodes to its neighbors as graph edges. While this paper uses the A* method to develop a set of discrete attitude path points, the paper's methodology of using

discrete MRP waypoints to develop a smooth, effort-optimized solution is not tied to the A* method generating these points.

First of all, start and goal nodes n_S and n_G must be added to the graph. The graph is searched for the two nodes whose Cartesian distances from n_S and n_G, respectively, are the lowest. To avoid redundancies, such two nodes are replaced by n_S and n_G, which are immediately connected to the neighbors of the nodes they replaced.

Different algorithms exist that can construct a tree (acyclic graph) rooted in n_S, which explores the neighboring nodes until reaching the leaf node n_G. There are two broad categories of such algorithms: breadth-first and depth-first. Breadth-first algorithms proceed exploring, at every iteration, all the nodes located at the same link length from the root [19]. Depth-first algorithms, on the contrary, start from the root node and explore nodes at increasing link lengths at every iteration until a leaf node is found [20]. Breadth-first algorithms have the advantage of exploring larger regions of the graph while running the query; therefore they can find the best path to goal in terms of the smallest link length. Depth-first algorithms, on the contrary, explore much less of the graph, but can find a suboptimal path to goal in less computational time.

Another class of algorithms exists, called greedy algorithms, which at every iteration explore the graph in the direction that appears to be the closest to the goal: an algorithm that belongs to this class is A*. The explored node is chosen as the node having the smallest total cost $f(n) = g(n) + h(n)$, where $g(n)$ is the cost of the path from n_S to n and $h(n)$ is a heuristic estimate of the distance between n and n_G [21]. A* is guaranteed to find the optimal solution in terms of the metric used to define the cost of a path. Moreover, A* is also efficient,

in the sense that its priority-driven nature allows to solve the query exploring the minimum number of graph nodes. For A* to be successful and efficient, the chosen heuristic $h(n)$ should be less than or equal to the actual cost of the path from n to n_G [22]. Two lists are used by the algorithm as it explores the graph: a closed list C containing the nodes that have already been processed, and an open list O containing the nodes that are yet to be processed. The algorithm receives the graph as input, and the open list initially contains only n_S. The pseudo-algorithm for A* is provided in Algorithm 1.

Two aspects should be noted about A* implementation:

i) The algorithm does not stop when n_G is encountered, but only when n_G is encountered as the node with the smallest total cost.

ii) When a node is encountered that is already in the open list with a higher total cost, it means that the algorithm found a more cost-efficient path to that node; therefore that node and its cost should be updated.

For a more comprehensive discussion about A* and other graph search algorithms, the reader is redirected to Refs. [22,23].

B. Nonsingular MRP-Based A* Implementation

A first, simple implementation of the pathfinding algorithm can be based on A*, as described in the previous section. In this implementation, the cost of paths, edges, and heuristics are based on the distance between nodes in MRP space. Specifically, defining $d(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ the Cartesian distance between nodes, the distance metric implemented in the algorithm is the following:

$$\hat{d}(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2) = \min\{d(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2), d(\boldsymbol{\sigma}_1^S, \boldsymbol{\sigma}_2), d(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2^S), d(\boldsymbol{\sigma}_1^S, \boldsymbol{\sigma}_2^S)\} \quad (10)$$

This metric allows to explore the entire discrete domain while avoiding the singularity introduced by MRPs, by automatically performing MRP switching. According to this metric, nodes on the $\sigma = 1$ boundary have a distance $\hat{d} = 0$ with their respective shadow set, therefore allowing to recognize a path that exits the unit sphere and reenters it from the opposite side as, potentially, the minimum-cost path. In Fig. 1, the green and purple paths are both minimum-distance paths, although the first one is entirely contained in the unit sphere, whereas the latter one is obtained via MRP switching. The cost of a path $p = [\boldsymbol{\sigma}_S, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n]$ is defined, in the following implementation, as

$$g(p) = \sum_{i=0}^{n-1} \hat{d}(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{i+1}) \quad (11)$$

and the heuristic

$$h(\boldsymbol{\sigma}_i) = \hat{d}(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_G) \quad (12)$$

Algorithm 1: A*

Result: Path n_S to n_G
Pick n_{best} such that $f(n_{\text{best}}) \leq f(n) \ \forall \ n$ in O
while $n_{\text{best}} \neq n_G$ **do**
 Move n_{best} from O to C
 for m in *Neighbors* (n) **do**
  if $m \notin O$ **then**
   Add m to O
  Else
   if $g(n_{\text{best}}) + \cos t(n_{\text{best}}, m) < g(m)$ **then**
    Change backpointer of m to point to n_{best}
   End
  End
 End
 Pick n_{best} such that $f(n_{\text{best}}) \leq f(n) \ \forall \ n$ in O
End
Backtrack path from n_G to n_S
return Path

The following test scenario features a spacecraft with a sensitive instrument mounted along the x body axis ${}^{\mathcal{B}}\hat{\boldsymbol{b}} = [1, 0, 0]$, which has a field of view of 20 deg. Three bright objects are present, whose inertial directions are ${}^{\mathcal{N}}\hat{\boldsymbol{s}}_1 = [0, -0.981, -0.196]$, ${}^{\mathcal{N}}\hat{\boldsymbol{s}}_2 = [-1, -0, 0]$, and ${}^{\mathcal{N}}\hat{\boldsymbol{s}}_3 = [0.958, 0, 0.287]$. Initial and final attitude, with respect to the inertial frame $\mathcal{N}$, are $\boldsymbol{\sigma}_S = [0, 0, 0.25]$ and $\boldsymbol{\sigma}_G = [0, 0, -0.75]$. Figure 5 shows the computed path in inertial space (a) and in MRP space (b). For ease of visualization, the inertial space is projected onto a 2D plane in terms of longitude and latitude with respect to the inertial frame $\mathcal{N}$.

The computed path is constraint compliant. However, it is only provided in terms of a sequence of waypoints, or attitudes, but it does not give any information on how to navigate between such waypoints. Angular rates and accelerations during the maneuver are unknown. One idea could be to treat every segment in the path as a rest-to-rest rigid body rotation. This approach would ensure constraint compliance, but would likely require a long time to repeatedly accelerate and slow down the spacecraft. Moreover, the computed path is only optimal with respect to the metric used, but it is agnostic to the dynamic properties of the spacecraft: the same path might, in fact, not be equally desirable for spacecraft with different inertia tensors, or different initial angular rates and/or target angular rates. The two-point boundary value problem could be solved to provide guidance between the intermediate points, provided that the angular rates are specified at each waypoint. However, this solution would present discontinuities in the required torque between one segment and the following one, and for a reaction wheel-controlled spacecraft this is not ideal, as an infinite acceleration would technically be required for an immediate change in the wheel angular speed. Ideally, the path should be transformed into a trajectory that interpolates between the waypoints: such trajectory should be differentiable and sufficiently smooth, and should match the required kinematic constraints (angular rates) at both the starting attitude and the target attitude. Such trajectory would allow to provide not just attitude as a function of time, but also rates and accelerations, from which, knowing the inertia properties of the spacecraft, the required torque can be computed.

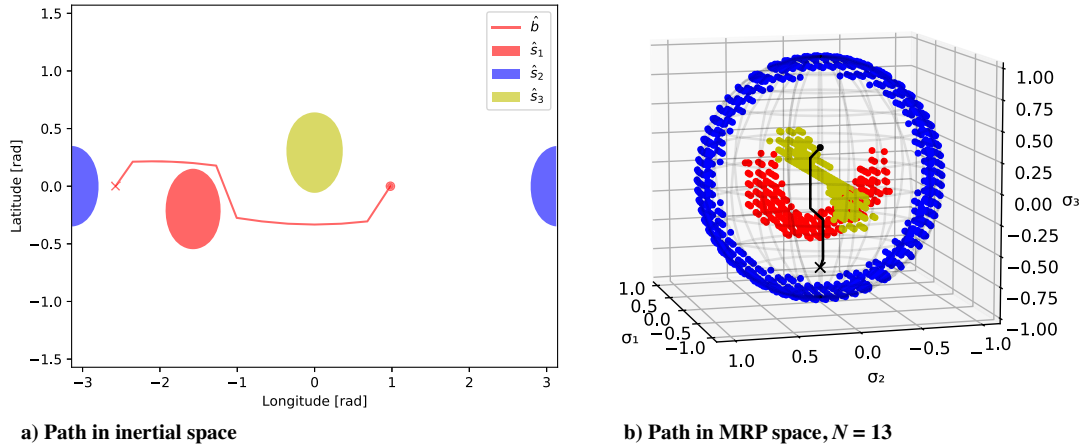## IV. Path Smoothing: B-Spline Interpolation

A. Global Interpolation Through MRP Waypoints

In robot navigation it is quite common to have to deal with paths that present sharp turns that are not suitable for the robot's motion. A strategy that is often used to smooth such paths consists waypoint interpolation. For this implementation, such interpolating function should yield a trajectory that is parametric continuous, i.e., the support of the curve in MRP space should be continuous, and each component σ_i of the MRP set should also be continuous with respect to the parameter of the curve, namely, the time t [24]. A technique that satisfies all the requirements for the interpolation of attitude waypoints is B-spline interpolation. B-spline functions are parametric, piecewise-polynomial functions in the intervals between waypoints. This makes them particularly suitable for this application, as they do not suffer from Runge's phenomenon of oscillation between data points when trying to fit too many data with a single high-order polynomial [25,26]. B-spline functions have the form

$$\boldsymbol{\sigma}(u) = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{P}_i \quad (13)$$

where u is a parameter in the range $u \in [0, 1]$, $\boldsymbol{P}_i$ are called control points, and $N_{i,p}(u)$ are the p th-degree spline basis functions, which are linearly combined to construe the polynomial arcs between waypoints. The degree p of such polynomials is defined by the user, and it can be higher the more waypoints are to be interpolated. The basis functions are defined over the knot vector U containing the $m + 1$ scalar knots:

$$U = \left\{ \underbrace{0, \ldots, 0}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{1, \ldots, 1}_{p+1} \right\} \quad (14)$$

a) Path in inertial space　　　　**b) Path in MRP space, $N = 13$**

Fig. 5　Metric-based A* solution to the path-planning query; "o" is starting point and "x" is target point.

from which the basis functions are computed with the De Boor recursive algorithm [27]:

$$
N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}
$$

$$
N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (15)
$$

For a global interpolation problem with $n + 1$ waypoints and no constraints on the endpoint derivatives, it is possible to build a linear system of $n + 1$ equations based on Eq. (13):

$$
\boldsymbol{\sigma}(\bar{u}_k) = \sum_{i=0}^{n} N_{i,p}(\bar{u}_k) \boldsymbol{P}_i \quad \text{for } k = 0, \ldots, n \quad (16)
$$

where the $\bar{u}_k$'s represent the parametric time at which the curve passes through the kth waypoint $\boldsymbol{\sigma}(\bar{u}_k)$. The $\bar{u}_k$'s can be defined from the total path length in MRP space:

$$
S = \sum_{i=1}^{n} \hat{d}(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{i-1}) \quad (17)
$$

from which

$$
\bar{u}_0 = 0
$$

$$
\bar{u}_k = \bar{u}_{k-1} + \frac{\hat{d}(\boldsymbol{\sigma}_k, \boldsymbol{\sigma}_{k-1})}{S} \quad (18)
$$

The knot vector U used to compute the basis function has size $m + 1$, where $m = n + p + 1$. The intermediate knots u_i can be computed averaging the $\bar{u}_k$'s:

$$
u_0 = \ldots = u_p = 0 \quad u_{m-p} = \ldots = u_m = 1
$$

$$
u_{p+j+1} = \frac{1}{p} \sum_{i=j+1}^{j+p} \bar{u}_i \quad \text{for } j = 0, \ldots, m - 2p - 2 \quad (19)
$$

B.　Global Interpolation with Specified Endpoint Derivatives

Solving Eq. (16) allows to find the $n + 1$ control points $\boldsymbol{P}_i$ that generate a pth-order B-spline curve that passes through the $n + 1$ waypoints $\boldsymbol{\sigma}_k$. However, it is desirable to interpolate the curve while also imposing endpoint derivatives: this allows to account for the dynamic conditions of the spacecraft, i.e., initial angular speed and desired angular speed upon arrival at target attitude. This problem is solved in a similar way, although imposing both endpoint derivatives requires to increase the dimension of the linear system by two. Specifically, two additional control points $\boldsymbol{P}$ are required to produce

the B-spline that interpolates the waypoints and also matches the endpoint derivatives. The linear system takes the form

$$
\boldsymbol{\sigma}(\bar{u}_k) = \sum_{i=0}^{n+2} N_{i,p}(\bar{u}_k) \boldsymbol{P}_i \quad \text{for } k = 0, \ldots, n \quad (20)
$$

whereas Eqs. (18) and (19) remain valid. In this case, the size of the knot vector becomes $m + 1$ with $m = n + p + 3$. Equation (20) provides $n + 1$ equations for $n + 3$ control points $\boldsymbol{P}_i$; therefore two more equations are required to univocally solve the linear system. Such two conditions are obtained from the general rule for the derivative of a B-spline curve:

$$
\boldsymbol{\sigma}^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u) \boldsymbol{P}_i^{(k)}
$$

$$
\boldsymbol{P}_i^{(k)} = \begin{cases} \boldsymbol{P}_i & k = 0 \\ \dfrac{p - k + 1}{u_{i+p+1} - u_{i+k}} \left(\boldsymbol{P}_{i+1}^{(k-1)} - \boldsymbol{P}_i^{(k-1)} \right) & k > 0 \end{cases} \quad (21)
$$

from which one obtains

$$
-\boldsymbol{P}_0 + \boldsymbol{P}_1 = \frac{u_{p+1}}{p} \frac{\mathrm{d}\boldsymbol{\sigma}(0)}{\mathrm{d}u}
$$

$$
-\boldsymbol{P}_{n+1} + \boldsymbol{P}_{n+2} = \frac{1 - u_{m-p-1}}{p} \frac{\mathrm{d}\boldsymbol{\sigma}(1)}{\mathrm{d}u} \quad (22)
$$

Equations (20) and (22) combined provide all the necessary conditions to univocally solve for the $n + 3$ control points $\boldsymbol{P}_i$. Equation (13) can then be used to compute the interpolated trajectory for any parametric time in the interval $u \in [0, 1]$. Moreover, the derivatives of the interpolated trajectory can easily be computed using Eq. (21). The B-spline interpolation allows to specify also the second-order derivative at the endpoints, and/or to specify the derivatives at every waypoint instead of just the endpoints. In each case, the dimension of the linear system would increase by one equation for each specified condition, thus becoming computationally more demanding. A detailed analysis on B-spline functions and their properties can be found in Refs. [25,28].

It is important to notice that the derivatives that appear on the right-hand side of Eq. (22) are computed with respect to the parameter u and not the maneuver time. The time derivatives $\dot{\boldsymbol{\sigma}}_S$ and $\dot{\boldsymbol{\sigma}}_G$ are computed from the initial and final angular rates using Eq. (3), and they relate to the u derivatives according to

$$
\frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}u} = \frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}t} \frac{\mathrm{d}t}{\mathrm{d}u} = \dot{\boldsymbol{\sigma}} \frac{\mathrm{d}t}{\mathrm{d}u} \quad (23)
$$

As a first approximation, it can be assumed that the total maneuver time $t \in [0, T]$ is linearly mapped to the parametric time $u \in [0, 1]$. This way, it is simply

$$\frac{dt}{du} = T \Rightarrow \frac{d\boldsymbol{\sigma}}{du} = \dot{\boldsymbol{\sigma}} \cdot T \qquad (24)$$

T, however, is not known a priori. An estimate of T, to use in Eq. (24), can be obtained using a user-defined value $\bar{\omega}$ that represents the desired average angular rate magnitude during the maneuver. Combining Eqs. (3) and (17) yields

$$T \approx \frac{4S}{\bar{\omega}} \qquad (25)$$

The path computed with the basic A* implementation in the previous subsection is interpolated with the fourth-order B-spline curve, with endpoint derivatives $^{\mathcal{B}}\boldsymbol{\omega}_S = {}^{\mathcal{B}}\boldsymbol{\omega}_G = [0, 0, 0]$, and an average angular rate $\bar{\omega} = 0.03 \ \text{rad}/s$. The angular rates and accelerations are obtained from $\dot{\boldsymbol{\sigma}}$ and $\ddot{\boldsymbol{\sigma}}$ applying Eqs. (4) and (8). Given the inertia tensor $[\boldsymbol{I}]$ of the spacecraft, it is possible to compute the required control torque $\boldsymbol{L}$ as

$$\boldsymbol{L} = {}^{\mathcal{B}}[\boldsymbol{I}]^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{\mathcal{BN}} + {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{BN}} \times ({}^{\mathcal{B}}[\boldsymbol{I}]^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{BN}}) \qquad (26)$$

The interpolated path is shown in Fig. 6. Figure 7 shows the evolution of attitude, rates, accelerations, and torque, along the maneuver, which is performed in about 150 s. The order $p = 4$ is chosen to ensure that the second-order derivative $\ddot{\boldsymbol{\sigma}}$ is at least $\mathcal{C}^1$ [28], which allows to achieve a smooth torque profile. The inertia tensor used to compute the torques corresponds to a three-unit CubeSat with a uniformly distributed mass $m = 4.2$ kg, expressed with respect to the principal body frame

$$^{\mathcal{B}}[\boldsymbol{I}] = \begin{bmatrix} 6.67 & 0 & 0 \\ 0 & 41.87 & 0 \\ 0 & 0 & 41.87 \end{bmatrix} \cdot 10^{-3} \ \text{kg} \cdot \text{m}^2 \qquad (27)$$

The angular rate plot shows also the norm of the angular rate vector, which maintains itself fairly close to the desired $\bar{\omega} = 0.03$ rad/s. The largest accelerations, and torques, are required at the endpoints, where the spacecraft needs to be accelerated/decelerated from/to the rest condition.

A challenge is presented for the interpolating spline when the sequence of waypoints involves MRP switching, because the interpolating $\boldsymbol{\sigma}(t)$ function would have to present a discontinuity. The B-spline routine described above cannot handle such case directly. However, $\boldsymbol{\sigma} \le 1$ is a soft constraint: it is therefore possible to exploit the shadow sets of a subset of the waypoints, which lie outside of the unit sphere, to interpolate the function in a continuous domain. Once the continuous spline is interpolated, the part of the function that lies outside of the $\boldsymbol{\sigma} \le 1$ constraint can be mapped back to the unit sphere. The angular rates and accelerations $^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{BN}}$ and $^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{\mathcal{BN}}$ can be computed using $\dot{\boldsymbol{\sigma}}$ and $\ddot{\boldsymbol{\sigma}}$ from either set. An example of this is reported in Fig. 8: the black curve in part (b) represents the interpolated path within the unit sphere, whereas the gray curve represents the interpolated curve using the shadow sets outside the unit sphere, where the gray "X" represents the shadow set of the target attitude $\boldsymbol{\sigma}_{\text{G}}^S$. When exploiting the points outside the unit sphere, it is particularly important to use the metric described in Eq. (10) to compute the values of parametric time $\bar{u}_k$ in Eq. (18). Nodes outside the unit sphere are
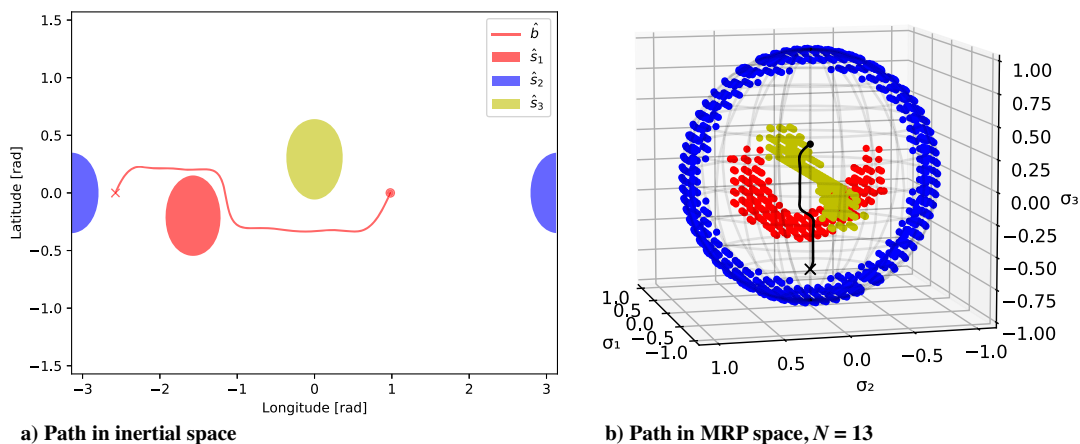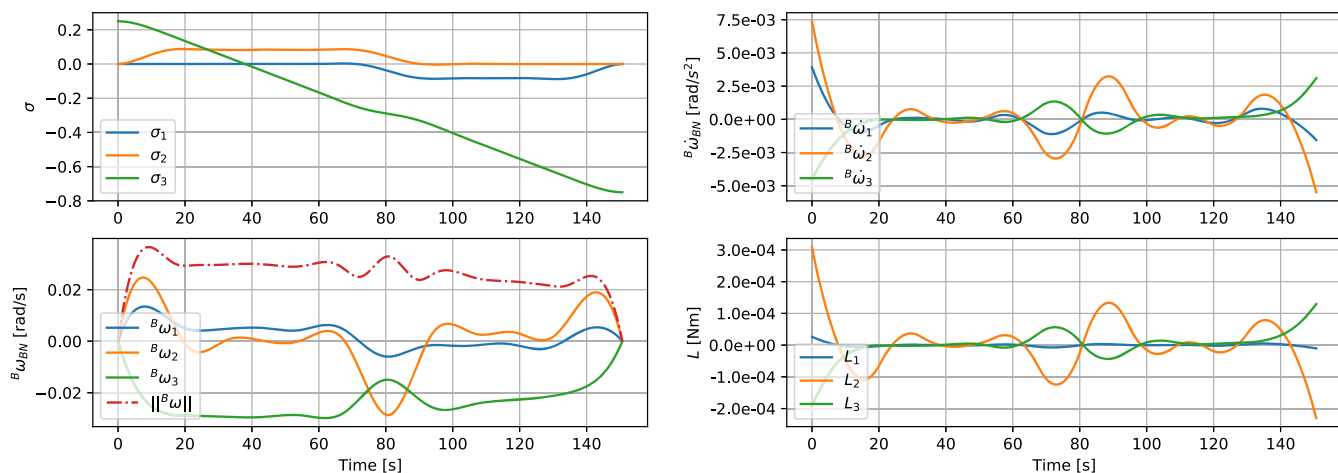

a) Path in inertial space

b) Path in MRP space, $N = 13$

Fig. 6 Interpolated metric-based A* solution to the path-planning query.

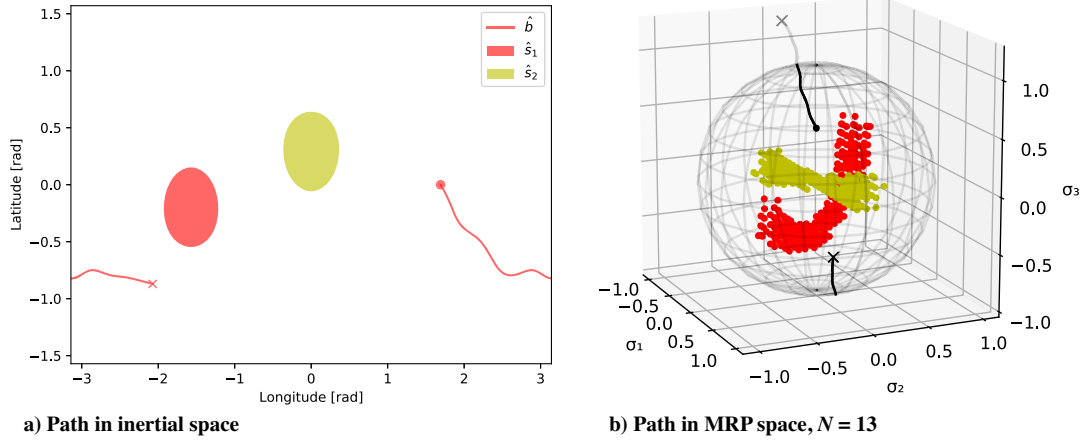Fig. 7 Attitude, rates, acceleration, and torque over time for the trajectory in Fig. 6.

a) Path in inertial space   b) Path in MRP space, $N = 13$

Fig. 8 Interpolated metric-based A* solution involving MRP switching.

farther apart than their corresponding shadow sets. The metric $\hat{d}(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ returns the Cartesian distance between the two corresponding sets of MRPs inside the unit sphere: this maintains the interpolating spline stable and the spacing between waypoints coherent across the function.

The method described above consists in a global interpolation, where the B-spline curve passes through all the $n + 1$ points precisely. However, a path consisting of a high number of very dense waypoints might still cause the interpolated curve to wiggle between the waypoints. Such oscillatory motion is not ideal for the spacecraft, for which a smooth trajectory is desired. To address this issue, it is possible to use B-spline curves to provide a hybrid interpolating–approximating curve. Such curve can be made pass through the endpoints precisely, whereas it tracks all the other waypoints through a least-squares fit, with higher weights given to nodes that are located closer to obstacles. Such approach will be the object of future studies, whereas for the present paper a globally interpolating spline is used.

V. Constant Nominal Angular Rate Maneuvering

The results plotted in Fig. 7 show that the interpolated trajectory does maintain an angular speed magnitude close to the desired one throughout the maneuver. However, such angular speed does not remain constant, but rather it floats around the desired value. When performing a slew maneuver, the desire is usually to execute it as fast as possible. On the other hand, spacecraft often present upper bounds on the maximum angular rate that can be tolerated by the payload. This causes two antithetic requirements to appear, for the maneuver to happen in the lowest time possible, but also for the spacecraft to remain within the bounds of admissible speeds [29]. This subsection aims to elaborate on the results of the previous subsection by ensuring a constant-angular-rate magnitude ω^* as the spacecraft follows the interpolated trajectory, without modifying the trajectory itself. Fixing ω^* to a value that is below the speed limit, automatically bounds all the components ω_i to be within such limit.

The aim of this subsection is to obtain an angular rate $\boldsymbol{\omega}(\theta)$ such that

$$\boldsymbol{\omega}(\theta) = \boldsymbol{\omega}(t) \frac{\mathrm{d}t}{\mathrm{d}\theta} \tag{28}$$

where $\theta(t)$ is a "warped" time that allows to navigate the trajectory $\boldsymbol{\sigma}(\theta)$ at the desired angular rate. The first condition that must be met to ensure that the same trajectory is tracked is that the first integral of the trajectory remains unaltered:

$$\Omega = \int_0^T \|\boldsymbol{\omega}(t)\| \, \mathrm{d}t = \int_0^\Theta \|\boldsymbol{\omega}(\theta)\| \, \mathrm{d}\theta \tag{29}$$

where Ω is computed by numerical integration of $\|\boldsymbol{\omega}(t)\|$ coming from the interpolated trajectory.

Clearly, a constant $\|\boldsymbol{\omega}(\theta)\|$ over the entire domain $\theta \in [0, \Theta]$ cannot match the required conditions at the endpoints. For this reason, the profile of $\|\boldsymbol{\omega}(\theta)\|$ is defined as a piecewise function that is a fourth-order polynomial around the endpoints and constant in the central part of the trajectory. Such piecewise function is defined as

$$\|\boldsymbol{\omega}(\theta)\| = \begin{cases} a_0 + a_1\theta + a_2\theta^2 + a_3\theta^3 + a_4\theta^4 & \text{for } 0 \leq \theta < \theta_a \\ \omega^* & \text{for } \theta_a \leq \theta \leq \theta_b \\ b_0 + b_1(\theta - \Theta) + b_2(\theta - \Theta)^2 + b_3(\theta - \Theta)^3 + b_4(\theta - \Theta)^4 & \text{for } \theta_b < \theta \leq \Theta \end{cases} \tag{30}$$

with

$$\int_0^{\theta_a} \|\boldsymbol{\omega}(\theta)\| \, \mathrm{d}\theta = \frac{1}{10}\Omega$$

$$\int_{\theta_a}^{\theta_b} \|\boldsymbol{\omega}(\theta)\| \, \mathrm{d}\theta = \frac{4}{5}\Omega$$

$$\int_{\theta_b}^{\Theta} \|\boldsymbol{\omega}(\theta)\| \, \mathrm{d}\theta = \frac{1}{10}\Omega \tag{31}$$

The problem now is to determine the coefficients of the polynomials a_i and b_i. The first four are obtained from the endpoint values and first endpoint derivatives of $\|\boldsymbol{\omega}(t)\|$, in order to ensure the same behavior around the endpoints:

$$a_0 = \|\boldsymbol{\omega}(0)\| \quad a_1 = \frac{\mathrm{d}\|\boldsymbol{\omega}(0)\|}{\mathrm{d}t} \tag{32}$$

$$b_0 = \|\boldsymbol{\omega}(T)\| \quad b_1 = \frac{\mathrm{d}\|\boldsymbol{\omega}(T)\|}{\mathrm{d}t} \tag{33}$$

The other six coefficients are obtained imposing continuity ($\|\boldsymbol{\omega}(\theta_a)\| = \|\boldsymbol{\omega}(\theta_b)\| = \omega^*$), together with first- and second-order derivatives equal to 0 for $\theta = \theta_a$ and $\theta = \theta_b$. The fourth-order polynomials, together with these constraints on first- and second-order derivatives in θ_a and θ_b, ensure the continuity and differentiability of both $\|\boldsymbol{\omega}(\theta)\|$ and $\|\dot{\boldsymbol{\omega}}(\theta)\|$. After some cumbersome algebra, this yields

$$a_2 = \frac{6(\omega^* - a_0)}{\theta_a^2} - \frac{3a_1}{\theta_a} \qquad b_2 = \frac{6(\omega^* - b_0)}{\theta_b^2} - \frac{3b_1}{\theta_b} \qquad (34)$$

$$a_3 = \frac{8(a_0 - \omega^*)}{\theta_a^3} + \frac{3a_1}{\theta_a^2} \qquad b_3 = \frac{8(b_0 - \omega^*)}{\theta_b^3} + \frac{3b_1}{\theta_b^2} \qquad (35)$$

$$a_4 = \frac{3(\omega^* - a_0)}{\theta_a^4} - \frac{a_1}{\theta_a^3} \qquad b_4 = \frac{3(\omega^* - b_0)}{\theta_b^4} - \frac{b_1}{\theta_b^3} \qquad (36)$$

At this point, knowing the expressions for all the coefficients, it is possible to analytically integrate Eq. (31), which yields a system of three equations in the variables θ_a, θ_b, and Θ, whose solution is

$$\theta_a = \frac{-6\omega^* - 4a_0 + \sqrt{(4a_0 + 6\omega^*)^2 + 2a_1\Omega}}{a_1} \qquad (37)$$

$$\theta_b = \theta_a + \frac{4\Omega}{5\omega^*} \qquad (38)$$

$$\Theta = \theta_b + \frac{6\omega^* + 4b_0 - \sqrt{(4b_0 + 6\omega^*)^2 - 2b_1\Omega}}{b_1} \qquad (39)$$

Now that the expression for $\|\boldsymbol{\omega}(\theta)\|$ is complete, the relation $\theta(t)$ is found solving the nonlinear equation

$$\int_0^\theta \|\boldsymbol{\omega}(\theta)\| \, d\theta = \int_0^t \|\boldsymbol{\omega}(t)\| \, dt \qquad (40)$$

for every t. The integral on the left-hand side of Eq. (40) is computed analytically, since the expression in the integral is a polynomial. The integral on the right-hand side, however, is computed numerically from the $\|\boldsymbol{\omega}(t)\|$ that is the output of the B-spline interpolation. Carrying out both integrals leads, for $0 \le \theta < \theta_a$ and $\theta_b < \theta \le \Theta$, to a fifth-order equation in the variable θ, which can be solved numerically with Newton–Rhapson's method.

The constant-angular-rate constraint is applied to the basic-A* trajectory described in the previous subsections. Since the geometry of the trajectory is unaltered, Fig. 6 remains a valid representation of the trajectory, whereas the evolution of rates, accelerations, and torque with respect to the new time variable θ are reported in Fig. 9.
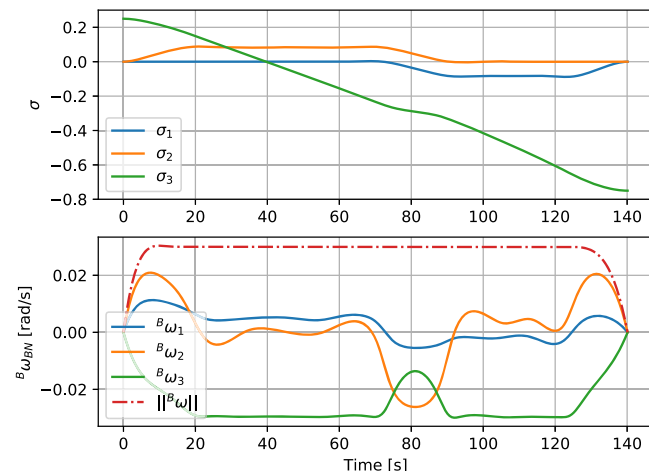
The magnitude of the angular rate vector ramps up and down from zero at the endpoints, and it remains constant at $\|\boldsymbol{\omega}\| = 0.03$ rad/s throughout most of the maneuver. With this correction, the total maneuver time is reduced to 140 s, instead of 150 s as shown in Fig. 7.
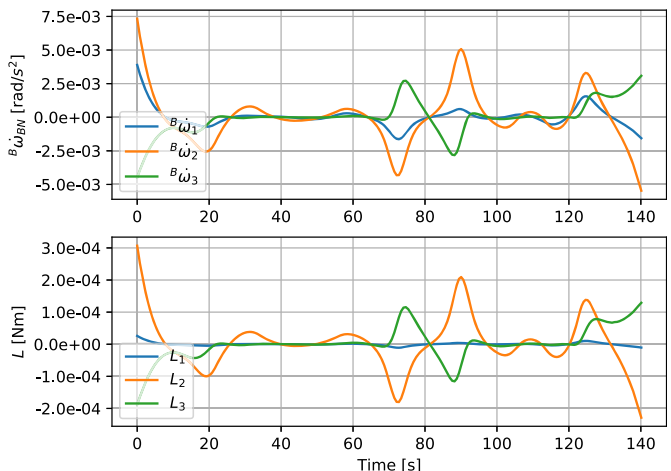
VI. Effort-Based A* Graph Search

The previous subsections allowed to compute a proper trajectory from a path composed by a sequence of waypoints. Until this stage, all the paths were found minimizing the total path length S in terms of cumulative distance between nodes in MRP space [Eq. (17)]. Within the limits of the nonlinearity introduced by Eq. (1), the minimum-S path also minimizes the total cumulative angular displacement of the spacecraft while performing the slew maneuver. From a dynamic perspective, however, this approach is not significant, as it does not account for the dynamic state of the spacecraft or for its inertia distribution. For example, if the initial angular velocity of the spacecraft were directed in the opposite direction with respect to the minimum-S path, tracking that trajectory would potentially require a large initial torque to steer the spacecraft. Moreover, more than one path can exist with the same length S, but they might not be equally feasible to track, depending on the required control torque, which itself depends on the inertia tensor $^B[\boldsymbol{I}]$. For all the above, this section aims to implement a modified version of A* that searches the graph for a path that is optimal in terms of required control effort. This presents some challenges, since the "optimality" of such path, or cost, must be computed in terms of a scalar quantity that can be compared with the cost of other paths. Such cost can be defined as the integral of the torque magnitude over the slew maneuver:

$$\text{cost}(p) = \int_0^T \|\boldsymbol{L}(t)\| \, dt \qquad (41)$$

The cost function described in Eq. (41) is meant to be reasonable for a general application, but is not tied to any specific set of actuators. In a more refined analysis, the actuators' equations of motion can be integrated based on the interpolated trajectory to yield the required reaction wheel control torques as functions of time. The cost function can therefore be modeled to minimize the reaction wheel torque or, alternatively, the power required to actuate them. A study on different types of actuator-based cost functions will be the subject of future related work.

As described previously in the paper, the A* algorithm chooses which nodes to expand based on the sum of the traveled path to current node ($g(n)$) and a heuristic cost from current node to goal ($h(n)$). Moreover, A* is efficient at finding the best path when the heuristic is optimistic, meaning that $h(n)$ should be less than or equal to the actual cost of the path from node n to goal.

Using the cost function described in Eq. (41) makes it challenging to compute $g(n)$ and $h(n)$. One option could be to interpolate two

Fig. 9 Attitude, rates, acceleration, and torque with constant $\|\boldsymbol{\omega}\|$ for the trajectory in Fig. 6.

paths, one going from the start node to n, $p_{[S,n]} = [\boldsymbol{\sigma}_S, \ldots, \boldsymbol{\sigma}_n]$, and the other one from node n to the goal node, $p_{[n,G]} = [\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_G]$. However, the required $\dot{\boldsymbol{\sigma}}_n$ at the intermediate node n is not known; therefore the two interpolated trajectories would have mismatching derivatives at $\boldsymbol{\sigma}_n$. This phenomenon could give an incorrect representation of the total path cost that would otherwise be obtained interpolating a full path from $\boldsymbol{\sigma}_S$ to $\boldsymbol{\sigma}_G$ with a single, smooth trajectory. To overcome this problem, the total cost of a node $f(n) = g(n) + h(n)$ is computed in one single step, applying Eq. (41) to the path

$$p_{[S,n,G]} = [\boldsymbol{\sigma}_S, \ldots, \boldsymbol{\sigma}_n, \boldsymbol{\sigma}_G] \tag{42}$$

For such path, the integral between $\boldsymbol{\sigma}_S$ and $\boldsymbol{\sigma}_n$ corresponds to $g(n)$, whereas the integral between $\boldsymbol{\sigma}_n$ and $\boldsymbol{\sigma}_G$ corresponds to the heuristic $h(n)$:

$$g(n) = \int_{t(\boldsymbol{\sigma}_S)}^{t(\boldsymbol{\sigma}_n)} \|\boldsymbol{L}(t)\| \, \mathrm{d}t \tag{43}$$

$$h(n) = \int_{t(\boldsymbol{\sigma}_n)}^{t(\boldsymbol{\sigma}_G)} \|\boldsymbol{L}(t)\| \, \mathrm{d}t \tag{44}$$

$$\mathrm{cost}(p_{[S,n,G]}) = f(n) = g(n) + h(n) = \int_{t(\boldsymbol{\sigma}_S)}^{t(\boldsymbol{\sigma}_G)} \|\boldsymbol{L}(t)\| \, \mathrm{d}t \tag{45}$$

Such heuristic is optimistic (or admissible [21]) in the sense that it assumes that it is possible to go directly from node n to n_G, neglecting the presence of obstacles along the way. Two arguments can be provided to support that the heuristic is admissible. The first one is

that, in the presence of an obstacle, the spacecraft would have to steer around it, which would require additional torquing effort. Secondly, it should be noted that the grid is an artificial construct. Therefore, forcing the trajectory to hit precisely all the waypoints of a path also requires additional torquing effort. The $h(n)$ defined above is based on the idea of "taking a shortcut" between the current node and goal node, thus avoiding unnecessary steering. Calaon et al. [30] show how the resulting control torque integral is reduced when the requirement of hitting all the waypoints precisely is lifted. Furthermore, the same paper shows evidence that the effort-optimal trajectory computed with the cost function described above yields a smaller control torque integral than a minimum-S-based path-planning query that also applies the spline interpolation. Lastly, Eq. (42) ensures that the cost is computed from a trajectory that matches the requisites of continuity and differentiability described in the previous subsections.

When computing the cost of such path, unless node n and goal node are neighbors, the distance $\hat{d}(\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_G)$ is higher than the average distance between neighboring nodes. When this mismatch in the spacing between nodes is significant, it can generate instabilities in the interpolating function, causing the spacecraft to overshoot the target. To circumvent this problem, when $\hat{d}(\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_{\mathrm{goal}}) > (\sqrt{3}/(N-1))$, a number of equally spaced "guidance nodes" with mutual distance $\approx (1/(N-1))$ are artificially added between nodes n and n_G to stabilize the interpolation.

The effort-based A* algorithm is run in the same scenario described in the previous subsections, with zero endpoint angular rates. The resulting interpolated path is reported in Figs. 10 and 11. As intuition suggests, the path computed by the regular A* in Fig. 5 is not the preferred one in terms of smallest effort. The effort-optimal path steers the spacecraft around the yellow obstacle in the first part of
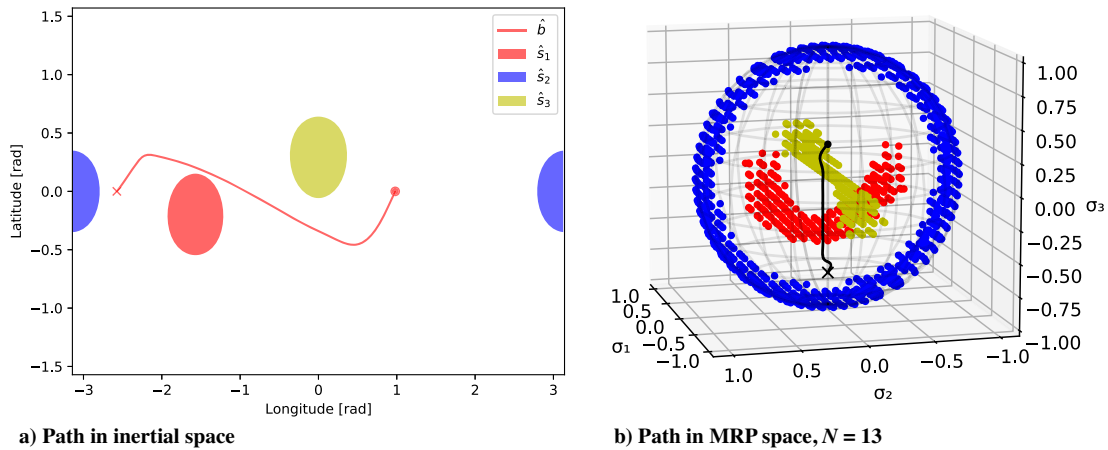

a) Path in inertial space

b) Path in MRP space, $N = 13$

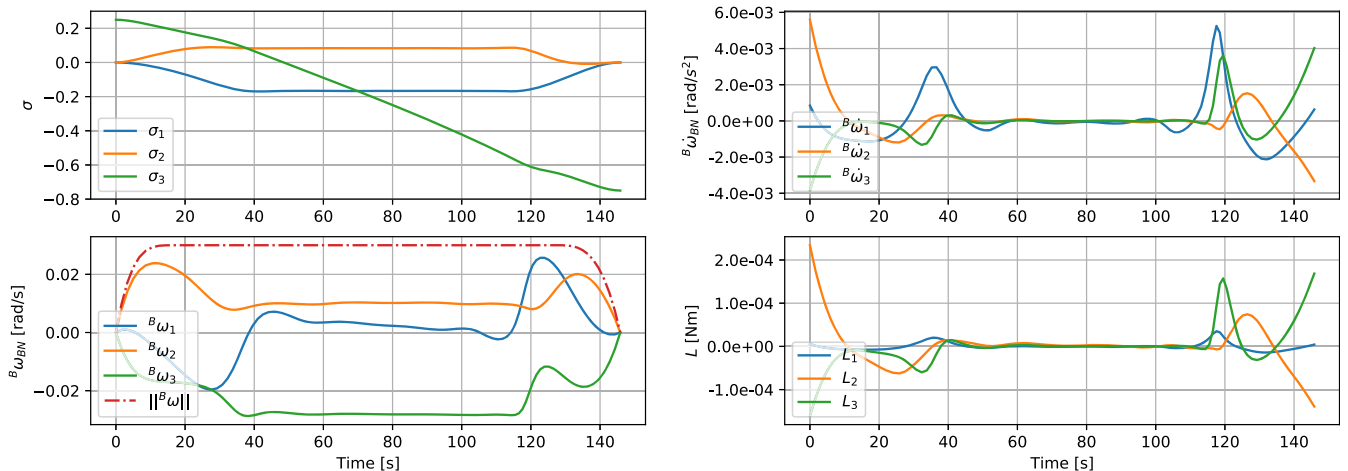Fig. 10 Interpolated effort-based A* solution.

Fig. 11 Attitude, rates, acceleration, and torque over time for the trajectory in Fig. 10.

the trajectory, to put the spacecraft onto a coasting arc between $t \approx$ 50 s and $t \approx 110$ s, where the torques remain close to zero. Finally, after the quasi-coasting arc, the spacecraft is steered toward the target node. The total cost of the effort-optimal path is $\mathrm{cost}(p_{\mathrm{effort-optimal}}) =$ $5.53 \cdot 10^{-3}$ N·m·s, as opposed to the S-optimal path $\mathrm{cost}(p_{S\text{-optimal}})$ $= 8.72 \cdot 10^{-3}$ N·m·s.

VII.   Numerical Performance Study

A.   Long Rotation and MRP Switching Scenario

An interesting case is obtained when, from the test scenario analyzed in the previous sections, the blue obstacle is removed. Doing so frees a path from initial to target attitude that consists of a simple eigenaxis rotation about $\hat{\boldsymbol{b}}_z$. Intuitively, such path should be the most cost-efficient for a rest-to-rest maneuver, since it could entirely avoid the obstacles and the insurgence of gyroscopic terms in the required torque. The effort-optimal trajectory is reported in Figs. 12 and 13. For a rest-to-rest case with a constant angular rate $\|\boldsymbol{\omega}\| = 0.03$ rad/s, the trajectory describes a bang-bang type of actuation along the $\boldsymbol{b}_z$ axis only, where the spacecraft is accelerated to the desired speed and decelerated before reaching the target, while it coasts effortlessly during most of the trajectory. The attitude plot shows MRP switching in correspondence of $\boldsymbol{\sigma} = [0, 0, 1]$ and follows from the shadow-set $\boldsymbol{\sigma}^S = [0, 0, -1]$. The same behavior is mirrored in Fig. 12b, where the trajectory exits the unit MRP sphere and reenters from the opposite side. The total cost of the path is $\mathrm{cost}(p_{\mathrm{effort-optimal}}) = 2.50 \cdot 10^{-3}$ N·m·s.

B.   Non-Rest-to-Rest Scenario

The previous sections have shown the development of the effort-based A* algorithm and ultimately applied it to find the most cost-efficient rest-to-rest trajectory given $\boldsymbol{\sigma}_S = [0, 0, 0.25]$ and $\boldsymbol{\sigma}_G = [0, 0, -0.75]$ and a set of three keep-out constraints. One of the advantages of an effort-based approach for pathfinding is that it optimizes not only according to the spacecraft's inertia, but also according to its initial (and final) conditions. This scenario aims to show the optimal trajectory for the same workspace, but with the spacecraft having an initial angular velocity $^B\boldsymbol{\omega}_S =$ $[0, 0, 0.03]$ rad/s ($\dot{\boldsymbol{\sigma}}_S = [0, 0, 0.063]$) and a target angular velocity $^B\boldsymbol{\omega}_G = [0, 0, 0]$ rad/s ($\dot{\boldsymbol{\sigma}}_G = [0, 0, 0]$). The resulting trajectory and the relative attitude, rates, accelerations, and torques are reported in Figs. 14 and 15. The total cost of the path is $\mathrm{cost}(p_{\mathrm{effort-optimal}}) = 5.45 \cdot 10^{-3}$ N·m·s. Clearly, the path takes advantage of the initial angular velocity of the spacecraft, letting it coast along the velocity direction until it starts being deflected when it gets close to the blue obstacle.

C.   Varying MRP Grid Step Sizes

All the previous results have been obtained with an MRP grid fineness level $N = 13$. Increasing the fineness level provides a more accurate description of the obstacles in MRP space and reduces the risk of obtaining a path that violates the constraints in the arc between waypoints. However, the total number of nodes grows with N^3, making the graph search slower as N increases. Specifically, the rest-to-rest case described in Fig. 10 requires processing 632 nodes before finding the optimal trajectory, whereas the non-rest-to-rest
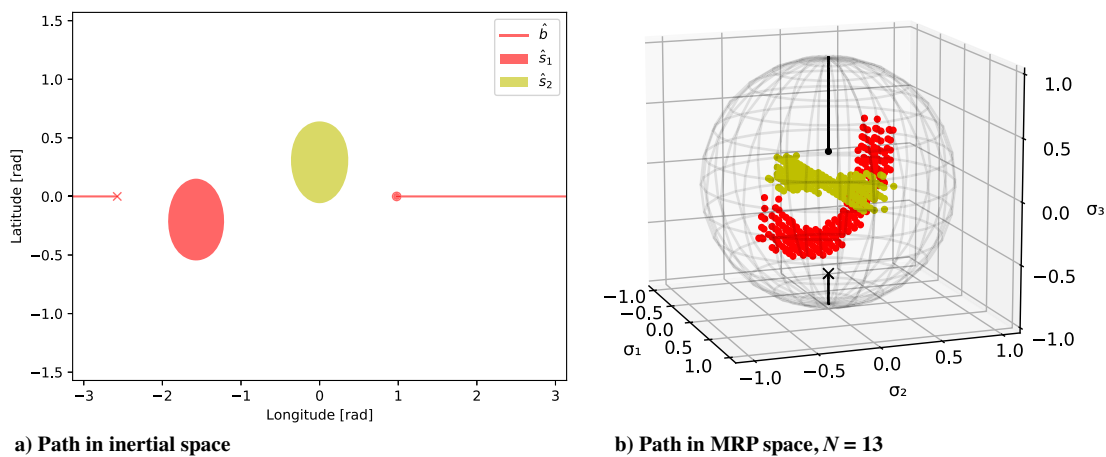

a) Path in inertial space

b) Path in MRP space, $N = 13$

Fig. 12   Interpolated effort-based A* solution, eigenaxis rotation, and MRP switching scenario.
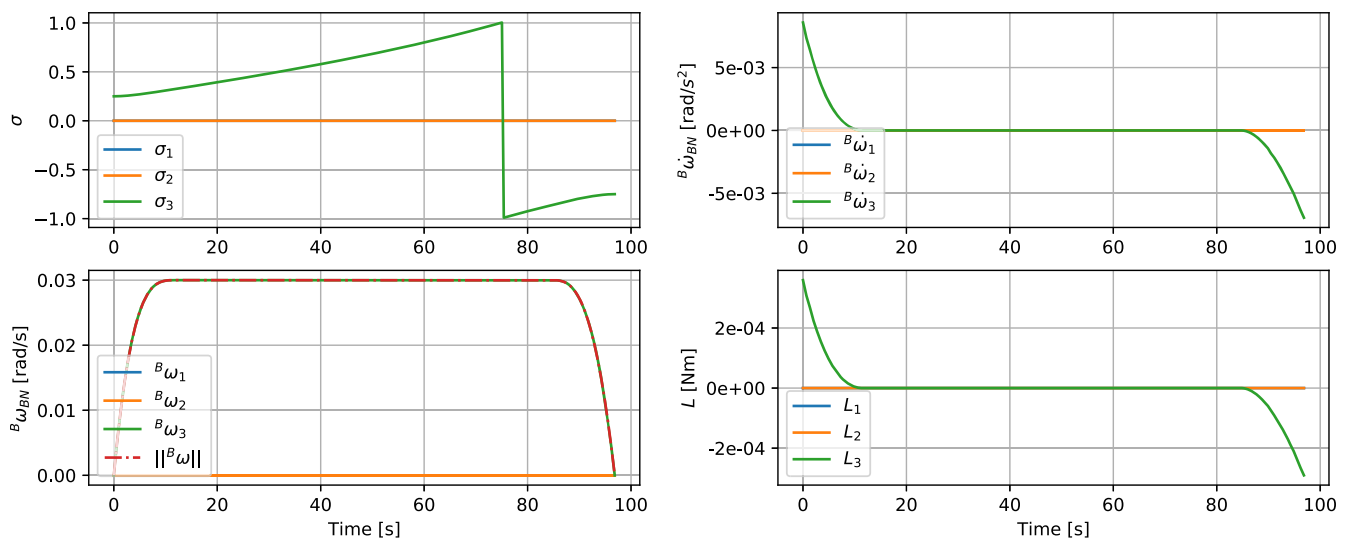
Fig. 13   Attitude, rates, acceleration, and torque over time for the trajectory in Fig. 12.
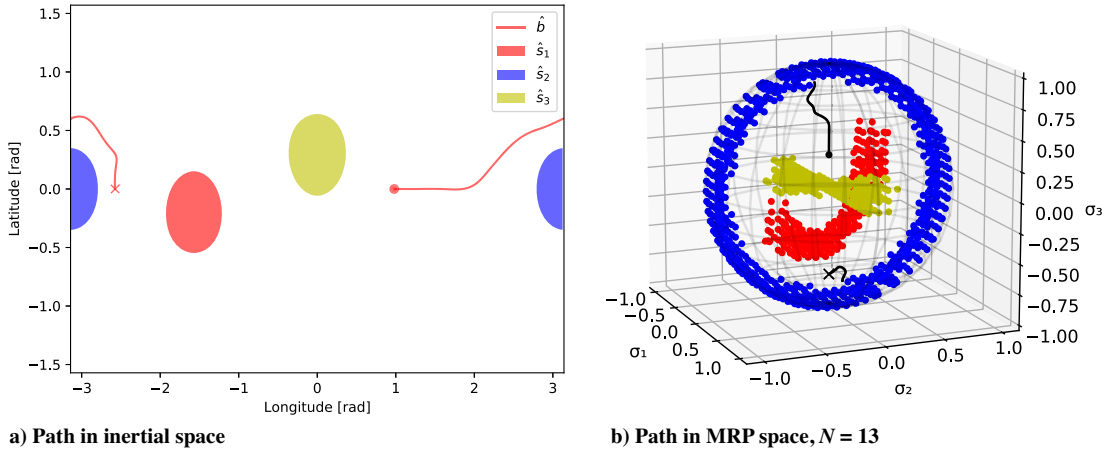
a) Path in inertial space

b) Path in MRP space, N = 13

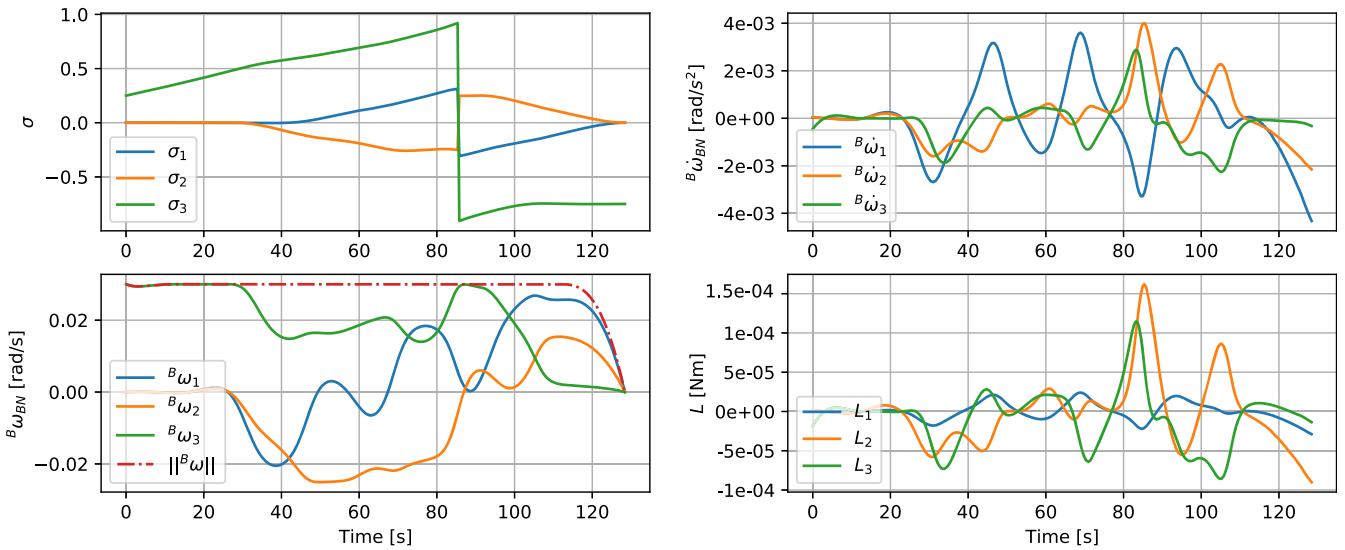**Fig. 14   Interpolated effort-based A* solution for non-rest-to-rest case.**

Fig. 15   Attitude, rates, acceleration, and torque over time for the trajectory in Fig. 14.

case in Fig. 14 requires to process 580 nodes. Moreover, for every processed node, the algorithm must interpolate a trajectory, warp such trajectory to obtain a constant-angular-rate profile, and numerically integrate the torque to provide the cost of the trajectory. All these computations sensibly slow down the algorithm. It is interesting to analyze the same problem with a different fineness level, to see how this impacts the computed solution. Figure 16 shows the trajectory

computed with a fineness level N = 7. The trajectory is still constraint-compliant, and still takes advantage of the natural motion of the spacecraft in the beginning. However, Fig. 17 shows that the angular rates start changing from the very beginning, as compared to Fig. 15 where they remained approximately constant for about 20 s. Comparing the torque profiles, it looks like the coarser grid also allows for smoother, less oscillating torques. This finds justification
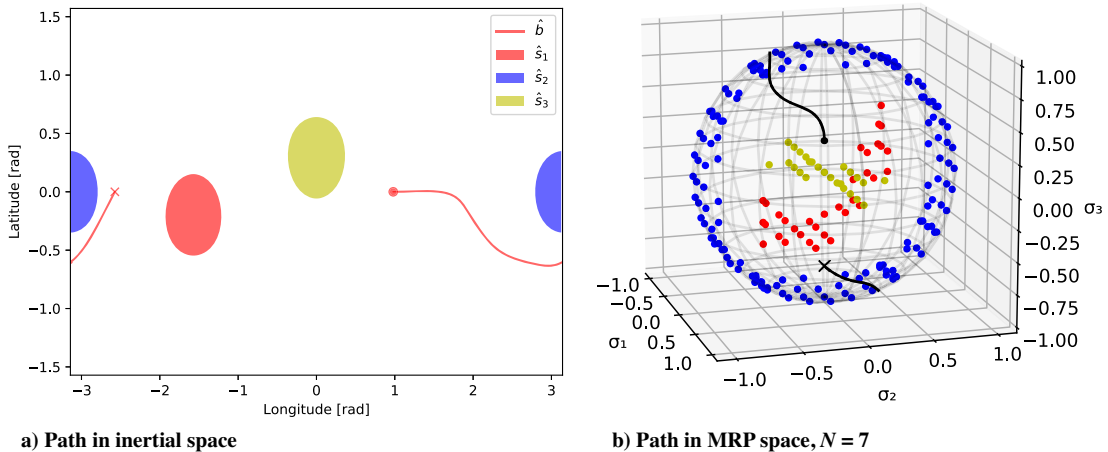

a) Path in inertial space

b) Path in MRP space, N = 7

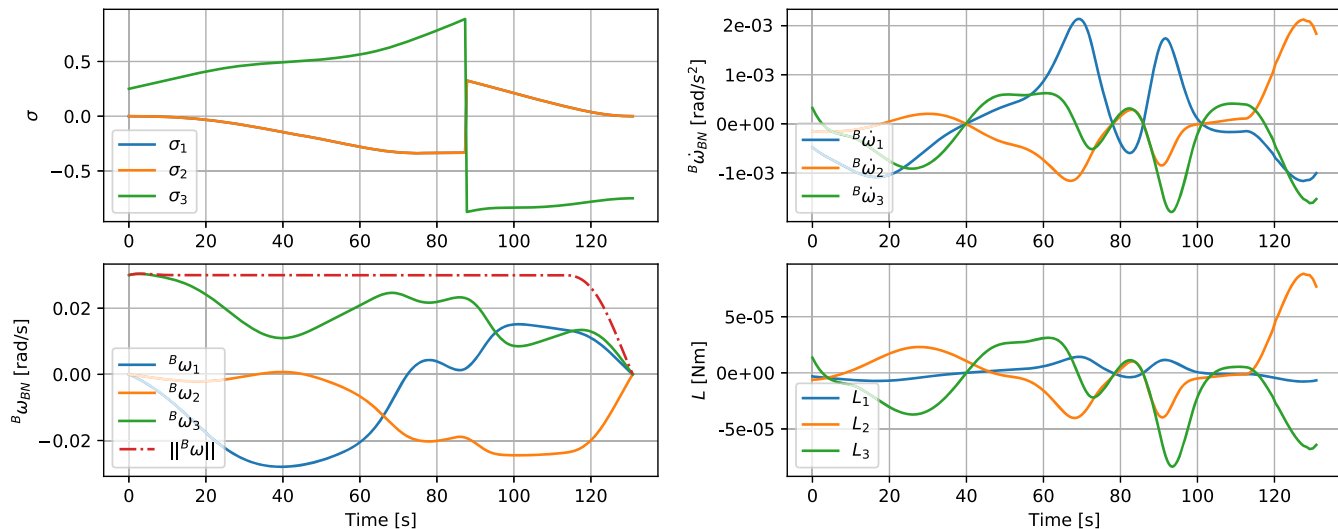**Fig. 16   Interpolated effort-based A* solution for coarse grid with N = 7.**

Fig. 17   Attitude, rates, acceleration, and torque over time for the trajectory in Fig. 16.

in the fact that, with a denser grid, the interpolated trajectory is more heavily constrained in the waypoints that it needs to pass through, causing the spacecraft to have to continuously deflect its trajectory. With a coarser grid, the transition between waypoints is smoother. Interestingly enough, the cost of the path computed with the coarse grid is $\text{cost}(p_{\text{effort}-\text{optimal}}) = 4.21 \cdot 10^{-3}$ N · m · s, less than the fine-grid path, and it is obtained after processing only 61 nodes.

D.   Keep-In and Keep-Out Constraints

Lastly, one example is provided as a proof-of-concept to show the versatility of the algorithm in solving the path-planning query not only for keep-out constraints, but also for keep-in constraints and, even more, combinations of the two. This final example aims at simulating the spacecraft with a sensitive instrument mounted along the $\hat{\boldsymbol{b}}_x$ axis with a field of view of 20 deg, as in the previous scenarios,

a) Path in inertial space

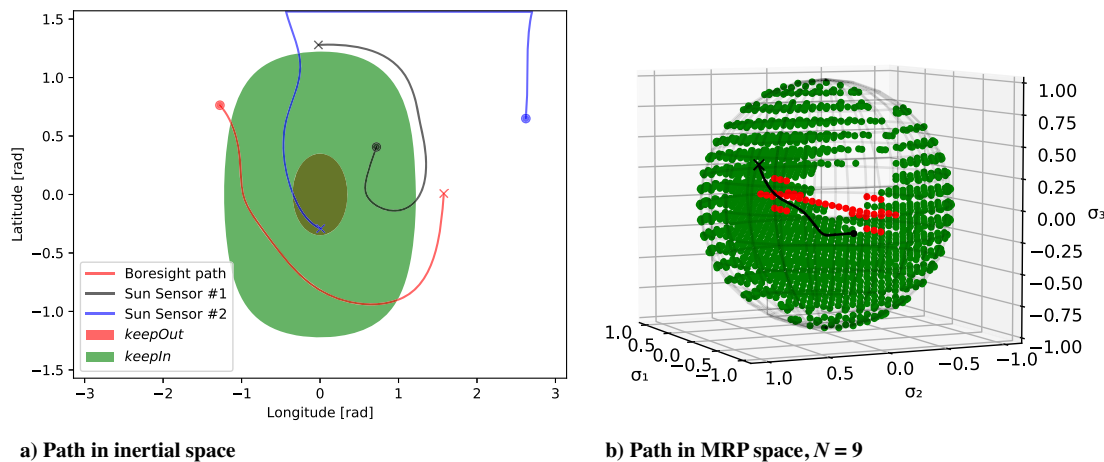**b) Path in MRP space, N = 9**

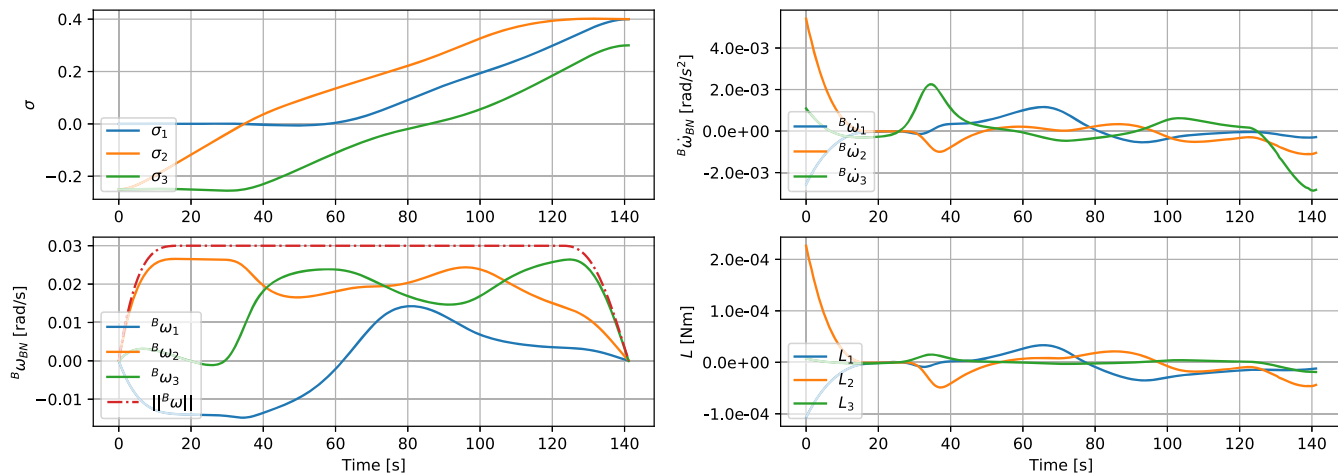**Fig. 18   Interpolated effort-based A* solution with both keep-in and keep-out constraints.**

Fig. 19   Attitude, rates, acceleration, and torque over time for the trajectory in Fig. 18.

together with two sun sensors mounted along the $\hat{\boldsymbol{b}}_y$ and $\hat{\boldsymbol{b}}_z$ axes. Both sun sensors have a field of view of 70 deg, and the keep-in constraint is modeled such that constraint compliance is achieved when at least one of the two sensors can see the sun within its field of view. This scenario, in its simple conceptual formulation, describes a complex problem: assuming that the sun is the only celestial body present, the aim is to maneuver the spacecraft while keeping one body axis ($\hat{\boldsymbol{b}}_x$) away from the direction of the sun, whereas one or both the other body axes must remain fairly close to the same direction of the sun. A rest-to-rest maneuver is simulated, between the attitudes $\boldsymbol{\sigma}_S = [0, -0.25, -0.25]$ and $\boldsymbol{\sigma}_G = [0.4, 0.4, 0.3]$. The inertial direction of the sun is ${}^{\mathcal{N}}\boldsymbol{s} = [1, 0, 0]$. Figure 18 shows the effort-optimal trajectory, which has a $\text{cost}(p_{\text{effort-optimal}}) = 4.47 \cdot 10^{-3}$ N·m·s. In particular, Fig. 18a shows not only the boresight path, but also the paths of the two sun sensors in inertial space: the red area is the keep-out zone for the boresight, whereas the green area represents the keep-in zone for the sensors. Interestingly, the keep-in constraints are satisfied by sensor #1 in the first arc of the trajectory, and by sensor #2 in the second arc, but not always by both simultaneously. Figure 18b shows the attitude path in MRP space, where the trajectory navigates in the green space and avoids the red space. The visualization is made difficult by the large keep-in constraint, which produces two tubular structures in MRP space as there are two sun sensors, making the available space for maneuver larger. The evolution over time of the trajectory is plotted in Fig. 19.

VIII. Conclusions

This paper presents a solution to the constrained attitude maneuvering problem. A spacecraft is reoriented following a constraint-compliant trajectory and maintaining a constant angular speed during the maneuver. Such trajectory is obtained as a sequence of waypoints in a discretized 3D space of possible attitudes by means of B-spline interpolating curves, which turn the set of waypoints into a smooth trajectory that matches the desired endpoint derivatives. The discrete attitude space is searched exploiting a nonsingular MRP-based variation of the A* algorithm, which uses the integral of the control torque over the trajectory as a cost function to determine the optimality of each trajectory.

The main achievement of this paper is that this approach provides not just a sequence of constraint compliant waypoints, but a smooth and differentiable trajectory with smooth derivatives, which allow for smooth torque profiles that can easily be tracked by a reaction wheel-based control subsystem. Moreover, the trajectory provided is the optimal in terms of total effort required by the control subsystem, since it accounts for the spacecraft's mass distribution and its initial and final conditions (angular rates) at the endpoints.

Computational effort can be reduced by lowering the discretization level N of the grid, although this comes at the expenses of a poorer representation of the obstacles in MRP space. Nonetheless, the trajectory computed with a lower N can be more cost-efficient, as it suffers less from the wobbling between waypoints.

In conclusion, the solution provided is cost-efficient with respect to the discretization of the attitude space that is used, i.e., the shape and the fineness of the grid that is used to map the attitude space.

Acknowledgments

References

[1] McInnes, C. R., "Large Angle Slew Maneuvers with Autonomous Sun Vector Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 875–877.
https://doi.org/10.2514/3.21283

[2] Diaz Ramos, M., and Schaub, H., "Kinematic Steering Law for Conically Constrained Torque-Limited Spacecraft Attitude Control," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 9, 2018, pp. 1990–2001.
https://doi.org/10.2514/1.g002873

[3] Lee, U., and Mesbahi, M., "Spacecraft Reorientation in Presence of Attitude Constraints via Logarithmic Barrier Potentials," *Proceedings of the 2011 American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, 2011, pp. 450–455.
https://doi.org/10.1109/acc.2011.5991284

[4] Lee, U., and Mesbahi, M., "Feedback Control for Spacecraft Reorientation Under Attitude Constraints via Convex Potentials," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, 2014, pp. 2578–2592.
https://doi.org/10.1109/taes.2014.120240

[5] Hu, Q., Chi, B., and Akella, M. R., "Anti-Unwinding Attitude Control of Spacecraft with Forbidden Pointing Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 4, 2019, pp. 822–835.
https://doi.org/10.2514/1.g003606

[6] Mabrouk, M., and McInnes, C., "Solving the Potential Field Local Minimum Problem Using Internal Agent States," *Robotics and Autonomous Systems*, Vol. 56, No. 12, 2008, pp. 1050–1060.
https://doi.org/10.1016/j.robot.2008.09.006

[7] Frazzoli, E., Dahleh, M., Feron, E., and Kornfeld, R., *A Randomized Attitude Slew Planning Algorithm for Autonomous Spacecraft*, Vol. 4155, AIAA, Reston, VA, 2001.
https://doi.org/10.2514/6.2001-4155

[8] Kjellberg, H. C., and Lightsey, E. G., "Discretized Constrained Attitude Pathfinding and Control for Satellites," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1301–1309.
https://doi.org/10.2514/1.60189

[9] Kjellberg, H. C., and Lightsey, E. G., "Discretized Quaternion Constrained Attitude Pathfinding," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2015, pp. 713–718.
https://doi.org/10.2514/1.g001063

[10] Tanygin, S., "Fast Autonomous Three-Axis Constrained Attitude Pathfinding and Visualization for Boresight Alignment," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 358–370.
https://doi.org/10.2514/1.g001801

[11] Tan, X., Berkane, S., and Dimarogonas, D. V., "Constrained Attitude Maneuvers on SO(3): Rotation Space Sampling, Planning and Low-Level Control," *Automatica*, Vol. 112, Oct. 2019, Paper 108659.
https://doi.org/10.1016/j.automatica.2019.108659

[12] Schaub, H., and Junkins, J. L., "Stereographic Orientation Parameters for Attitude Dynamics: A Generalization of the Rodrigues Parameters," *Journal of the Astronautical Sciences*, Vol. 44, No. 1, 1996, pp. 1–19.

[13] Hu, Q., Li, L., and Friswell, M. I., "Spacecraft Anti-Unwinding Attitude Control with Actuator Nonlinearities and Velocity Limit," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 10, 2015, pp. 2042–2050.
https://doi.org/10.2514/1.g000980

[14] Schaub, H., and Junkins, J. L., *Analytical Mechanics of Space Systems*, 4th ed., AIAA, Reston, VA, 2018, Chap. 3.
https://doi.org/10.2514/4.105210

[15] Tsiotras, P., Junkins, J. L., and Schaub, H., "Higher-Order Cayley Transforms with Applications to Attitude Representations," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 528–534.
https://doi.org/10.2514/2.4072

[16] Tanygin, S., "Attitude Parameterizations as Higher-Dimensional Map Projections," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 13–24.
https://doi.org/10.2514/1.54085

[17] Tanygin, S., "Projective Geometry of Attitude Parameterizations with Applications to Control," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, 2013, pp. 656–666.
https://doi.org/10.2514/1.60081

[18] Yershova, A., Jain, S., Lavalle, S. M., and Mitchell, J. C., "Generating Uniform Incremental Grids on SO(3) Using the Hopf Fibration," *International Journal of Robotics Research*, Vol. 29, No. 7, 2010, pp. 801–812.
https://doi.org/10.1177/0278364909352700

[19] Lee, C. Y., "An Algorithm for Path Connections and its Applications," *IRE Transactions on Electronic Computers*, Vol. EC-10, No. 3, 1961, pp. 346–365.
https://doi.org/10.1109/tec.1961.5219222

[20] Tarjan, R., "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, Vol. 1, No. 2, 1972, pp. 146–160.
https://doi.org/10.1137/0201010

[21] Hart, P. E., Nilsson, N. J., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100–107.
https://doi.org/10.1109/tssc.1968.300136

[22] Choset, H. M., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S., and Arkin, R. C., *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT Press, Cambridge, MA, 2005, Appendix H.

[23] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2009, Chap. VI.

[24] Ravankar, A., Ravankar, A. A., Kobayashi, Y., Hoshino, Y., and Peng, C.-C., "Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges," *Sensors*, Vol. 18, No. 9, 2018, p. 3170.
https://doi.org/10.3390/s18093170

[25] Piegl, L., and Tiller, W., *The NURBS Book*, Springer Science & Business Media, Berlin, 1996.
https://doi.org/10.1007/978-3-642-59223-2

[26] Fornberg, B., and Zuev, J., "The Runge Phenomenon and Spatially Variable Shape Parameters in RBF Interpolation," *Computers & Mathematics with Applications*, Vol. 54, No. 3, 2007, pp. 379–398.
https://doi.org/10.1016/j.camwa.2007.01.028

[27] De Boor, C., "On Calculating with B-Splines," *Journal of Approximation Theory*, Vol. 6, No. 1, 1972, pp. 50–62.
https://doi.org/10.1016/0021-9045(72)90080-9

[28] Farin, G. E., and Farin, G., *Curves and Surfaces for CAGD: A Practical Guide*, Morgan Kaufmann, San Mateo, CA, 2002.
https://doi.org/10.1016/B978-0-444-51104-1.X5000-X

[29] Schaub, H., and Piggott, S., "Speed-Constrained Three-Axes Attitude Control Using Kinematic Steering," *Acta Astronautica*, Vol. 147, March 2018, pp. 1–8.
https://doi.org/10.1016/j.actaastro.2018.03.022

[30] Calaon, R., Trowbridge, M., and Schaub, H., "A Basilisk-Based Benchmark Analysis of Different Constrained Attitude Dynamics Planners," *AIAA SciTech Forum*, AIAA Paper 2022-2277, 2022.
https://doi.org/10.2514/6.2022-2277

O. Abdelkhalik
Associate Editor