Spacecraft Attitude Control using the Invariant-Set Motion-Planner

Claus Danielson[†], Joseph Kloeppel[‡], Christopher Petersen[‡]

***Abstract*— This paper adapts the invariant-set motion-planner for safe spacecraft attitude control. The invariant-set motion-planner is a motion-planning algorithm that uses the positive-invariant sets of the closed-loop dynamics to find a constraint admissible path to a desired target through an obstacle filled environment. We use the invariant-set motion-planner to plan a sequence of reference quaternion waypoints that safely guides the spacecraft attitude around keep-out cones to a desired orientation. Our main contribution is the use of parametric optimization to derive a computationally efficient method solving the non-convex safety certification optimization problem. The computational efficiency of our safety certification is demonstrated in a numerical example.**

***Index Terms*— Optimization, Aerospace, Constrained control, Network analysis and control**

I. Introduction

THE invariant-set motion-planner (ISMP) is an algorithm for generating dynamically feasible trajectories from an initial state to a target equilibrium through an obstacle-filled environment [1]–[7]. Like other motion-planning algorithms [8], [9], the ISMP abstracts the motion-planning problem as a graph search. The defining feature of the ISMP is that knowledge of the closed-loop system dynamics is incorporated into the search graph using constraint admissible positive invariant (PI) sets, which we call safe sets (also called viable sets [10]). These safe sets describe regions of the state-space where the system can safely track the corresponding references. The ISMP uses a graph search to find a corridor of safe sets that safely guides the system through the obstacle filled environment to the target equilibrium.

The ISMP has several beneficial properties. It allows for aggressive, but safe maneuvers since, by definition, the system state will never leave the safe PI sets. It typically has low online computational costs since the PI sets can be pre-computed as they only depend on the time-invariant closed-loop dynamics, rather than the time varying environment. It is inherently robust since it incorporates feedback into the design and the PI sets provide a natural buffer that can absorb tracking errors due to model uncertainty and disturbances [6]. It reduces the curse-of-dimensionality by sampling from the output-space instead of the state-space. Indeed for this paper, we sample the orientation-space (quaternions) rather than the full state-space, which also includes the spacecraft angular velocity. Since it plans motion based on the closed-loop dynamics, the ISMP does not require replacing the existing controller with a customized controller. Indeed, one of the novel contributions of this paper is adapting the ISMP for quaternion-based attitude controllers, which are commonly used in practice.

This paper applies the ISMP to the problem of spacecraft attitude control. The objective of the motion-planning problem is to re-orient the spacecraft while avoiding undesirable orientations, for instance, orientations that would point a sensitive instrument at a bright object like the sun or earth. Although the ISMP was previously applied to attitude control [1], this paper has several crucial differences. This paper considers a quaternion-based spacecraft attitude controller. This is an important contribution since one of the primary advantages of the ISMP is its potential compatibility with existing controllers and quaternion attitude controllers are widely used for attitude control [11]. However, this presents a unique challenge since the quaternions are a double-cover of $\mathbb{SO}(3)$, which introduces non-convexity that complicates the safety checks. Another (minor) contribution of this paper is the use of semi-uniform sampling of the search space. This is important since it reduces the influence of the sampling-method, allowing the dynamics and constraints to dictate the desired attitude trajectory of the spacecraft.

The main contribution of this paper is a closed-form solution for certifying the safety of the PI sets. We provide the necessary and sufficient conditions for guaranteeing that the PI sets are constraint admissible (also known as collision detection). In previous work [1], safety certification was performed by solving an optimization problem to find a safe level-set of a Lyapunov function. This renders real-time safety certification computationally difficult since it must be performed for thousands of candidate safe sets. In contrast, this paper uses parametric optimization to obtain a closed-form solution to the non-convex safety certification optimization problem. We demonstrate that this closed-form solution allows for more than $100{,}000$ safety checks in less than 3 milliseconds. Indeed, using the method presented in this paper, safety certification becomes faster than the graph search. This computationally efficient safety certification can be useful for other attitude control schemes, for instance, the formal-methods based approaches [12]. In summary, our contributions are:

- Computationally efficient and rigorous safety certification.
- Semi-uniform sampling of $\mathbb{SO}(3)$.
- Integration of quaternion control with the ISMP.

This work was supported in part by NSF under the grant CMMI-2105631. [†]cdanielson@unm.edu, Assistant Professor, Department of Mechanical Engineering, University of New Mexico [‡]Air Force Research Laboratory AFRL/VSSVOrgMailbox@us.af.mil, Distro A: Approved for Public Release

Notation and Definitions: A set $\mathcal{O}$ is positive invariant if $x(t_0) \in \mathcal{O} \Rightarrow x(t) \in \mathcal{O}\ \forall t > t_0$. Level-sets $\{x : V(x) \leq l\}$ of Lyapunov functions are positive invariant. $\mathbb{SO}(3) \subset \mathbb{R}^{3\times 3}$ is the group of rotation matrices i.e. $R^\top R = RR^\top = I$ and $\det(R) = +1$. With abuse of terminology, we will use $\mathbb{SO}(3)$ to refer to an abstract group isomorphic to $\mathbb{SO}(3)$. $\mathbb{SE}(3) = \mathbb{SO}(3) \times \mathbb{R}^3$. $\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} : x^\top x = 1\}$. The quaternions $q = (q_0, q_1, q_2, q_3) \in \mathbb{H}$ are a group of hyper-complex numbers with the Hamilton product $q \otimes p = (q_0 p_0 - q_v^\top p_v, q_0 p_v + p_0 q_v + q_v \times p_v)$ where $q_v = (q_1, q_2, q_3)$ is the vector part of q. $\mathbb{1} = (1, 0, 0, 0) \in \mathbb{H}$ is the identity quaternion. $\bar{q} = (q_0, -q_v)$ is the conjugate of $q = (q_0, q_v)$. $\bar{\mathbb{H}} = \mathbb{H} \cap \mathbb{S}^3$ denotes the unit quaternions. The unit quaternions $\bar{\mathbb{H}}$ are a double-cover of $\mathbb{SO}(3)$ since $\pm q \in \bar{\mathbb{H}}$ represent the same element of $\mathbb{SO}(3)$. With abuse of notation, $q \otimes v$ is the quaternion product of $q \in \mathbb{H}$ and $(0, v) \in \mathbb{H}$ where $v \in \mathbb{R}^3$.

A directed graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$ is a set of vertices $\mathbb{I}$ together with a set of ordered pairs $\mathbb{E} \subseteq \mathbb{I} \times \mathbb{I}$ called edges. Vertices $i, j \in \mathbb{I}$ are called adjacent if $(i, j) \in \mathbb{E}$ is an edge. A path is a sequence of adjacent vertices. A graph search is an algorithm for finding a path through a graph. A connected component of a graph is subset of vertices which are connected by paths.

II. Problem Statement

A. Spacecraft Attitude Dynamics

The spacecraft attitude dynamics are modeled by

$$\dot{q}(t) = \tfrac{1}{2} q(t) \otimes \omega(t) \tag{1a}$$

$$J\dot{\omega}(t) = -\omega(t) \times J\omega(t) + \tau(t) \tag{1b}$$

where $J \in \mathbb{R}^{3\times 3}$ is the spacecraft moment-of-inertia matrix, $q \in \bar{\mathbb{H}}$ and $\omega \in \mathbb{R}^3$ are the spacecraft orientation and angular velocity, respectively, and $\tau \in \mathbb{R}^3$ is the torque applied to control the spacecraft attitude. The output-matrix $C = [I_4, 0] \in \mathbb{R}^{4\times 7}$ extracts the orientation $q = Cx \in \bar{\mathbb{H}}$ from the state $x = (q, \omega) \in \mathbb{SE}(3)$.

A standard quaternion attitude controller [11] is used to orient the spacecraft

$$\tau(t) = \omega(t) \times J\omega(t) - K_p e_v(t) - K_d \omega(t) \tag{1c}$$

where $e(t) = \pm q(t) \otimes \bar{r} \in \bar{\mathbb{H}}$ is the error-quaternion between the actual $q(t) \in \bar{\mathbb{H}}$ and desired $r \in \bar{\mathbb{H}}$ orientations of the spacecraft. Importantly, we select the error quaternion $\pm e(t)$ with $e_0 \geq 0$ where $\bar{\mathbb{H}}$ is a double-cover of $\mathbb{SO}(3)$. The proportional and derivative gains of the controller are $K_p \in \mathbb{R}^{3\times 3}$ and $K_d \in \mathbb{R}^{3\times 3}$, respectively. In practice, a torque allocation algorithm will compute actuators commands to supply the desired torque $\tau(t)$ e.g. from thrusters, reaction wheels, or control moment gyroscopes. However, this is outside the scope of this paper.

The asymptotic stability of the equilibrium $(e, \omega) = (0, 0)$ for the closed-loop system (1) is certified by the following Lyapunov function [11]

$$V(e, \omega) = \begin{bmatrix} e - \mathbb{1} \\ \omega \end{bmatrix}^\top \begin{bmatrix} I & 0 \\ 0 & K_p^{-1} J \end{bmatrix} \begin{bmatrix} e - \mathbb{1} \\ \omega \end{bmatrix} \tag{2}$$

where $\mathbb{1} \in \bar{\mathbb{H}}$ is the identity quaternion and $e_0 \geq 0$. The invariant-set motion-planner will use the Lyapunov function (2) to plan safe pointing maneuvers.

B. Pointing Constraints

The spacecraft is subject to state and input constraints. A common type of state constraints are the keep-out constraints, in which the spacecraft orientation $q(t)$ is kept out $q(t) \notin \mathcal{B} \subseteq \bar{\mathbb{H}}$ of the cone

$$\mathcal{B}(d, b, c) = \Big\{ q \in \bar{\mathbb{H}} : d^\top R(q) b \geq c \Big\} \tag{3}$$

where $c = \cos\theta$ is the cone angle, $d \in \mathbb{S}^2$ and $b \in \mathbb{S}^2$ are unit-vectors in the inertial and body frames, respectively, and $R(q)$ is the rotation-matrix between these frames, which depends on the spacecraft orientation $q \in \bar{\mathbb{H}}$. The constraint $q(t) \notin \mathcal{B}$ keeps the angle $\cos^{-1}(d^\top R(q) b)$ between the inertial-frame d and body-frame b unit-vectors sufficiently large $\cos^{-1}(d^\top R(q) b) > \theta$. Keep-out constraints can be used to ensure that a sensitive onboard instrument (e.g. a star-tracker) is not pointed at a bright object (e.g. the sun, earth, or moon) which could temporarily blind or even permanently damage the instrument.

Another common type of state-constraints are keep-in constraints, in which the spacecraft orientation $q(t)$ is kept inside $q(t) \in \mathcal{Q}$ a cone $\mathcal{Q} = \Big\{ q \in \bar{\mathbb{H}} : d_{\mathcal{Q}}^\top R(q) b > \cos\theta_{\mathcal{Q}} \Big\}$. Keep-in constraints can be used to ensure that an onboard instrument (e.g. a telescope) points in a particular direction within a given bound. Keep-in constraints are equivalent to enforcing keep-out constraints $q(t) \notin \bar{\mathbb{H}} \setminus \mathcal{Q}$ on the complement $\mathcal{B} = \bar{\mathbb{H}} \setminus \mathcal{Q}$ of the keep-in set $\mathcal{Q}$. Note that the complement $\bar{\mathbb{H}} \setminus \mathcal{Q}$ has the form (3) where $d = -d_{\mathcal{Q}}$ and $\theta = \pi - \theta_{\mathcal{Q}}$. Thus, in this brief paper we will focus only on enforcing keep-out constraints.

Bounds on the control torque τ and angular velocity ω can be enforced by limiting the size ℓ of the PI sets $\mathcal{O}$, which can be computed offline using the method from e.g. [1]. Thus, we will not consider this solved problem in this paper.

III. Invariant-Set Motion-Planner

The ISMP is described by Algorithm 1. The ISMP searches an appropriately constructed directed graph $\mathbb{G}$ for a sequence $\{\bar{r}_i\}_{i=1}^N$ of intermediate references $\bar{r}_i \in \bar{\mathbb{H}}$ that guide the spacecraft (1) state $x(t) = (q(t), \omega(t)) \in \mathbb{SE}(3)$ from an initial state $x(0) = x_0$ to a target equilibrium orientation $r_\infty \in \bar{\mathbb{H}}$ while enforcing keep-out constraints $q(t) = Cx(t) \notin \mathcal{B}_k$. The defining feature of the ISMP is that knowledge of the closed-loop spacecraft dynamics (1) is incorporated into the graph $\mathbb{G}$ using its PI sets. Associated with each node $i \in \mathbb{I}$ is a safe set $\mathcal{O}_i$, which is constraint admissible $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ and positive invariant $x(0) \in \mathcal{O}_i \Rightarrow x(t) \in \mathcal{O}_i\ \forall t > 0$. The edges $(i, j) \in \mathbb{E}$ of the graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$ indicate that the state (1) will enter the j-th safe-set $\mathcal{O}_j$ while tracking the i-th node without leaving the current safe-set $\mathcal{O}_i$. Thus, the ISMP avoids obstacles by moving the spacecraft state through a sequence of safe-sets $\mathcal{O}_{\sigma_i}$ for $\{\sigma_i\}_{i=0}^N$.

In this section, we apply the ISMP (Algorithm 1) to spacecraft attitude control. We will describe the PI sets $\mathcal{O}$ and the offline construction of the search graph $\mathbb{G}$. Like other path planning algorithms, safety certification (collision detection) $C\mathcal{O}_i \cap \mathcal{B}_k \neq \varnothing$ is typically the most computationally expensive operation. Thus, the main challenge addressed in this paper is efficiently performing the safety checks $C\mathcal{O}_i \cap \mathcal{B}_k = \varnothing$ in line 1 of Algorithm 1 for all pairs $(i, k) \in \mathbb{I} \times \mathbb{K}$.

Algorithm 1 Invariant-Set Motion-Planner

```
1: Remove unsafe $\mathcal{CO}_i \cap \mathcal{B}_k \neq \varnothing$ nodes $i \in \mathbb{I}$ from $\mathbb{G}$
2: Search the graph $\mathbb{G}$ for a path $\{r_{\sigma_0}, \dots, r_{\sigma_N}\}$ from $r_{\sigma_0} = q(0)$ to $r_{\sigma_N} = r_\infty$
3: set $k \leftarrow 0$
4: repeat
5:     if $x(t) \in \mathcal{O}_{\sigma_{k+1}}$ then
6:         $k \leftarrow k+1$
7:     end if
8:     track current target state $r(t) = r_{\sigma_k}$
9: until $r(t) = r_\infty$
```

A. Invariant-Sets

The PI sets $\mathcal{O}_i$ of the closed-loop system (1) are level-sets of the Lyapunov function (2)

$$\mathcal{O}(r, \ell) = \left\{ \begin{bmatrix} q \\ \omega \end{bmatrix} \in \mathbb{SE}(3) : V(q \otimes \bar{r}, \omega) \leq 2 - 2\ell \right\} \tag{4}$$

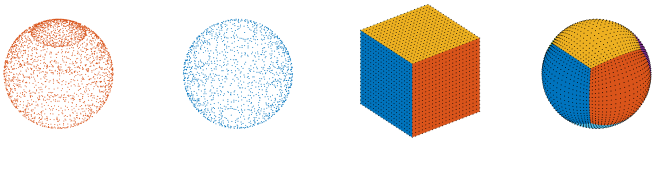
where the level $\ell > 0$ of the Lyapunov function (2) is a tuning parameter that will be used in the graph $\mathbb{G}$ construction, $r \in \bar{\mathbb{H}}$ is the reference orientation for the PI set, and $\bar{r} \in \bar{\mathbb{H}}$ is its conjugate. The parametrization $2 - 2\ell$ of the level-sets will simplify the notation later in the paper.

B. Search Graph Construction

The nodes $i \in \mathbb{I}$ of the graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$ index references $r_i \in \bar{\mathbb{H}}$ and a corresponding PI set $\mathcal{O}_i$ such that $(r_i, 0) \in \mathcal{O}_i$. The grid $\{r_i\}_{i \in \mathbb{I}}$ should be dense and uniform to ensure that graph $\mathbb{G}$ is connected to provide the Algorithm 1 with alternative paths around potential obstacles (3). Furthermore, a dense and uniform grid will ensure that any desired target $r_\infty \in \bar{\mathbb{H}}$ is near an existing reference $r_i \in \bar{\mathbb{H}}$ in the grid $i \in \mathbb{I}$.

Producing a uniform grid on $\mathbb{SO}(3)$ is more challenging than gridding a Euclidean space $\mathbb{R}^n$. Exploiting the axis-angle representation, a uniform grid on $\mathbb{SO}(3)$ can be produced by uniformly gridding axes $n \in \mathbb{S}^2 \subset \mathbb{R}^3$ and angles $\theta \in \mathbb{S}^1 \subset \mathbb{R}^2$ [13]. This approach was used in [1], albeit with non-uniformly gridding of $\mathbb{S}^2$. Motivated by representing functions on the sphere $\mathbb{S}^2$ (e.g. temperature distributions on earth), many methods have been proposed for uniformly gridding $\mathbb{S}^2$ [14]. In particular, a semi-uniform grid on $\mathbb{S}^2$ can be obtained by gridding the facets of a regular polytope in $\mathbb{R}^3$ (i.e. the platonic solids) and then projecting those grid-points onto the unit-sphere $\mathbb{S}^2$ [14] e.g. a geodesic dome is produced by gridding the facets of an icosahedron. Unfortunately, we have found that axis-angle grids are poorly suited for this application. An axis-angle grid will move a body-frame vector b along a collection great circle on $\mathbb{S}^2$ that intersect at b. The resulting grid become sparser farther from b, making it difficult to plan large maneuvers.

Instead, we propose extending the method [14] for gridding $\mathbb{S}^2 \subset \mathbb{R}^3$ to grid $\mathbb{S}^3 \subset \mathbb{R}^4$ where the group $\bar{\mathbb{H}}$ is the set $\mathbb{S}^3$ equipped with the Hamilton product. Among the six regular polytopes in $\mathbb{R}^4$, the unit-tesseract is the easiest to grid since its 8 facets are unit-cubes in $\mathbb{R}^3$. Note that since $\bar{\mathbb{H}}$ is a double-cover of $\mathbb{SO}(3)$, we only need to grid 4 of these 8 facets. In particular, our grid points have the form $r_i = (1, r_{1i}, r_{2i}, r_{3i}) \in \mathbb{H}$ where $r_{ji} \in \mathbb{R}$ for $j = 1, 2, 3$ are sampled from an N point grid of the scalar interval $[-1, 1] \subset \mathbb{R}$. To obtain grids for the other 3 facets, we transpose the position of the 1 with r_{ji} for $j = 1, 2, 3$. To obtain unit-quaternions, we project the grid-points $r_i \in \mathbb{H}$ onto the sphere $\mathbb{S}^3$ i.e. $r_i \leftarrow r_i / \|r_i\| \in \bar{\mathbb{H}}$. This produces a grid with $|\mathbb{I}| = 4N^3$ quaternions. This procedure is illustrated by Fig. 1.



(a) (b) (c) (d)

Fig. 1: (a) Rotations of z under axis-angle gridding of $\bar{\mathbb{H}}$. (b) Rotations of z under tesseract gridding of $\bar{\mathbb{H}}$. (c) Gridding facets of cube in $\mathbb{R}^3$. (d) Projecting grid points onto the sphere $\mathbb{S}^2$. Analogous procedure used to grid $\mathbb{S}^3$.

The edges $(i, j) \in \mathbb{E}$ of the graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$ indicate that the state $x(t)$ of the spacecraft dynamics (1) will enter the safe set $\mathcal{O}_j$ while tracking the i-th reference $r_i \in \bar{\mathbb{H}}$ without leaving the current safe set $\mathcal{O}_i$ (see line 5 of Algorithm 1). This will occur if the equilibrium state $(r_i, 0) \in \mathbb{SE}(3)$ is contained $(r_i, 0) \in \operatorname{int}(\mathcal{O}_j)$ in the interior $\operatorname{int}(\mathcal{O}_j)$ of the j-th PI set $\mathcal{O}_j$. This can be efficiently checked using the expression

$$|r_i^\top r_j| > \ell_i \tag{5}$$

for $i, j \in \mathbb{I}$. This condition follows directly from Proposition 1 in Section V. According to (4), decreasing ℓ_i will produce a larger PI set $\mathcal{O}_i$, increasing the connectivity (5) of the graph, but also increasing the chance of collision $\mathcal{CO}_i \cap \mathcal{B}_j \neq \varnothing$.

C. Safety Certification

Our main contribution is a computationally efficient method for detecting when an invariant-set $\mathcal{O}$ is safe $\mathcal{CO} \cap \mathcal{B} = \varnothing$, i.e., line 1 of Algorithm 1. Our method is described by the following Theorem.

Theorem 1: The PI set $\mathcal{O}$ is safe $\mathcal{CO}(r, \ell) \cap \mathcal{B}(d, b, c) = \varnothing$ if and only if $r_+ = 0$ or $r_- \neq 0$ and

$$\|r_+\| < \ell \text{ and } \left(\ell \|r_+\| + \sqrt{1 - \ell^2} \|r_-\| \right)^2 < \tfrac{1}{2} + \tfrac{1}{2} c \tag{6}$$

where

$$r_+ = \frac{1}{\sqrt{2 + 2d^\top b}} \begin{bmatrix} 0 & 1 + d^\top b \\ d + b & -d \times b \end{bmatrix}^\top r \in \mathbb{R}^2 \tag{7a}$$

$$r_- = \frac{1}{\sqrt{2 - 2d^\top b}} \begin{bmatrix} 0 & 1 - d^\top b \\ d - b & d \times b \end{bmatrix}^\top r \in \mathbb{R}^2. \tag{7b}$$

The proof is provided in Section V. Theorem 1 provides the necessary and sufficient conditions for safety $\mathcal{CO} \cap \mathcal{B} \neq \varnothing$. The safety certification (6) is computationally inexpensive since it only requires checking the norms of low-dimensional vectors $r_+, r_- \in \mathbb{R}^2$. Furthermore, for each keep-out cone $\mathcal{B}(d, b, c)$, the projections (7) can be vectorized so that the safety condition (6) can be simultaneously checked for all the PI sets $\mathcal{O}(r_i, \ell_i)$ for $i \in \mathbb{I}$.

IV. Numerical Example

In this section, we demonstrate Algorithm 1 for spacecraft attitude control, with an emphasis on the computational efficiency of the safety certification described by Theorem 1.

For this case study, the spacecraft will perform a 180° rotation about its z-axis. It starts from the equilibrium state $x(0) = (q(0), \omega(0))$ where it is initially stationary $\omega(0) = 0$ with $q(0) = \mathbb{1} \in \bar{\mathbb{H}}$. The target equilibrium state is $x_\infty = (r_\infty, 0)$ where $r_\infty = (0, 0, 0, 1) \in \bar{\mathbb{H}}$. During the maneuver, the x-axis of the spacecraft $b_{1,2} = (1, 0, 0)$ must be kept out of two cones (3) with $d_1 = (0, 1, 0)$, $\theta_1 = 30°$ and $d_2 = (0, -1, 0)$, $\theta_2 = 5°$. In addition, the z-axis $b_3 = (0, 0, 1)$ must be kept in a 30° cone. This is equivalent to a keep-out cone (3) with $d_3 = (0, 0, -1)$ and $\theta_3 = 180° - 45° = 135°$. The cones are shown in Fig. 2. The first cone is large, forcing the spacecraft to rotate counter-clockwise about its z-axis. To avoid the second cone, the spacecraft will need to rotate the x-axis out of the xy-plane. The deviation from the xy-plane is limited by the third cone, which limits the z-axis movement.

The graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$ was constructed using the procedure described in Section III-B with a grid density $N = 21$ and uniform level $\ell = \cos 6°$. Since the grid is only *semi*-uniform, some of the nodes maybe disconnected for this choice of ℓ. After removing disconnected nodes using a connected components algorithm, the resulting graph $\mathbb{G}$ had $|\mathbb{I}| = 34{,}647$ nodes and $|\mathbb{E}| = 843{,}137$ edges, requiring 15.15 MB of memory to store. It required 9060 milliseconds to fully construct using MATLAB on a 2020 MacBook Pro with a 2.6 GHz i7 processor and 16 GB of RAM. Note that since the search graph only depends on the spacecraft dynamics, it can be constructed offline and stored in memory for online use in Algorithm 1. Table I compares the computation resources required by Algorithm 1, which motivates the decision to construct the graph $\mathbb{G}$ offline and store it in memory for online use.

TABLE I: Computation-time and memory for Algorithm 1.

Computational resources required by Algorithm 1	Time / Memory
Construction of the search graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$	9060 ms
Storage of the search graph $\mathbb{G} = (\mathbb{I}, \mathbb{E})$ and r, ℓ	15.15 MB
Removal of unsafe nodes $i \in \mathbb{I}$ from $\mathbb{G}$	
$\mathcal{O}_i \cap \mathcal{B}_1 = \varnothing \; \forall i \in \mathbb{I}$	0.99 ms
$\mathcal{O}_i \cap \mathcal{B}_2 = \varnothing \; \forall i \in \mathbb{I}$	1.12 ms
$\mathcal{O}_i \cap \mathcal{B}_3 = \varnothing \; \forall i \in \mathbb{I}$	0.71 ms
Search graph $\mathbb{G}$ for path $\{r_{\sigma_1}, \ldots, r_{\sigma_N}\}$	26.67 ms

The first step of Algorithm 1 is the removal of unsafe $C\mathcal{O}_i \cap \mathcal{B}_k \neq \varnothing$ nodes $i \in \mathbb{I}$. This was accomplished using the method described in Theorem 1. The time required to perform the $|\mathbb{I}| = 34{,}647$ safety checks $C\mathcal{O}_i \cap \mathcal{B}_k \neq \varnothing$ for the $|\mathbb{K}| = 3$ obstacles are reported in Table I. Using the proposed method, the total computation-time 2.82 milliseconds for performing the $|\mathbb{I}||\mathbb{K}| = 103{,}941$ safety checks is nearly 10× faster than the graph search (typically a fast operation) and more than 3000× faster than the graph construction. Note that our code takes advantage of the vectorization provided by MATLAB for efficient computation. The remaining 1600 safe nodes $i \in \mathbb{I}$ are shown in Fig. 2, which also shows the three keep-out cones (3). The buffer around the keep-out cones is due to the angle $\cos^{-1} \ell = 6°$ of the PI sets (4).

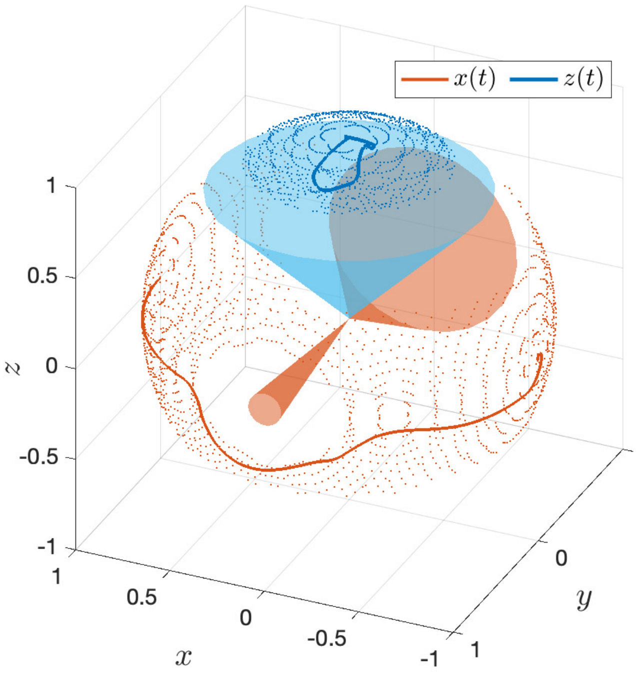


Fig. 2: The trajectories of the spacecraft x-axis around the keep-out cones and z-axis within the keep-in cone.

Searching the graph $\mathbb{G}$ produced a sequence $\{r_{\sigma_k}\}_{k=1}^{28} \subset \mathbb{I}$ of 28 references $r_{\sigma_k} \in \bar{\mathbb{H}}$ that safely guides the spacecraft from the initial state $x_0 = (q_0, 0) \in \mathbb{SE}(3)$ to the target equilibrium state $x_\infty = (r_\infty, 0) \in \mathbb{SE}(3)$. Following Algorithm 1, the reference r_{σ_k} tracked by the controller (1c) is updated $k \leftarrow k + 1$ each time the spacecraft (1) state $x(t) = (q(t), \omega(t) \in \mathbb{SE}(3)$ enters the next PI set $\mathcal{O}_{\sigma_{k+1}} \subseteq \mathbb{SE}(3)$. The switching condition $x(t) \in \mathcal{O}_{\sigma_{k+1}}$ was checked every 1 second. Between updates, the spacecraft dynamics (1) were simulated using MATLAB's `ode45` solver. The resulting state $x(t) = (q(t), \omega(t))$ and input $\tau(t)$ trajectories are shown in Fig. 3. Note that the spacecraft converges to the desired orientation $r_\infty = \pm(0, 0, 0, 1)$.

V. Proof of Theorem 1

To prove Theorem 1, we first transform the sets (3) and (4) into a form more amenable to checking safety $C\mathcal{O} \cap \mathcal{B} \neq \varnothing$.

Proposition 1: The obstacle set (3) and projected PI set (4) can be written as

$$\mathcal{B} = \{q \in \bar{\mathbb{H}} : q^\top P q \geq c\} \tag{8}$$

$$C\mathcal{O} = \{q \in \bar{\mathbb{H}} : q^\top \bar{r}\bar{r}^\top q \geq \ell^2\} \tag{9}$$

where

$$P = \begin{bmatrix} d^\top b & -(d \times b)^\top \\ -d \times b & db^\top + bd^\top - d^\top b I \end{bmatrix} \in \mathbb{R}^{4 \times 4} \tag{10}$$

Proof: Substituting the quaternion parametrization of rotation matrices $R(q)$ into the set description (3), yields

$$d^\top R(q)b = d^\top\big(q_0^2 I - 2q_0 q_v^\times + q_v q_v^\top + (q_v^\times)^2\big)b$$

where $q_v^\times \in \mathbb{R}^{3\times 3}$ is the pseudo-cross-product matrix of q_v. Through liberal use of the property $x \times y = -y \times x$, we can transform (3) into (8) where (10) depends on the unit-vectors d and b.

Next, we derive (9). First, we project $\mathcal{O} \subseteq \mathbb{SE}(3)$ onto the quaternion subspace $C\mathcal{O} \subseteq \bar{\mathbb{H}}$. Since the matrix in the Lyapunov function (2) is block-diagonal, this is trivial

$$C\mathcal{O} = \{q \in \bar{\mathbb{H}} : (\pm q \otimes \bar r - \mathbb{1})^\top(\pm q \otimes \bar r - \mathbb{1}) \le 2 - 2\ell\}.$$

Recall that we selected the error quaternion $e = \pm q \otimes \bar r$ from the double-cover $\bar{\mathbb{H}}$ of $\mathbb{SO}(3)$ such that $e_0 \ge 0$ i.e. $e_0 = |q^\top \bar r|$. Expanding the quadratic, we obtain

$$\begin{aligned}(\pm q \otimes \bar r - \mathbb{1})^\top(\pm q \otimes \bar r - \mathbb{1}) &= (e_0 - 1)^2 + e_1^2 + e_2^2 + e_3^2 \\ &= 2 - 2e_0 = 2 - 2|q^\top \bar r|\end{aligned}$$

where $e_1^2 + e_2^2 + e_3^2 = 1 - e_0^2$ since $e \in \bar{\mathbb{H}}$. Thus, $q \in C\mathcal{O}$ if and only if $2 - 2|q^\top \bar r| \le 2 - 2\ell$ or equivalently $|q^\top \bar r| \ge \ell$. Since, both $|q^\top \bar r|$ and $\ell > 0$ are positive, this is equivalent to $|q^\top \bar r|^2 = q^\top \bar r \bar r^\top q \ge \ell^2$. ■

Proposition 1 says that both $C\mathcal{O}$ and $\mathcal{B}$ are the sets of quaternions *outside* a (degenerate) ellipsoid. From (9), it is apparent that $(r_j, 0) \in \mathcal{O}_i$ if and only (5) holds.

Consider the following optimization problem which finds the quaternion $q \in C\mathcal{O}$ in the PI set closest to the obstacle $\mathcal{B}$

$$J^\star(d,b,r,\ell) = \ \max_{q\in\bar{\mathbb{H}}} \ q^\top P q \tag{11a}$$

$$\text{s.t.} \ \ q^\top \bar r \bar r^\top q \ge \ell^2. \tag{11b}$$

The following lemma relates this optimization problem (11) with the safety certification problem $C\mathcal{O} \cap \mathcal{B} = \varnothing$.

Lemma 1: $C\mathcal{O} \cap \mathcal{B} = \varnothing$ if and only if $J^\star(d,b,r,\ell) < c$.

Proof: We prove the contra-positive of each implication. If $C\mathcal{O} \cap \mathcal{B} \ne \varnothing$ then there exists $q \in C\mathcal{O} \cap \mathcal{B}$ that satisfies (11b) and $q^\top P q \ge c$ by Proposition 1. Thus, $J^\star \ge c$.

Conversely, if $J^\star \ge c$ then there exists $q^\star$ such that $q^{\star\top} P q^\star \ge c$ i.e. $q^\star \in \mathcal{B}$ by Proposition 1. Furthermore, since $q^\star$ satisfies (11b), $q^\star \in C\mathcal{O}$. Thus, $q^\star \in C\mathcal{O} \cap \mathcal{B} \ne \varnothing$. ■

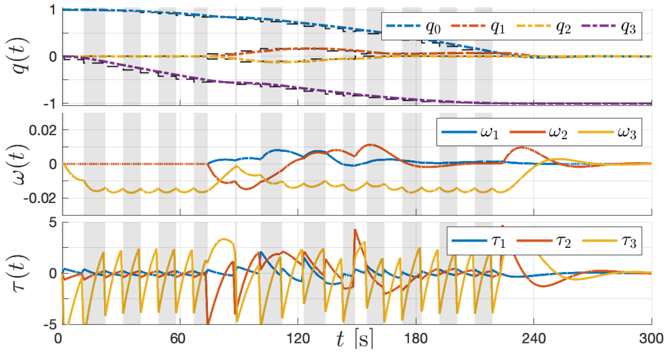


Fig. 3: State $(q(t), \omega(t)) \in \mathbb{SE}(3)$ and input $\tau(t) \in \mathbb{R}^3$ trajectories for the pointing maneuver. The background shading indicates when the waypoint r_i is updated.

The proof of Theorem 1 is based on solving (11) parametrically i.e. we will solve (11) when (11b) is both active and inactive. Unfortunately, (11) is non-convex due to the non-convex constraint (11b), which requires that $q \in \bar{\mathbb{H}}$ is outside the degenerate ellipsoid (9). Less obvious, the cost (11a) is also non-convex (and non-concave), as shown by the following proposition analyzing the eigenstructure of P.

Proposition 2: The matrix $P \in \mathbb{R}^{4\times 4}$ defined in (10) has eigenvalue $\lambda_+ = +1$ with multiplicity 2 and eigenvectors

$$P_+ = \frac{1}{\sqrt{2 + 2d^\top b}}\begin{bmatrix} 0 & 1 + d^\top b \\ d + b & -d \times b\end{bmatrix} \in \mathbb{R}^{4\times 2} \tag{12a}$$

and eigenvalue $\lambda_- = -1$ with multiplicity 2 and eigenvectors

$$P_- = \frac{1}{\sqrt{2 - 2d^\top b}}\begin{bmatrix} 0 & 1 - d^\top b \\ d - b & d \times b\end{bmatrix} \in \mathbb{R}^{4\times 2}. \tag{12b}$$

Proof: The proposition can be verified by direction computation, noting that $d \times b$ is orthogonal to both d and b, and $(d^\top b)^2 + \|d \times b\|^2 = \cos(\psi)^2 + \sin(\psi)^2 = 1$ where ψ is the angle between the unit-vectors d and b. ■

According to Proposition 2, the matrix P is indefinite and thus (11) is neither convex nor concave. Despite the non-convexity, we will exploit the eigenstructure described in Proposition 2 to solve (11) parametrically. First, we consider the case when the sole constraint (11b) is inactive.

Lemma 2: If $\|r_+\| \ge \ell$ then $\mathcal{O}$ is unsafe $C\mathcal{O} \cap \mathcal{B} \ne \varnothing$.

Proof: If the constraint (11b) is inactive then (11) simplifies $J^\star(d,b,r,\ell) = \max_{q\in\bar{\mathbb{H}}} \ q^\top P q$. This non-convex optimization problem has a closed-form solution, specifically $J^\star = \lambda_+ = 1$ is the largest eigenvalue of P. Since $c = \cos(\theta) \le 1 = J^\star$, $C\mathcal{O} \cap \mathcal{B} \ne \varnothing$ by Lemma 1.

This solution is valid when $q^\star$ satisfy (11b) where the non-unique optimal solutions $q^\star$ lie in the eigenspace of λ_+. By Proposition 2 we have $q^\star = P_+ \alpha_+ \in \bar{\mathbb{H}}$ for some $\alpha_+ \in \mathbb{S}^1$. Thus, (11b) is equivalent to

$$(r^\top q^\star)^2 = \alpha_+^\top P_+^\top r r^\top P_+ \alpha_+ = (r_+ \alpha_+)^2 \ge \ell^2.$$

where $r_+ = P_+^\top r$ matches the definition (7). This constraint holds for some $\alpha_+ \in \mathbb{S}^1$ if and only if it holds for the best-case $\alpha_+ = r_+ / \|r_+\| \in \mathbb{S}^1$. Substituting, we obtain the condition $\|r_+\|^2 \ge \ell^2$. ■

The contra-positive of Lemma 2 is the first portion of the safety condition (6) in Theorem 1. Next, we consider the case when the sole constraint (11b) is active.

Lemma 3: Let $\|r_+\| < \ell$. Then, $C\mathcal{O} \cap \mathcal{B} = \varnothing$ if and only if $r_+ = 0$ or $r_- \ne 0$ and $\big(\ell\|r_+\| + \sqrt{1 - \ell^2}\|r_-\|\big)^2 < \frac{1}{2} + \frac{1}{2}c$.

Proof: Define $q_+ = P_+^\top q$ and $q_- = P_-^\top q$ as the projection of $q \in \bar{\mathbb{H}}$ onto the eigenspaces (12) of P. Then, (11) can be written as

$$J^\star = \max \ \ q_+^\top q_+ - q_-^\top q_- \tag{13a}$$

$$\text{s.t.} \ \ (r_+^\top q_+ + r_-^\top q_-)^2 = \ell^2 \tag{13b}$$

$$q_+^\top q_+ + q_-^\top q_- = 1 \tag{13c}$$

where $r_+ = P_+^\top r$ and $r_- = P_-^\top r$ match the definition (7) and $q = P_+ q_+ + P_- q_-$. The new constraint (13c) represents the fact that $q^\top q = 1$ since $q \in \bar{\mathbb{H}}$ where the eigenvectors (12) are orthogonal. The constraint (13b) is active according to Lemma 2 since $\|r_+\| < \ell$. Problem (13) can be rewritten as

$$\begin{aligned} J^\star = \max\ \ & 2q_+^\top q_+ - 1 && \text{(14a)}\\ \text{s.t.}\ \ & (r_+^\top q_+ + r_-^\top q_-)^2 = \ell^2 && \text{(14b)}\\ & q_-^\top q_- = 1 - q_+^\top q_+. && \text{(14c)}\end{aligned}$$

To invoke Lemma 1, we only need the cost (14a) which only depends on $q_+ \in \mathbb{R}^2$ in this formulation. For the degenerate case when $r_- = 0$, we immediately have $q_-^\star = 0$ and $q_+^{\star\top} q_+^\star = 1$ where $\|r_+\|^2 = 1 - \|r_-\|^2 = 1$ and $\ell \leq 1$. Thus, $J^\star = 1 \geq \cos\theta$ and $\mathcal{CO} \cap \mathcal{B} \neq \varnothing$ by Lemma 1.

Otherwise for $r_- \neq 0$, the feasibility of (14) requires the existence of $q_- \in \mathbb{R}^2$ that satisfies the two scalar equality constraints (14b) and (14c). In particular, (14) is feasible if and only if one of the lines (14b) passes through the circle (14c)

$$\begin{aligned} 1 - q_+^\top q_+ \geq \ \min\ \ & q_-^\top q_- && \text{(15a)}\\ \text{s.t.}\ \ & r_+^\top q_+ + r_-^\top q_- = \pm\ell && \text{(15b)}\end{aligned}$$

where q_+ is considered a fixed parameter and q_- is the decision variable. We will solve this multi-parametric program to obtain $q_-^\star$ as a function of q_+. The decision variable can be parameterized as $q_-^\star = \alpha_-^\star r_- + \alpha_\perp^\star r_\perp$ where $r_\perp \in \mathbb{R}^2$ is the vector orthogonal to $r_- \in \mathbb{R}^2$. Since $\alpha_\perp \neq 0$ increases the cost (15a) without helping satisfy the constraint (15b), we can conclude $\alpha_\perp^\star = 0$. Substituting $q_-^\star = \alpha_-^\star r_- \in \mathbb{R}^2$, the constraint (15b) we obtain

$$\alpha_-^\star = \frac{\pm\ell - r_+^\top q_+}{\|r_-\|^2}$$

where $r_- \neq 0$ and $\alpha_-^\star = (\ell - |r_+^\top q_+|)/\|r_-\|^2$ is the solution with the lower cost (15a). Thus, (15) reduces to the constraint

$$1 - q_+^\top q_+ \geq (\alpha_-^\star)^2 \|r_-\|^2 = (\ell - |r_+^\top q_+|)^2 / \|r_-\|^2$$

which can replace the constraints in (14), yielding

$$\begin{aligned} J^\star = \max\ \ & 2q_+^\top q_+ - 1\\ \text{s.t.}\ \ & q_+^\top\big(\|r_-\|^2 I + r_+ r_+^\top\big) q_+ - 2\ell|r_+^\top q_+| \leq \|r_-\|^2 - \ell^2.\end{aligned}$$

Note that the absolute value is redundant since the optimization problem will not choose $r_+^\top q_+ < 0$ since $-q_+$ has the same cost and provides extra slack in the constraint. Thus, we have obtained the following optimization problem which is equivalent to (11) when $\|r_+\| < \ell$

$$\begin{aligned} \max\ \ & 2q_+^\top q_+ - 1 && \text{(16a)}\\ \text{s.t.}\ \ & q_+^\top\big(\|r_-\|^2 I + r_+ r_+^\top\big) q_+ - 2\ell r_+^\top q_+ \leq \|r_-\|^2 - \ell^2. && \text{(16b)}\end{aligned}$$

This non-convex problem (16) has a semi-closed-form solution [15], specifically the extrema occur when the gradient of the cost (16a) is aligned with the gradient of the constraint (16b) i.e.

$$\mu q_+^\star = \big(\|r_-\|^2 + r_+ r_+^\top\big) q_+^\star - \ell r_+$$

where $\mu \in \mathbb{R}$ typically must be iteratively computed so that $q_+^\star$ satisfies (16b). Fortunately, r_+ is an eigenvector of the matrix $(|r_-\|^2 - \mu) I + r_+ r_+^\top$ with eigenvalue $\lambda = \|r_+\|^2 + \|r_-\|^2 - \mu = 1 - \mu$. Thus,

$$q_+^\star = \frac{\ell}{1-\mu} r_+ = \rho \frac{r_+}{\|r_+\|} \qquad (17)$$

for $r_+ \neq 0$ where $\rho = \ell\|r_+\|/(1-\mu)$ is a parameter we introduce to simplify the notation. To (non-iteratively) solve for ρ, we substitute (17) into (16b) and obtain

$$\rho^2 - 2\ell\|r_+\|\rho + \ell^2 - \|r_-\|^2 = 0$$

where r_+ is also an eigenvector of the matrix $|r_-\|^2 I + r_+ r_+^\top$ with eigenvalue $\lambda = 1$. Using the quadratic formula and noting that $\|r_+\|^2 = 1 - \|r_-\|^2$, we obtain

$$\rho = \ell\|r_+\| \pm \sqrt{1-\ell^2}\,\|r_-\|.$$

The cost (16a) is then $J^\star = 2\rho^2 - 1 = 2(\ell\|r_+\| + \sqrt{1-\ell^2}\|r_-\|)^2 - 1$ where the $+$ solution provides the larger cost (16a). According to Lemma 1, $\mathcal{CO} \cap \mathcal{B} = \varnothing$ if and only if $J^\star < c$. Rearranging terms in $J^\star$ we obtain the second part of the statement of this lemma.

Finally, we consider the degenerate case when $r_+ = 0$ in which case (17) is invalid. Instead, we immediately obtain $q_+ = 0$ and thus $J^\star = -1 \leq c = \cos\theta$. Therefore, $\mathcal{CO} \cap \mathcal{B} = \varnothing$ by Lemma 1. ■

Proof: [Proof of Theorem 1] Follow directly from Lemma 3. ■

References

[1] A. Weiss, C. Petersen, M. Baldwin, R. Erwin, and I. Kolmanovsky, "Safe positively invariant sets for spacecraft obstacle avoidance," *J. of Guidance, Control, and Dynamics*, 2015.

[2] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *Conf. on Decision and Control*, 2016.

[3] A. Weiss, C. Danielson, K. Berntorp, I. Kolmanovsky, and S. Di Cairano, "Motion planning with invariant set trees," in *Conf. on Control Technology and Applications*, 2017.

[4] K. Berntorp, A. Weiss, C. Danielson, S. Di Cairano, and I. Kolmanovsky, "Automated driving: Safe motion planning using positive-invariant sets," in *Intelligent Transportation Systems Conference*, 2017.

[5] K. Berntorp, R. Bai, K. F. Erliksson, C. Danielson, A. Weiss, and S. D. Cairano, "Positive invariant sets for safe integrated vehicle motion planning and control," *IEEE Transactions on Intelligent Vehicles*, 2020.

[6] C. Danielson, K. Berntorp, A. Weiss, and S. Di Cairano, "Robust motion-planning for uncertain systems with disturbances using the invariant-set motion-planner," *Accepted to Transactions on Automatic Control*, 2019.

[7] C. Danielson, K. Berntorp, S. Di Cairano, and A. Weiss, "Motion-planning for unicycles using the invariant-set motion-planner," in *American Control Conference*, 2020.

[8] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, 2001.

[9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int J of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[10] J.-P. Aubin, *Viability Theory*. Birkhauser Boston Inc., 1991.

[11] B. Wie, H. Weiss, and A. Arapostathis, "Quarternion feedback regulator for spacecraft eigenaxis rotations," *J. of Guidance, Control, and Dynamics*, vol. 12, no. 3, pp. 375–380, 1989.

[12] K. Lang, C. Klett, K. Hawkins, E. Feron, P. Tsiotras, and S. Phillips, "Formal verification applied to spacecraft attitude control," in *AIAA Scitech 2021 Forum*, 2021.

[13] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell, "Generating uniform incremental grids on so(3) using the hopf fibration," *Int J of robotics research*, vol. 29, no. 7, pp. 801–812, 06 2010.

[14] M. F. Carfora, "Interpolation on spherical geodesic grids: A comparative study," *Journal of Computational and Applied Mathematics*, vol. 210, no. 1, pp. 99–105, 2007.

[15] W. W. Hager, "Minimizing a quadratic over a sphere," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 188–208, 2001.