AAS 21-527

CONSTRAINED ATTITUDE MANEUVERING VIA MODIFIED RODRIGUES PARAMETERS - BASED MOTION PLANNING ALGORITHMS

R. Calaon* and H. Schaub†

The attitude dynamics and control of a spacecraft become complex in the presence of constraints in its orientation. Such constraints consist in not pointing sensitive payload towards bright objects in space or, viceversa, keeping the same bright objects within some instrument's field of view. This paper investigates the applicability of motion planning algorithms to navigate a 3D grid in the Modified Rodrigues Parameters (MRPs) configuration space. The aim is to compute a feasible, constraint-compliant reference trajectory in MRP 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. The path is constructed with B-Spline curves that pass through each attitude way points. Simulations show that constraint-compliant, control torque minimizing paths are found if the grid spacing is chosen to be fine enough.

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 such as a telescope or a star tracker towards 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 are 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 repell the S/C from the obstacles.[1–4] These implementations are computationally easy, and provide smooth, differentiable trajectories. However, they can fail in the presence of non-convex obstacles that can lure the trajectory into local minima, causing the spacecraft to remain stuck in the wrong configuration.[5] 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.[6] provide a solution based on Probabilistic RoadMaps (PRM), 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,[7,8] as well as Tanygin,[9] propose a path-planning algorithm that is based on a discretization of the workspace, which eliminates the random component in the planner. Tanygin's approach aims at achieving boresight alignment with a target, regardless

*Ph.D. student, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 3775 Discovery Drive, Boulder, CO, 80303

†Professor, Glenn L. Murphy Chair in Engineering, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309. AAS Fellow, AIAA Fellow.

of the twist component of the attitude around the boresight direction, reducing the problem to a 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 3D orientation of the spacecraft where the final path also aims at minimizing the total angular displacement, while 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.

The problem addressed by this paper is the reorientation of a spacecraft from an initial attitude $\boldsymbol{\sigma}_{\mathrm{S}}$ and angular rate $\boldsymbol{\omega}_{\mathrm{S}}$ to a final attitude and angular rate $\boldsymbol{\sigma}_{\mathrm{G}}$ and $\boldsymbol{\omega}_{\mathrm{G}}$, where "S" and "G" stand for start and goal, 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 Parameters (MRP) space, which represent the discrete space of admissible, constraint-compliant attitudes. The MRP shadow set properties are exploited to yield a globally non-singular attitude path solution. 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.

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.

MRP WORKSPACE DISCRETIZATION

Modified Rodrigues Parameters

The 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 $\boldsymbol{\sigma}$ is obtained from a stereographic projection of the four-dimensional Euler parameter (quaternion) set β onto a 3-dimensional hyperplane.[10] Defining $(\hat{\boldsymbol{e}}, \Phi)$ as the principal rotation vector and angle in a rigid body rotation, the MRP set is obtained as:

$$\boldsymbol{\sigma} = \frac{1}{1+\cos(\Phi/2)} \left\{ \begin{array}{c} e_1 \sin(\Phi/2) \\ e_2 \sin(\Phi/2) \\ e_3 \sin(\Phi/2) \end{array} \right\} = \hat{\boldsymbol{e}} \tan(\Phi/4). \tag{1}$$

The advantage of using MRPs over EPs is that the first are a 3D, constraint-free set, which means that any point in $\mathbf{R}^3$ represents a valid attitude. In contrast, EPs only represent a valid attitude when β lies on the surface of the 4D hypersphere $||\boldsymbol{\beta}|| = 1$. On the other hand, reducing the dimension of the parametrization set from 4 to 3 causes a singularity to appear in the MRP formulation when the set describes a 360 degree 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 $(\boldsymbol{e}, \Phi)$ there exists a shadow set $(\boldsymbol{e}, \Phi')$ 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:

$$\sigma_i^S = \frac{e_i \sin(\Phi'/2)}{1+\cos(\Phi'/2)} = \frac{e_i \sin(\Phi/2 - \pi)}{1+\cos(\Phi/2 - \pi)} = \frac{-\sigma_i}{\sigma^2} \quad \text{for } i = 1,2,3 \tag{2}$$

where $\sigma^2 = ||\boldsymbol{\sigma}||^2$. Therefore, any two MRP sets for which Equation (2) holds true are indeed different MRP sets, which however represent the same attitude. Another interesting property can be inferred from

Equation (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 $\boldsymbol{\sigma}^S = -\boldsymbol{\sigma}$. 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 $\mathbf{R}^3$ contained in a sphere of radius 1, including boundary points.

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

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

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

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

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

with $[\mathbf{I}_{3\times3}]$ 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, Equations (3) and (4) must be differentiated, for which purpose it is useful to define the derivative of $[B(\boldsymbol{\sigma})]$:

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

The differential kinematic equations for the angular accelerations are:

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

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

For a more detailed derivation of MRPs and the related equations, the reader is redirected to Reference 11.

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 semiaxis. A visual representation of the cartesian grid, with a fineness level $N = 6$, is reported in Figure 1: such representation is two dimensional, for ease of visualization, and describes an obstacle-free workspace. As shown in Figure 1, non-equally 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 node and all its neighbors is reported in Figure 2: in purple is the internal node, the nodes 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.

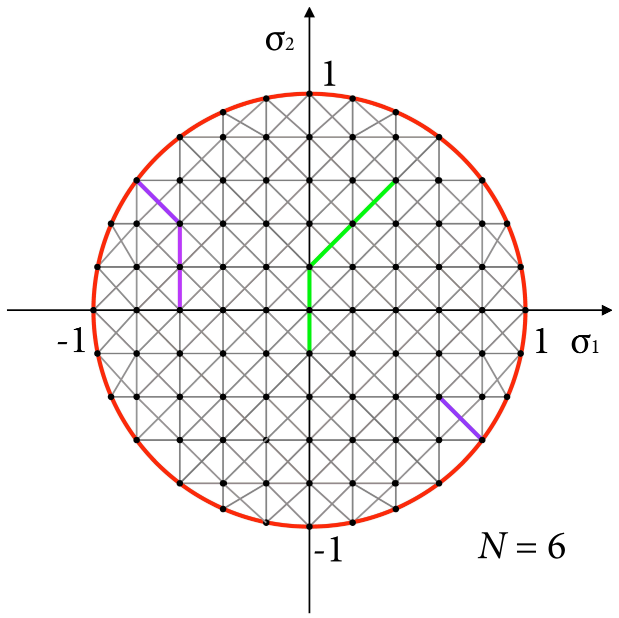


Figure 1: MRP grid with two minimum-distance paths: i) green: no MRP switching; ii) purple: MRP switching

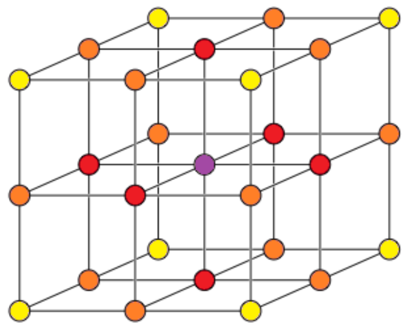


Figure 2: Neighboring nodes

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 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 Equation (1) is fairly linear. The nonlinearity can be further reduced by using higher-order Rodrigues parameters,[12] at the expenses of having to perform multiple set switching to avoid the singularities introduced by more complex formulations. The following work accepts the small deviation from linearity introduced by MRPs. The nonlinearity introduced by Equations (1) and (3) is handled later in this paper, where a constant angular rate magnitude $||\boldsymbol{\omega}||$ is imposed. This allows to track e trajectory for which the rate of variation of attitude over time is constant.

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 ${}^{\mathcal{N}}\hat{\boldsymbol{s}}$, 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:

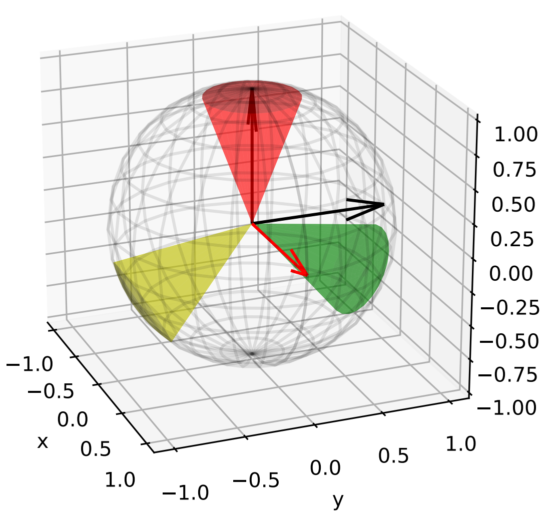
$$ {}^{\mathcal{B}}\hat{\boldsymbol{b}}_i \cdot [\mathcal{BN}]^{\mathcal{N}}\hat{\boldsymbol{s}}_i \lesseqgtr \cos\beta_i \tag{9} $$

where the $<$ sign is used for keep-out constraints, whereas the $\geq$ for keep-in constraints.

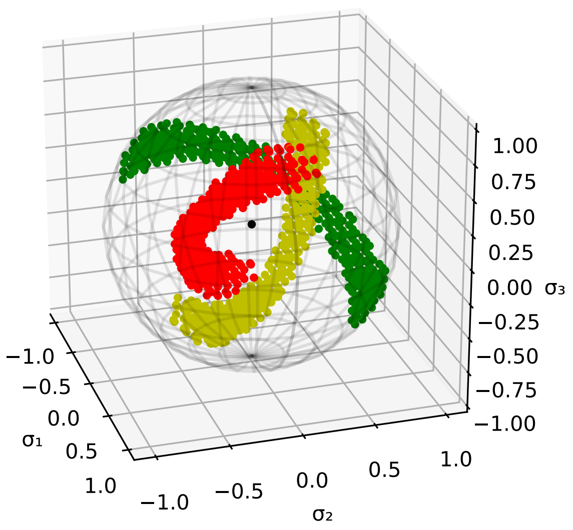
Each of these constraints, in their simplest form, can be visualized as a conical region in the unit sphere. For keep-out constraints, the boresight vector of the sensitive instrument must avoid such conical regions as the spacecraft executes the maneuver. On the contrary, for keep-in constraints, the 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 Equation (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 geometrical 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 3 (a) 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 3 (b) 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

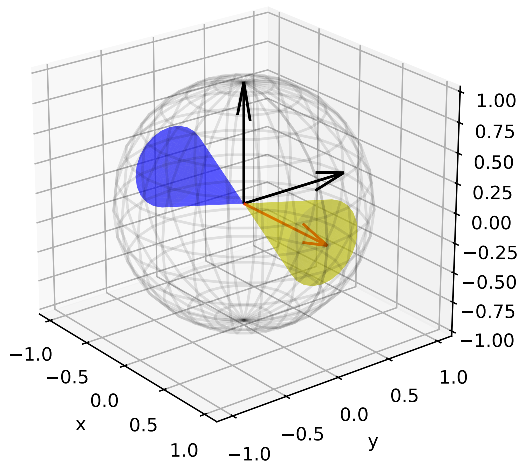


(a) Keep-out cones in inertial space

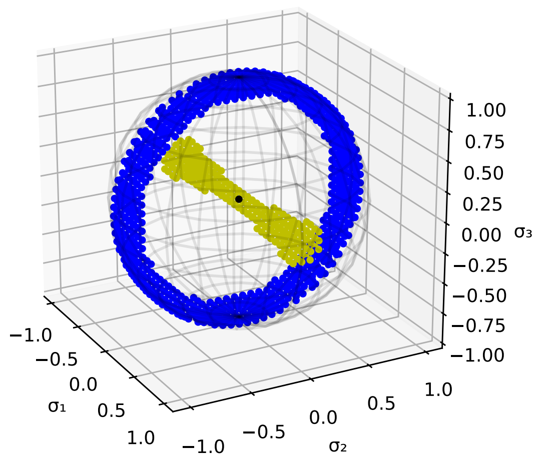


(b) Obstacles in MRP space, $N = 15$

Figure 3: Three general keep-out constraints, $\beta = 20$ deg



(a) Keep-out cones in inertial space



(b) Obstacles in MRP space, $N = 15$

Figure 4: Two constraints, 0 deg and 180 deg apart from boresight, $\beta = 20$ deg

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 Figure 3 (b): 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 Figure 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 Figure 4 (b). 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_y, σ_z) plane.

The results in Figures 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 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 Equation (9). For this reason, the result are analogous to those of Figure 3 (b) and Figure 4 (b), 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.

GRAPH SEARCH: THE A* ALGORITHM

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-optimizing 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.

Algorithm 1: A*

```
Result: Path $n_\text{S}$ to $n_\text{G}$
Pick $n_\text{best}$ such that
  $f(n_\text{best}) \leq f(n)\ \forall n$ in $O$;
while $n_{best} \neq n_G$ do
    Move $n_\text{best}$ from $O$ to $C$;
    for $m$ in $\texttt{Neighbors}(n)$ do
        if $m \notin O$ then
            Add $m$ to $O$;
        else
            if $g(n_{best}) + \texttt{cost}(n_{best}, m) < g(m)$
             then
                Change backpointer of
                  $m$ to point to $n_\text{best}$;
            end
        end
    end
    Pick $n_\text{best}$ such that
      $f(n_\text{best}) \leq f(n)\ \forall n$ in $O$;
end
Backtrack path from $n_\text{G}$ to $n_\text{S}$;
return Path
```

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. Depth-first algorithms, on the contrary, start from the root node and explore nodes at increasingly high link lengths at every iteration until a leaf node is found. 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}. 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}.[13] 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 things 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 more details about A* and other graph search algorithms, the reader is redirected to Reference 13.

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:

$$\hat{d}(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2) = \texttt{min}\left\{d(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2), d(\boldsymbol{\sigma}_1^S, \boldsymbol{\sigma}_2), d(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2^S)\right\}. \tag{10}$$

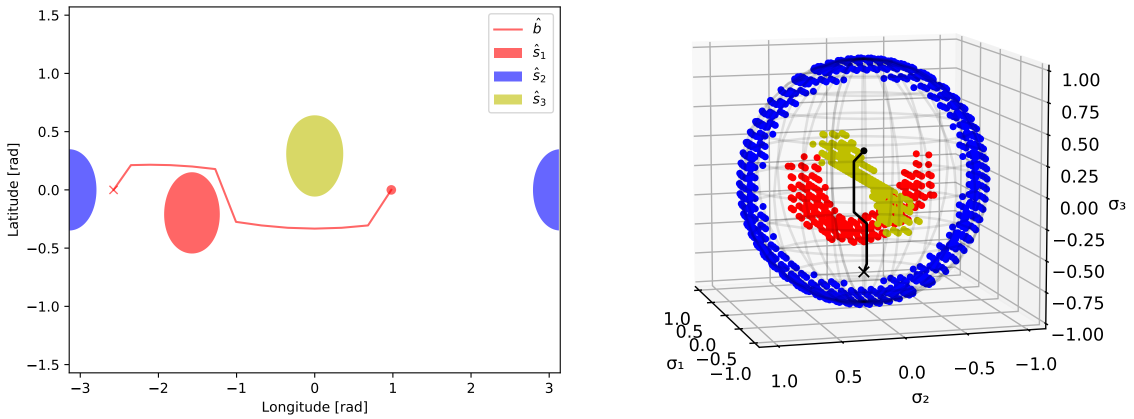
This metric allows to explore the entire discrete domain while avoiding the singularity introduced by MRPs, by automatically performing MRP switching. According to this metric, nodes on the $\sigma = 1$ boundary have a distance $\hat{d} = 0$ with their respective shadow set, therefore allowing to recognize a path that exits the unit sphere and reenters it from the opposite side as, potentially, the minimum-cost path. In Figure 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 $p = [\boldsymbol{\sigma}_S, \boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_n]$ is defined as:

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

and the heuristic:

$$h(\boldsymbol{\sigma}_i) = \hat{d}(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{\mathrm{G}}). \tag{12}$$

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 $\boldsymbol{\sigma}_{\mathrm{S}} = [0, 0, 0.25]$ and $\boldsymbol{\sigma}_{\mathrm{G}} = [0, 0, -0.75]$. 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 $\mathcal{N}$.



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

Figure 5: Metric-based A* solution to the path planning query, 'o' is starting point and 'x' is target point

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.

PATH SMOOTHING: B-SPLINE INTERPOLATION

Global interpolation through MRP waypoints

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.[14,15] B-Spline functions have the form:

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

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:

$$U = \{\underbrace{0,...,0}_{p+1}, u_{p+1}, ..., u_{m-p-1}, \underbrace{1,...,1}_{p+1}\} \tag{14}$$

from which the basis functions are computed as:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \tag{15}$$

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 Equation (13):

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

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

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

from which:

$$\begin{aligned} \bar{u}_0 &= 0 \\ \bar{u}_k &= \bar{u}_{k-1} + \frac{\hat{d}(\boldsymbol{\sigma}_k, \boldsymbol{\sigma}_{k-1})}{S}. \end{aligned} \tag{18}$$

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:

$$\begin{aligned} u_0 &= ... = u_p = 0 & u_{m-p} &= ... = u_m = 1 \\ u_{p+j+1} &= \frac{1}{p}\sum_{i=j}^{j+p-1} \bar{u}_i & &\text{for } j = 0, ..., m-2p-2. \end{aligned} \tag{19}$$

Global interpolation with specified endpoint derivatives

Solving Equation (16) allows to find the $n+1$ control points $\boldsymbol{P}_i$ that generate a p-th 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 al 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:

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

whereas Equations (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:

$$\begin{aligned}
\boldsymbol{\sigma}^{(k)}(u) &= \sum_{i=0}^{n-k} N_{i,p-k}(u)\boldsymbol{P}_i^{(k)} \\
\boldsymbol{P}_i^{(k)} &= \begin{cases} \boldsymbol{P}_i & k=0 \\ \dfrac{p-k+1}{u_{i+p+1}-u_{i+k}}\left(\boldsymbol{P}_{i+1}^{(k-1)} - \boldsymbol{P}_i^{(k-1)}\right) & k>0 \end{cases}
\end{aligned} \tag{21}$$

from which one obtains:

$$\begin{aligned}
-\boldsymbol{P}_0 + \boldsymbol{P}_1 &= \frac{u_{p+1}}{p}\frac{\mathrm{d}\boldsymbol{\sigma}(0)}{\mathrm{d}u} \\
-\boldsymbol{P}_{n+1} + \boldsymbol{P}_{n+2} &= \frac{1-u_{m-p-1}}{p}\frac{\mathrm{d}\boldsymbol{\sigma}(1)}{\mathrm{d}u}.
\end{aligned} \tag{22}$$

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 Equation (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 Reference 14.

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

$$\frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}u} = \frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}u} = \dot{\boldsymbol{\sigma}}\frac{\mathrm{d}t}{\mathrm{d}u}. \tag{23}$$

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

$$\frac{\mathrm{d}t}{\mathrm{d}u} = T \Rightarrow \frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}u} = \dot{\boldsymbol{\sigma}} \cdot T. \tag{24}$$

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

$$T \approx \frac{4S}{\bar{\omega}}. \tag{25}$$

The path computed with the basic A* implementation in the previous subsection is interpolated with a 4-th order B-Spline curve, with endpoint derivatives ${}^\mathcal{B}\boldsymbol{\omega}_\text{S} = {}^\mathcal{B}\boldsymbol{\omega}_\text{G} = [0,0,0]$, and an average angular rate $\bar{\omega} = 0.03$ rad/s. After computing first and second order derivatives $\dot{\boldsymbol{\sigma}}$ and $\ddot{\boldsymbol{\sigma}}$ from the interpolated trajectory, the angular rates and accelerations are obtained from them applying Equations (4) and (8). Knowing the inertia tensor $[\boldsymbol{I}]$ of the spacecraft, it is possible to compute the required control torque $\boldsymbol{L}$ as:

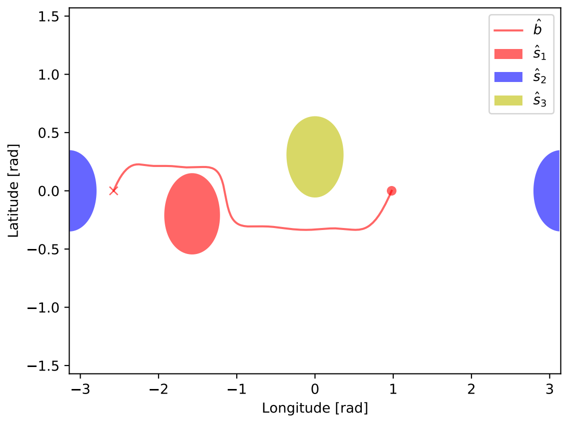
$$\boldsymbol{L} = [\boldsymbol{I}]^\mathcal{B}\dot{\boldsymbol{\omega}}_{\mathcal{B}\mathcal{N}} + {}^\mathcal{B}\boldsymbol{\omega}_{\mathcal{B}\mathcal{N}} \times \left([\boldsymbol{I}]^\mathcal{B}\boldsymbol{\omega}_{\mathcal{B}\mathcal{N}}\right). \tag{26}$$

The interpolated path is shown in Figure 6. Figure 7 shows the evolution of attitude, rates, accelerations and torque, along the maneuver, which is performed in about 150 seconds. 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:

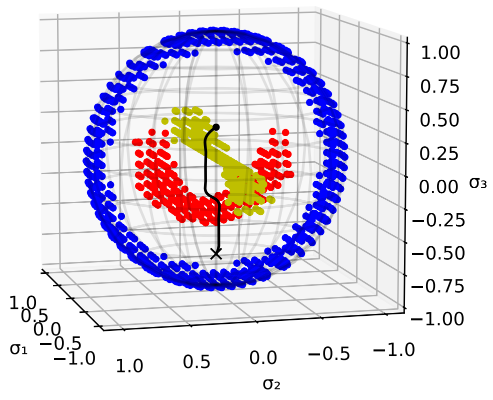
$$^{\mathcal{B}}[\mathbf{I}] = \begin{bmatrix} 6.67 & 0 & 0 \\ 0 & 41.87 & 0 \\ 0 & 0 & 41.87 \end{bmatrix} \cdot \mathbf{10^{-3}}\mathrm{kg} \cdot \mathrm{m}^{\mathbf{2}}. \tag{27}$$

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. Accelerations are also needed when the path requires a deviation from a linear trajectory.

The proposed method 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 wobble 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



(a) Path in inertial space



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

Figure 6: Interpolated metric-based A* solution to the path planning query

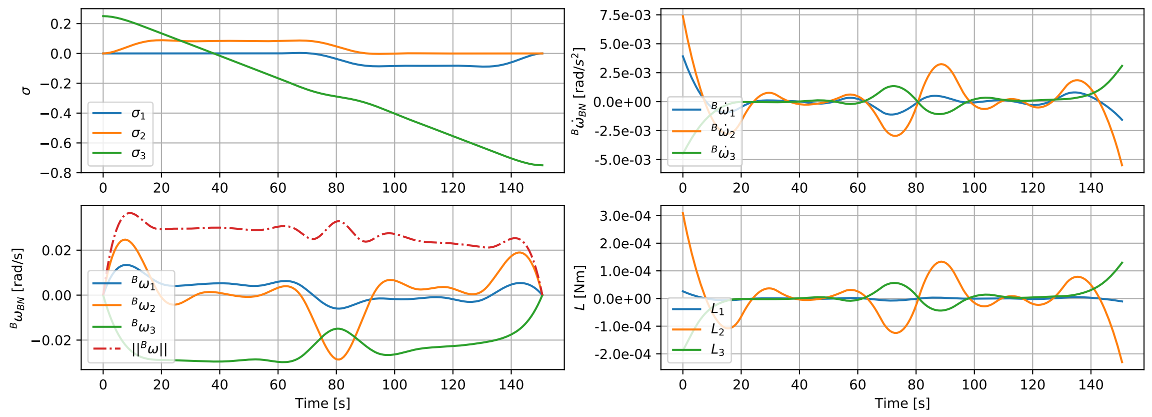


Figure 7: Attitude, rates, acceleration and torque over time for the trajectory in Figure 6

nodes that are located closer to obstacles. Such approach will be the object of future studies, while for the present paper a globally interpolating spline is used.

Lastly, it is challenging to directly interpolate a path that requires MRP switching, because it presents a discontinuity. Since $\sigma \leq 1$ is a soft constraint, part of the waypoint sequence can be mapped to the respective shadow set sequence, allowing to interpolate a curve that has no discontinuity and that exists both inside and outside the unit MRP sphere. After the interpolation, points outside the unit sphere can be mapped again to the respective shadow sets within the unit sphere.

CONSTANT ANGULAR RATE MANEUVERING

The results plotted in Figure 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.[16] 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 remain 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 4th-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:

$$\int_0^{\theta_a} ||\boldsymbol{\omega}(\theta)||\mathrm{d}\theta = \frac{1}{10}\Omega \qquad \int_{\theta_a}^{\theta_b} ||\boldsymbol{\omega}(\theta)||\mathrm{d}\theta = \frac{4}{5}\Omega \qquad \int_{\theta_b}^{\Theta} ||\boldsymbol{\omega}(\theta)||\mathrm{d}\theta = \frac{1}{10}\Omega. \tag{31}$$

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

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

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

The other six coefficients are obtained imposing continuity ($||\boldsymbol{\omega}(\theta_a)|| = ||\boldsymbol{\omega}(\theta_b)|| = \omega^*$), as well as first and second order derivatives equal to 0 for $\theta = \theta_a$ and $\theta = \theta_b$. After some cumbersome algebra, this yields:

$$a_2 = \frac{6(\omega^* - a_0)}{\theta_a^2} - \frac{3a_1}{\theta_a} \qquad a_3 = \frac{8(a_0 - \omega^*)}{\theta_a^3} + \frac{3a_1}{\theta_a^2} \qquad a_4 = \frac{3(\omega^* - a_0)}{\theta_a^4} - \frac{a_1}{\theta_a^3} \tag{34}$$

$$b_2 = \frac{6(\omega^* - b_0)}{\theta_b^2} - \frac{3b_1}{\theta_b} \qquad b_3 = \frac{8(b_0 - \omega^*)}{\theta_b^3} + \frac{3b_1}{\theta_b^2} \qquad b_4 = \frac{3(\omega^* - b_0)}{\theta_b^4} - \frac{b_1}{\theta_b^3}. \tag{35}$$

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

$$\theta_a = \frac{-6\omega^* - 4a_0 + \sqrt{(4a_0 + 6\omega^*)^2 + 2a_1\Omega}}{a_1} \tag{36}$$

$$\theta_b = \theta_a + \frac{4\Omega}{5\omega^*} \tag{37}$$

$$\Theta = \theta_b + \frac{6\omega^* + 4b_0 - \sqrt{(4b_0 + 6\omega^*)^2 - 2b_1\Omega}}{b_1}. \tag{38}$$

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

$$\int_0^\theta ||\boldsymbol{\omega}(\theta)||\mathrm{d}\theta = \int_0^t ||\boldsymbol{\omega}(t)||\mathrm{d}t \tag{39}$$

for every t. The integral on the left-hand side of Equation (39) 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 $0 \leq \theta < \theta_a$ and $\theta_b < \theta \leq \Theta$, to a 5th-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, Figure 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 Figure 8. The magnitude of the angular rate vector ramps up and down from zero at the endpoints, and 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 Figure 7.

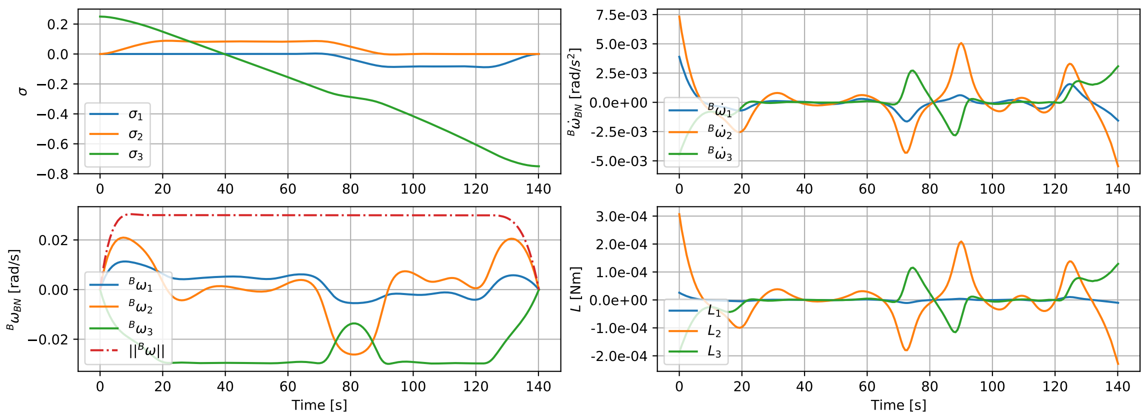


Figure 8: Attitude, rates, acceleration and torque with constant $||\boldsymbol{\omega}||$ for the trajectory in Figure 6

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 point, all the paths were found minimizing the total path length S in terms of cumulative distance between nodes in MRP space (Equation (17)). Within the limits of the nonlinearity introduced by Equation (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, nor for its inertia distribution. Fore 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 ${}^{\mathcal{B}}[\boldsymbol{I}]$. 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:

$$\texttt{cost}(p) = \int_0^T ||\boldsymbol{L}(t)||\mathrm{d}t. \tag{40}$$

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 ($f(n)$) and a heuristic cost to goal from current node ($g(n)$). Moreover, A* is efficient at finding the best path when the heuristic is optimistic, meaning that $h(n)$ should be less or equal than the actual cost of the path from node n to goal.

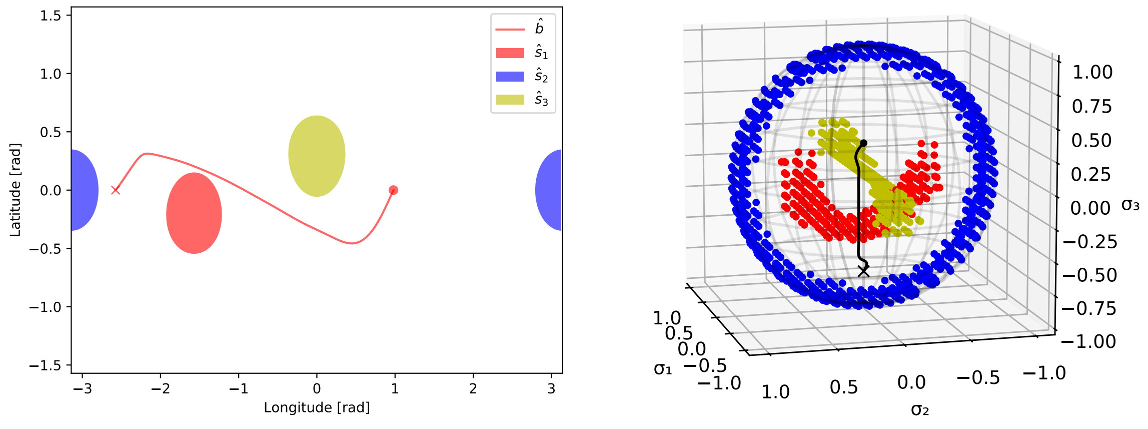
Using the cost function described in Equation (40) makes it challenging to compute $g(n)$ and $h(n)$. One option could be to interpolate two paths, one going from the start node to n: $p_{[\mathrm{S},n]} = [\boldsymbol{\sigma}_\mathrm{S}, ..., \boldsymbol{\sigma}_n]$, and the other one from node n to the goal node: $p_{[n,\mathrm{G}]} = [\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_\mathrm{G}]$. However, the required $\dot{\boldsymbol{\sigma}}_n$ at the intermediate node n is not known, therefore the two interpolated trajectories would have mismatching speeds 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 $\boldsymbol{\sigma}_\mathrm{S}$ to $\boldsymbol{\sigma}_\mathrm{G}$ with a single, smooth trajectory. To overcome this problem, the total cost of a node $p(n) = g(n) + h(n)$ is computed in one single step, applying Equation (40) to a the path

$$p_{[\mathrm{S},n,\mathrm{G}]} = [\boldsymbol{\sigma}_\mathrm{S}, ..., \boldsymbol{\sigma}_n, \boldsymbol{\sigma}_\mathrm{G}]. \tag{41}$$

For such path, the integral between $\boldsymbol{\sigma}_\mathrm{S}$ and $\boldsymbol{\sigma}_n$ corresponds to $g(n)$, whereas the integral between $\boldsymbol{\sigma}_n$ and $\boldsymbol{\sigma}_\mathrm{G}$ corresponds to the heuristic $h(n)$. Such heuristic is optimistic in the sense that it assumes that it is possible to go directly from node n to n_G, regardless of the presence of obstacles along the way. At the same time, Equation (41) 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 $\hat{d}(\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_\mathrm{G})$ 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 Figures 9 and 10. As intuition suggests, the path computed by the regular A* 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 the trajectory, to put the spacecraft onto a coasting arc between $t \approx 50$ sec and $t \approx 110$ sec, where the torques remain close to zero. Finally, after the quasi-coasting arc, the spacecraft is steered towards the target node. The total cost of the effort-optimal path is $\texttt{cost}(p_{\text{effort-optimal}}) = 5.53 \cdot 10^{-3}$ Nms, as opposed to the S-optimal path $\texttt{cost}(p_{S-\text{optimal}}) = 8.72 \cdot 10^{-3}$ Nms.



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

Figure 9: Interpolated effort-based A* solution

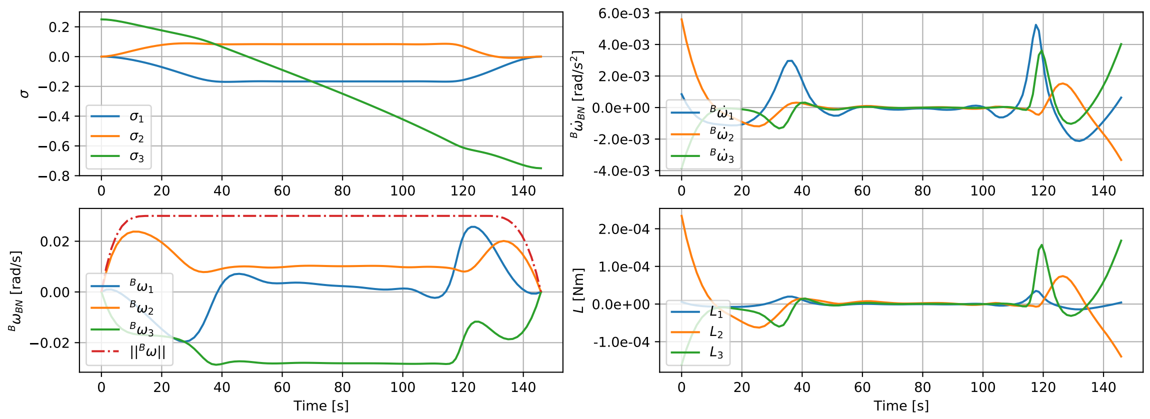
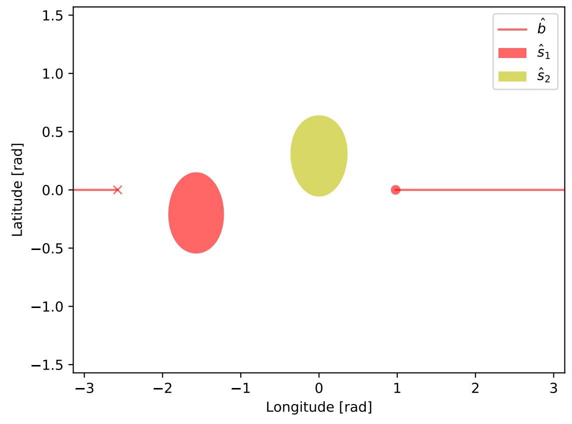


Figure 10: Attitude, rates, acceleration and torque over time for the trajectory in Figure 9

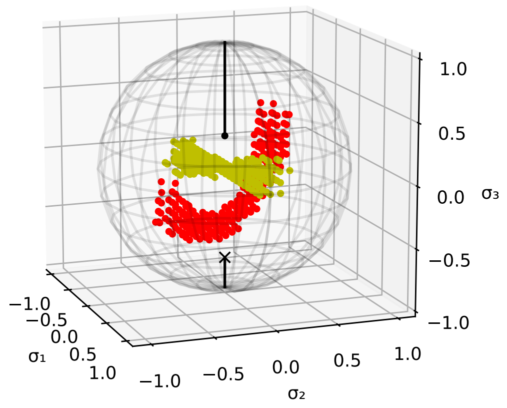
NUMERICAL PERFORMANCE STUDY

Long rotation & 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 can be tracked performing 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 and the relative information are reported in Figures 11 and 12. 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 Figure 11 (b), where the trajectory exits the unit MRP sphere and reenters from the opposite side. The total cost of the path is $\texttt{cost}(p_{\text{effort-optimal}}) = 2.50 \cdot 10^{-3}$ Nms.



(a) Path in inertial space



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

Figure 11: Interpolated effort-based A* solution, eigenaxis rotation and MRP switching scenario

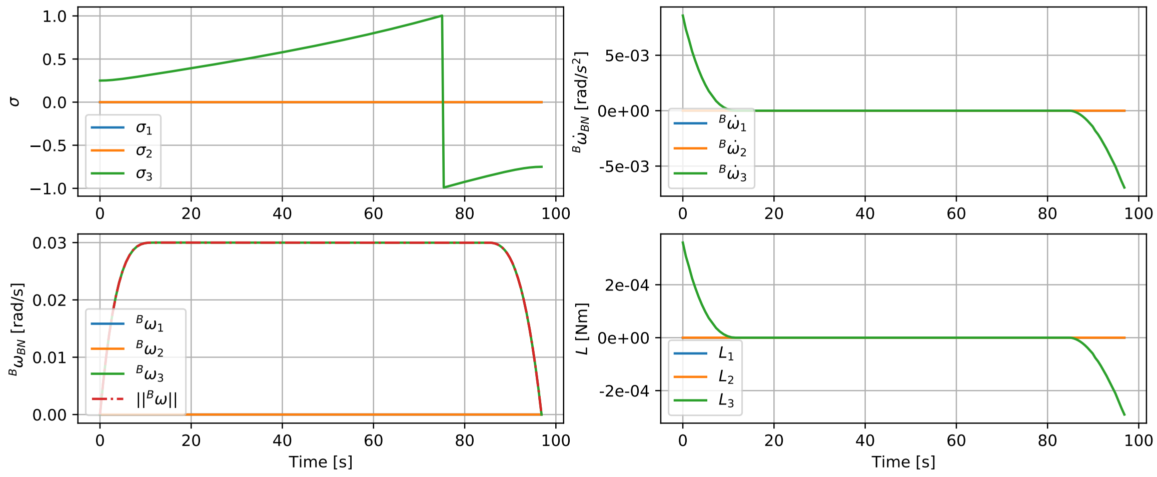
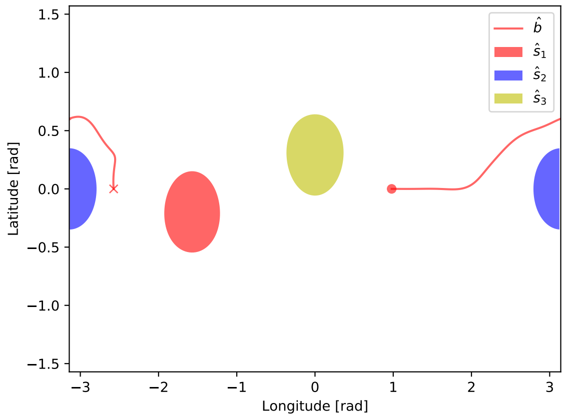


Figure 12: Attitude, rates, acceleration and torque over time for the trajectory in Figure 11

Non-rest-to-rest scenario

The previous sections have shown the development of the effort-based A* algorithm an ultimately applied it to find the most cost-efficient rest-to-rest trajectory given $\boldsymbol{\sigma}_{\mathrm{S}} = [0, 0, 0.25]$ and $\boldsymbol{\sigma}_{\mathrm{G}} = [0, 0, -0.75]$ and a set of three keep-out constraints. One of the advantages of an effort-based approach for pathfinding is that not only it optimizes 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 ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathrm{S}} = [0, 0, 0.03]$ rad/s ($\dot{\boldsymbol{\sigma}}_{\mathrm{S}} = [0, 0, 0.063]$) and a target angular velocity ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathrm{G}} = [0, 0, 0]$ rad/s ($\dot{\boldsymbol{\sigma}}_{\mathrm{G}} = [0, 0, 0]$). The resulting trajectory and the relative attitude, rates, accelerations and torques are reported in Figures 13 and 14. The total cost of the path is $\texttt{cost}(p_{\text{effort-optimal}}) = 5.45 \cdot 10^{-3}$ Nms. 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.


(a) Path in inertial space

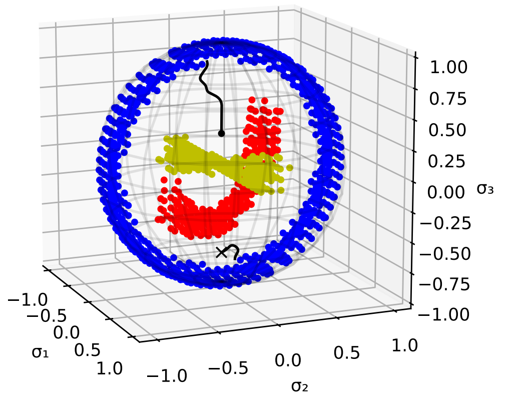

(b) Path in MRP space, $N = 13$

Figure 13: Interpolated effort-based A* solution for non-rest-to-rest case

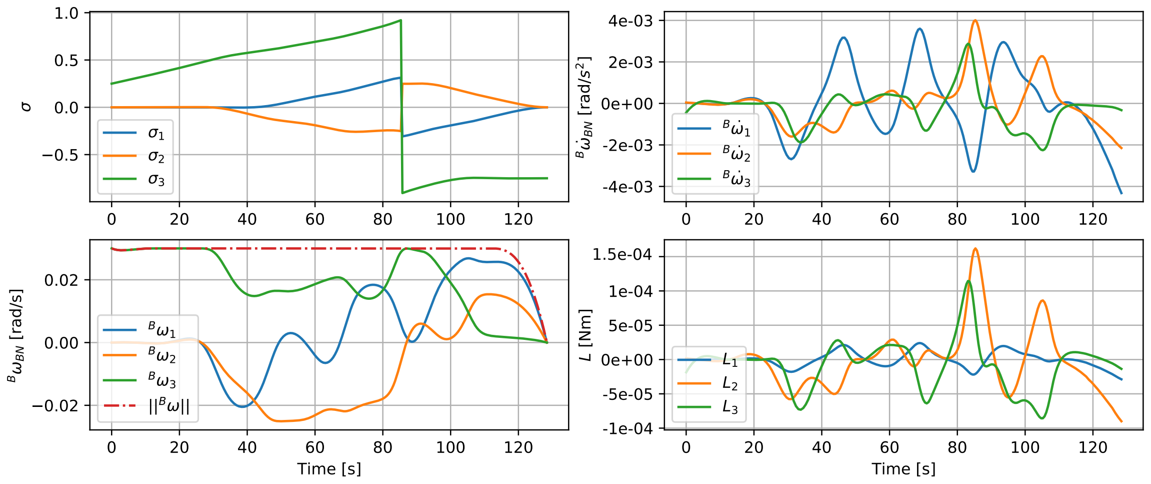
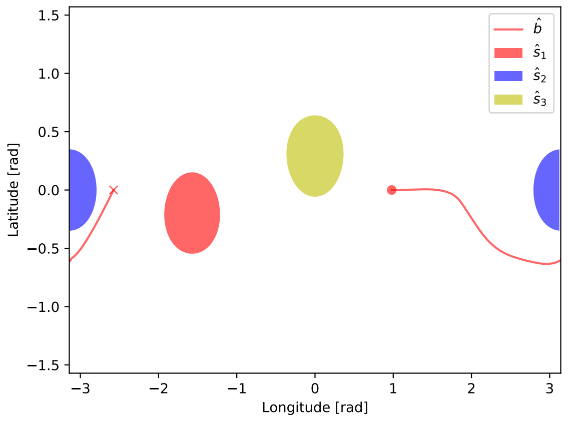


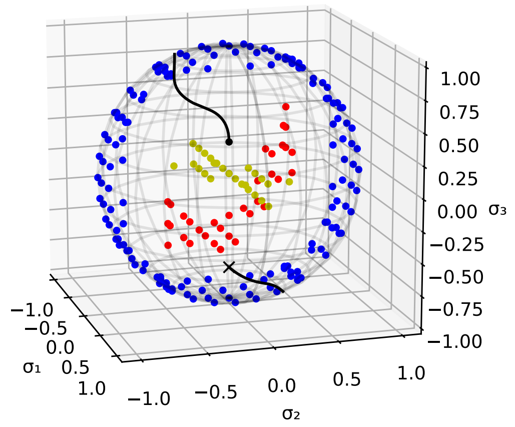
Figure 14: Attitude, rates, acceleration and torque over time for the trajectory in Figure 13

Varying fineness level N

All the previous results have been obtained with a 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 Figure 9 requires processing 632 nodes before finding the optimal trajectory, whereas the non-rest-to-rest case in Figure 13 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 15 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, Figure 16 shows that the angular rates start changing from the very beginning, as compared to Figure 14 where they remained approximately constant for about 20 seconds. Comparing the torque profiles, it looks like the coarser grid also allows for smoother, less oscillating torques. This finds justification 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 $\texttt{cost}(p_{\text{effort-optimal}}) = 4.21 \cdot 10^{-3}$ Nms, less than the fine-grid path, and it is obtained after processing only 61 nodes.



(a) Path in inertial space



(b) Path in MRP space, $N = 7$

Figure 15: Interpolated effort-based A* solution for coarse grid with $N = 7$

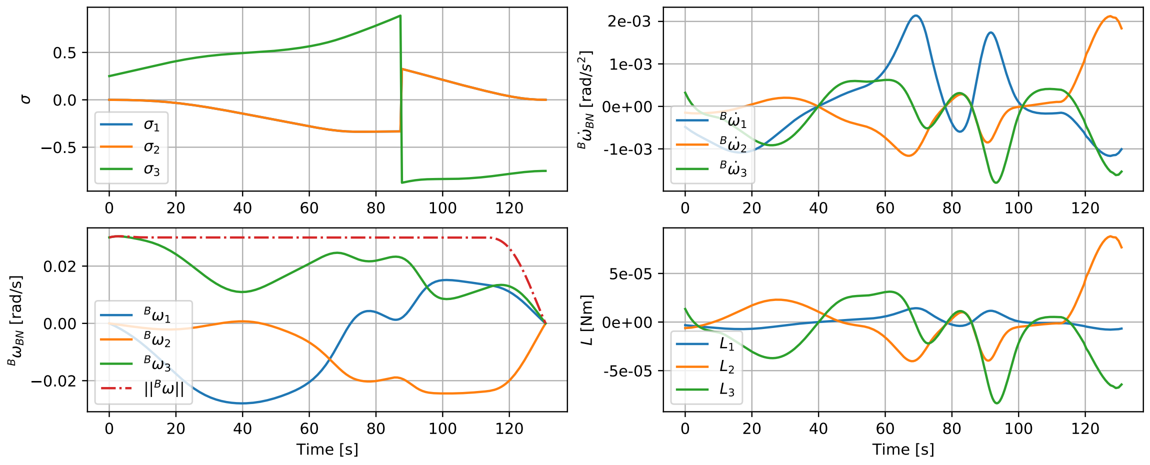
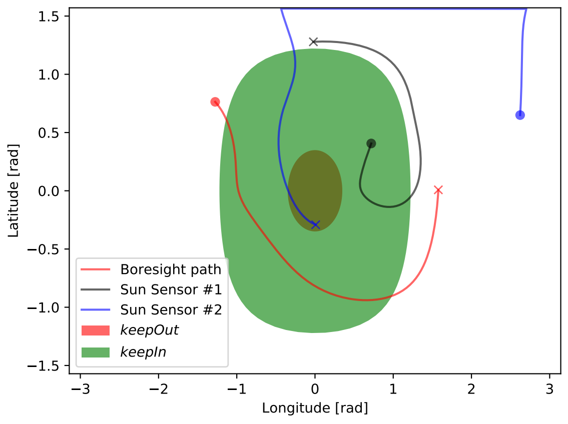


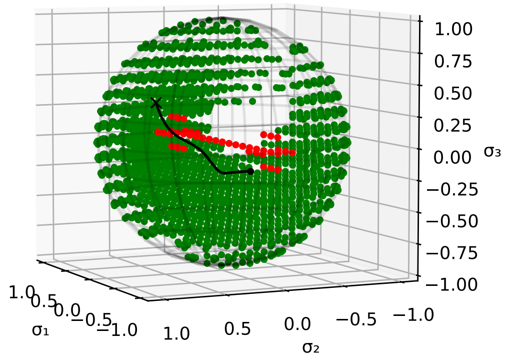
Figure 16: Attitude, rates, acceleration and torque over time for the trajectory in Figure 15

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, 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 modelled 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 $\boldsymbol{\sigma}_\mathrm{S} = [0, -0.25, -0.25]$ and $\boldsymbol{\sigma}_\mathrm{G} = [0.4, 0.4, 0.3]$. The inertial direction of the Sun is ${}^\mathcal{N}\boldsymbol{s} = [1, 0, 0]$. Figure 17 shows the effort-optimal trajectory, which has a $\texttt{cost}(p_\text{effort-optimal}) = 4.47 \cdot 10^{-3}$ Nms. In particular, Figure 17 (a) 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 is 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 17 (b) 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 Figure 18.



(a) Path in inertial space



(b) Path in MRP space, $N = 9$

Figure 17: Interpolated effort-based A* solution with both keep-in and keep-out constraints

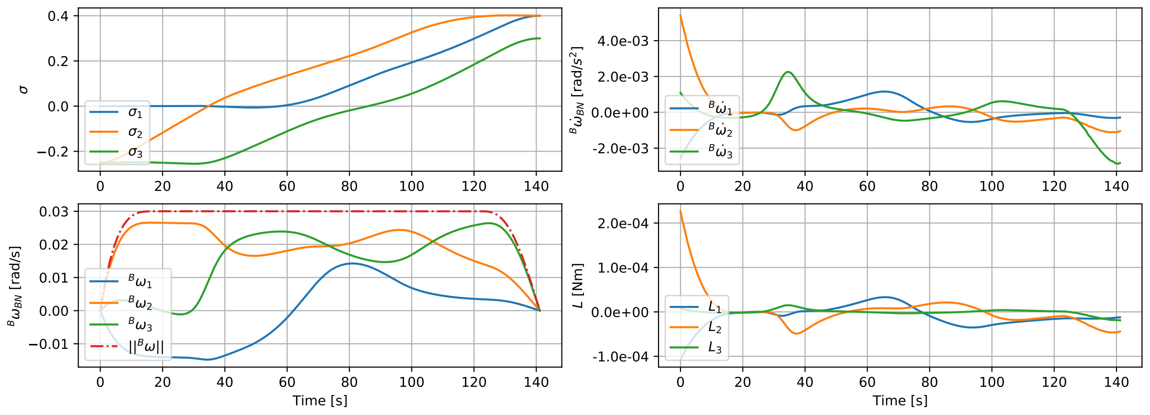


Figure 18: Attitude, rates, acceleration and torque over time for the trajectory in Figure 17

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 non-singular 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. A future development of this work will focus on the role of the discretization level as a tuning parameter for the optimality of the solution. Additionally, different types of grid discretizations will be investigated, with the purpose of sampling the workspace in a way that better describes the attitude space.

REFERENCES

[1] C. R. Mclnnes, "Large angle slew maneuvers with autonomous sun vector avoidance," *Journal of guidance, control, and dynamics*, Vol. 17, No. 4, 1994, pp. 875–877.

[2] M. Diaz Ramos and H. Schaub, "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.

[3] U. Lee and M. Mesbahi, "Spacecraft reorientation in presence of attitude constraints via logarithmic barrier potentials," *Proceedings of the 2011 American Control Conference*, IEEE, 2011, pp. 450–455.

[4] U. Lee and M. Mesbahi, "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.

[5] M. Mabrouk and C. McInnes, "Solving the potential field local minimum problem using internal agent states," *Robotics and Autonomous Systems*, Vol. 56, No. 12, 2008, pp. 1050–1060.

[6] E. Frazzoli, M. Dahleh, E. Feron, and R. Kornfeld, *A randomized attitude slew planning algorithm for autonomous spacecraft*, Vol. 4155. AIAA Reston, VA, 2001.

[7] H. C. Kjellberg and E. G. Lightsey, "Discretized constrained attitude pathfinding and control for satellites," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1301–1309.

[8] H. C. Kjellberg and E. G. Lightsey, "Discretized quaternion constrained attitude pathfinding," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2015, pp. 713–718.

[9] S. Tanygin, "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.

[10] H. Schaub and J. L. Junkins, "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.

[11] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. AIAA, 4 ed., 2018.

[12] P. Tsiotras, J. L. Junkins, and H. Schaub, "Higher-order Cayley transforms with applications to attitude representations," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 528–534.

[13] H. M. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, and R. C. Arkin, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[14] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 1996.

[15] B. Fornberg and J. Zuev, "The Runge phenomenon and spatially variable shape parameters in RBF interpolation," *Computers & Mathematics with Applications*, Vol. 54, No. 3, 2007, pp. 379–398.

[16] H. Schaub and S. Piggott, "Speed-constrained three-axes attitude control using kinematic steering," *Acta Astronautica*, Vol. 147, 2018, pp. 1–8.