Construction of control barrier function and C^2 reference trajectory for constrained attitude maneuvers

Xiao Tan, and Dimos V. Dimarogonas

***Abstract*— Constrained attitude maneuvers have numerous applications in robotics and aerospace. In our previous work, a general framework to this problem was proposed with resolution completeness guarantee. However, a smooth reference trajectory and a low-level safety-critical controller were lacking. In this work, we propose a novel construction of a C^2 continuous reference trajectory based on Bézier curves on $SO(3)$ that evolves within predetermined cells and eliminates previous stop-and-go behavior. Moreover, we propose a novel zeroing control barrier function on $SO(3)$ that provides a safety certificate over a set of overlapping cells on $SO(3)$ while avoiding nonsmooth analysis. The safety certificate is given as a linear constraint on the control input and implemented in real-time. A remedy is proposed to handle the states where the coefficient of the control input in the linear constraint vanishes. Numerical simulations are given to verify the advantages of the proposed method.**

I. Introduction

The study of the attitude (orientation) control problem arised from early space and aerial applications and became prevalent in autonomous robotic systems. A recent trend in this field is to study this problem using Lie group theory [1]–[3], motivated by the fact that there exists no attitude parametrization other than the special orthogonal group $SO(3)$ that both globally and uniquely represents the rotational space and avoids singularities and the unwinding phenomenon. Many safe-critical applications, such as space telescopes observing some celestial regions while avoiding bright stars [4], and the anisotropy sensitive imaging and communication payloads on UAVs and AUVs, motivate further study of the attitude planning and control problem in the presence of orientation constraints.

There exist two main approaches for the constrained attitude maneuver problem: the potential-function [5]–[7] and the planning based methods [8]–[10]. By delicately designing a potential function, the feedback controller utilizes the negative gradient to guide the rotational movement. Generally speaking, potential-function based methods are easy to implement, but the state trajectory may get stuck at local minima (where the gradient vanishes) and requires convexity of the safe regions. On the contrary, planning-based methods try to find a feasible trajectory leading to the target state and then a tracking controller is utilized. This approach, however, mainly suffers from the lack of completeness guarantees, i.e., derive a solution if it exists, and safety guarantees, i.e., a certificate that the actual trajectory will not derivate from the reference and remain in the safe region.

In our previous work [11], a hierarchy framework was proposed for the constrained attitude maneuver problem consisting of 1) discretizing the rotation group $SO(3)$ into finite overlapping cells, 2) planning over the cells, and 3) reference trajectory generation and tracking control. Viewing the sampling step as the resolution level, we guarantee a feasible path is to be found in finite time when one exists at that resolution. However, the reference trajectory in [11] is constructed by the concatenation of geodesic paths and has to reach zero velocities at end points for each sub-maneuvers. This is a potential drawback as it requires the vehicle to stop and go from configuration to configuration. Moreover, no safety guarantee is developed for the low-level tracking controller.

In this work, we follow the framework in [11] and further construct a C^2 reference trajectory and develop a safety certificate by designing zeroing control barrier functions on $SO(3)$. The C^2 reference trajectory is generated by a Bézier curve on $SO(3)$. By choosing the controlling points carefully, we show that the constructed curve is of C^2 continuity, connects the initial and target orientations, and evolves within the set of given cells.

This paper has two additional contributions: 1) Noting that the safety region is a union of a set of overlapping cells, we formulate a smooth control barrier function and thus avoid the nonsmooth analysis as the case in [12]. The proposed formulation is at the cost of shrinking the safety region and this conservativeness can be explicitly adjusted by a user-defined parameter; 2) Since the Lie derivative of the control barrier function candidate vanishes at certain states, existing high-order control barrier function design methods [13], [14] are not directly applicable. To address this issue, we introduce a remedy with a proof to render the constraint on the control input feasible for all states in the safety region. All results are illustrated though relevant simulations. Detailed proofs of propositions and theorems in this paper as well as some technical discussions can be found in the extended version online [15].

II. Preliminaries and problem formulation

The set of real, non-negative real, and positive integer numbers are denoted as $\mathbb{R}, \mathbb{R}_{\geq}, \mathbb{N}$, respectively. $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. The 2-norm of a vector $x \in \mathbb{R}^n$ is $\|x\|_2 := \sqrt{x^\top x}$. I is the 3-dimensional identity matrix. The Frobenius norm of A is defined as $\|A\|_F =$

*This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the SSF COIN project, the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation.

Xiao Tan and Dimos V. Dimarogonas are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. xiaotan@kth.se, dimos@kth.se

$\mathbf{tr}(A^\top A)$, where $\mathbf{tr}(\cdot)$ denotes the trace of a matrix. The Lie derivatives of a function $h(x)$ for the system $\dot{x} = f(x) + g(x)u$ are denoted by $L_f h := \frac{\partial h}{\partial x} f(x)$ and $L_g h := \frac{\partial h}{\partial x} g(x)$, respectively. A continuous function $\alpha : (-b, a) \to (-\infty, \infty)$ is said to belong to *extended class* $\mathcal{K}$ for some $a, b > 0$ if it is strictly increasing and $\alpha(0) = 0$ [16].

Any rotation matrix is an element of the Special Orthogonal group $SO(3) := \{R \in \mathbb{R}^{3\times3} | R^\top R = RR^\top = I, \det(R) = 1\}$ which, when associated with the matrix multiplication operation, forms a Lie group. The associated Lie algebra, denoted by $\mathfrak{so}(3)$, consists of the set of all skew-symmetric 3×3 matrices, *i.e.,* $\mathfrak{so}(3) := \{\Omega \in \mathbb{R}^{3\times3} : \Omega^\top = -\Omega\}$. The Lie bracket for $\mathfrak{so}(3)$ is given as $[V, W] = VW - WV$, for any $V, W \in \mathfrak{so}(3)$. The map $[(\cdot)]_\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ and its inverse map $\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ are explicitly defined as $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \underset{(\cdot)^\vee}{\overset{[(\cdot)]_\times}{\rightleftharpoons}} [x]_\times = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}$. The Lie algebra $\mathfrak{so}(3)$ allows to represent rotation matrices on $SO(3)$ via the matrix exponential $\exp(\cdot)$, which admits an inverse logarithmic map $\log(\cdot)$. Details about the matrix exponential and logarithmic map can be found in [15].

Given any $R_1, R_2 \in SO(3)$ with $\mathbf{tr}(R_1^\top R_2) \neq -1$, the *geodesic path* between R_1 and R_2 is $R(\tau) = R_1 \exp(\tau \log(R_1^\top R_2)), 0 \le \tau \le 1$. The *angular distance* between R_1, R_2 is given by $\mathrm{d}(R_1, R_2) := \|\log(R_1 R_2^\top)\|_2$.

In [11], we proposed a general framework for the constrained attitude maneuver problem consisting of $SO(3)$ space partitioning, planning, and reference trajectory generation. We briefly recall the results here. Let the sampling set $U := \{R_1, R_2, \ldots, R_i, \ldots, R_n\}$ be a finite set with n elements in $SO(3)$ and let $\mathcal{N}' := \{1, 2, \ldots, n\}$ be an index set. For each $i \in \mathcal{N}'$, define the cell region S_i as the open ball centered at R_i with a radius $\theta \in (0, \pi/2)$, i.e., $S_i := \{R \in SO(3) : \mathrm{d}(R, R_i) < \theta\}, \forall i \in \mathcal{N}'$. The neighborhood set N_i of R_i is defined as $N_i := \{R \in U : \mathrm{d}(R, R_i) < 2\theta, R \neq R_i\}, \forall i \in \mathcal{N}'$. Cells S_i, S_j are *adjacent* if $S_i \cap S_j \neq \emptyset$.

By choosing U and θ such that the conditions in [11, Theorem 1] are satisfied, we have

i. For all $i \in \mathcal{N}', N_i \neq \emptyset$;
ii. For all $i, j \in \mathcal{N}', i \neq j$, we have $R_j \notin S_i$;
iii. For all $R_i \in U$, and all $R_j \in N_i$, $\theta < \mathrm{d}(R_i, R_j) < 2\theta$;
iv. $\bigcup_{i \in \mathcal{N}'} S_i = SO(3)$.

Lemma 1 ([11]). *For any cell $S_i, i \in \mathcal{N}'$ and two arbitrary points $R_{i1}, R_{i2} \in S_i$, the geodesic path between R_{i1} and R_{i2} is within S_i, i.e., for any $R_{i1} \in S_i, R_{i2} \in S_i$, $R(\tau) = R_{i1} \exp(\tau \log(R_{i1}^\top R_{i2})) \in S_i, 0 \le \tau \le 1$, holds.*

Lemma 2 ([11]). *The geodesic path between any two neighboring sampling rotations R_i and R_j is within $S_i \cup S_j$, i.e., $R(\tau) = R_i \exp(\tau \log(R_i^\top R_j)) \in S_i \cup S_j, 0 \le \tau \le 1$.*

We approximate a generic safe attitude zone on $SO(3)$ by a set of cells $\{S_i\}, i \in \mathcal{N}, \mathcal{N} \subset \mathcal{N}'$ and a graph search algorithm is utilized that gives out a sequence of cells whenever feasible at the given resolution level. Without loss of generality, by re-labeling the cells, we assume that the initial orientation $R_0 \in S_1$, the target orientation $R_f \in S_m$, S_i and S_{i+1} are adjacent cells for $i \in \{1, \cdots, m-1\}$. Based on Lemmas 1,2, a center-to-center attitude maneuver was then designed, as illustrated in blue dash line in Fig. 1. Though the reference trajectory is guaranteed to be within the feasible region, it requires the rigid-body to reach zero velocities at the end points for each sub-maneuvers.

A. Problem formulation

The attitude dynamics of a rigid body are given by

$$\begin{cases} \dot{R} = R[\omega]_\times, \\ J\dot{\omega} + [\omega]_\times J\omega = u, \end{cases} \tag{1}$$

where the attitude $R \in SO(3)$, $\omega \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame, J is the constant and known inertia matrix and $u \in \mathbb{R}^3$ is the input torque. Given a set of cells $\{S_i\}, i \in \mathcal{N}, \mathcal{N} \subset \mathcal{N}'$, we call a trajectory $\gamma : t \mapsto R(t)$ is *safe* if the trajectory always evolves within $\cup_{i \in \mathcal{N}} S_i$.

The control scheme consists of two parts: reference generation and trajectory tracking.

Problem 1. (reference generation) Given $\{S_i\}, i \in \mathcal{N}, \mathcal{N} \subset \mathcal{N}'$ such that $\cup_{i\in\mathcal{N}} S_i$ is path-connected. For any given $R_0, R_f \in \cup_{i\in\mathcal{N}} S_i$, find a C^2 curve $\gamma : \mathbb{R}_\ge \to \cup_{i\in\mathcal{N}} S_i$ such that $d\gamma/dt(0) = d\gamma/dt(T) = 0, D^2\gamma/dt^2(0) = D^2\gamma/dt^2(T) = 0$[1], $\gamma(0) = R_0, \gamma(t) = R_f, \forall t \ge T$.

Problem 2. (trajectory tracking) Given a curve $\gamma : \mathbb{R}_\ge \to \cup_{i\in\mathcal{N}} S_i$, design a control law u for the system (1) such that $R(t) \in \cup_{i\in\mathcal{N}} S_i$ for $t \ge 0$ and $\lim_{t\to\infty} R(t) = \gamma(t)$.

In the following, we will solve Problem 1 and Problem 2 in Section III and Section IV, respectively.

III. Bézier curve construction over cells

In this section, we construct a reference trajectory based on Bézier curve on $SO(3)$ that solves Problem 1. Bézier curve is chosen here because De Casteljau Algorithm, which generates Bézier curves, is in essence a geometric construction, and naturally generalizes to $SO(3)$ manifold, while other splines are either not defined nor not easy to compute.

A. De Casteljau Algorithm on $SO(3)$

We briefly recall De Casteljau Algorithm from [18] as follows. Taking the geodesics on $SO(3)$ as the analog of straight lines, De Casteljau Algorithm connects two points in $SO(3)$ via an iterative linear interpolation process. Let $n+1$ ordered points of $SO(3)$ be $\{x_0, x_1, \cdots, x_n\}$. The sequence of curves is defined recursively on $SO(3)$ as

$$x_i^k(\tau) = x_{i-1}^{k-1}(\tau) \exp(\tau \log([x_{i-1}^{k-1}(\tau)]^\top x_i^{k-1}(\tau))), \quad k = 0, 1, \cdots, n, \quad i = k, k+1, \cdots, n, \tag{2}$$

where $x_i^0(\tau) = x_i$. The Bézier curve is then given by

$$x_n^n(\tau) = x_{n-1}^{n-1}(\tau) \exp(\tau \log([x_{n-1}^{n-1}(\tau)]^\top x_n^{n-1}(\tau))). \tag{3}$$

[1]For a curve $\gamma : \mathbb{R} \to SO(3)$, $D^2\gamma/dt^2$ represents the geometric acceleration instead of the second-order total derivatives, following the terminology in [17].

Lemma 3 ([19]). *Let $n+1$ ordered points of $SO(3)$ be $\{x_0, x_1, \cdots, x_n\}$. The corresponding Bézier curve generated from* (3) *satisfies the following boundary conditions:* $x_n^n(0) = x_0, x_n^n(1) = x_n, \frac{d}{d\tau}x_n^n(\tau)|_{\tau=0} = nx_0V_0, \frac{d}{d\tau}x_n^n(\tau)|_{\tau=1} = nx_nV_{n-1}, \frac{D^2}{d\tau^2}x_n^n(\tau)|_{\tau=0} = n(n-1)x_0\Upsilon_0^{-1}(V_1-V_0), \frac{D^2}{d\tau^2}x_n^n(\tau)|_{\tau=1} = n(n-1)x_n\Upsilon_1^{-1}(V_{n-1}-V_{n-2})$*, where* $V_i = \log(x_i^\top x_{i+1}) \in \mathfrak{so}(3), i = 0, 1, \cdots, n-1$*,* Υ_0^{-1} *and* Υ_1^{-1} *are respectively the inverses of the operators* $\Upsilon_0(W) = \int_0^1 \exp(u\mathrm{ad}V_0)W du, \Upsilon_1(W) = \int_0^1 \exp(-u\mathrm{ad}V_{n-1})W du$*.*

For any $W \in \mathfrak{so}(3)$, the operator $\Upsilon_0 : \mathfrak{so}(3) \to \mathfrak{so}(3)$ is given explicitly by the power series $\int_0^1 \exp(u\mathrm{ad}V_0)W du = \int_0^1 W + u[V_0, W] + \frac{u^2}{2!}[V_0, [V_0, W]] + \cdots du$. The operator $\Upsilon_1(W)$ is given in a similar way. It can be easily verified that both operators Υ_0, Υ_1 are linear transformations on $\mathfrak{so}(3)$, i.e., $\Upsilon_i(aW) = a\Upsilon_i(W), \Upsilon_i(W+V) = \Upsilon_i(W) + \Upsilon_i(V)$ for $W, V \in \mathfrak{so}(3), a \in \mathbb{R}, i = 1, 2$. In [19], it is shown that the inverse operator Υ_i^{-1} always exists for $i = 1, 2$.

Lemma 3 introduces the analytical expression of the velocity and geometric acceleration at the boundary point that will facilitate our construction of the reference trajectory.

Remark 1. Note that the De Casteljau algorithm in (2) is not well-defined for arbitrary points $x_0, x_1, \cdots, x_n$ on $SO(3)$ when $\mathbf{tr}([x_{i-1}^{k-1}(\tau)]^\top x_i^{k-1}(\tau)) = -1$ occurs.

B. Bézier curve construction in one cell

In this subsection, we demonstrate the procedure to design the controlling points and analyze the properties of the constructed Bézier curve.

Given a cell $S_i, i \in \mathcal{N}$ with center point x_2 and two arbitrary points $x_0, x_4 \in S_i$, the curve $c_{x_0,x_2,x_4} : [0, 1] \to SO(3)$ is generated as follows: first add controlling points x_1, x_3 as the midpoints of x_0, x_2, and x_2, x_4, respectively; then apply De Casteljau algorithm with $n = 4$. The construction is given in Algorithm 1:

Algorithm 1 Bézier curve construction in one cell.

Input: start point x_0, cell center x_2, end point x_4
Output: curve c_{x_0,x_2,x_4}

1: $x_1 \leftarrow x_0 \exp(1/2 \log(x_0^\top x_2))$
2: $x_3 \leftarrow x_2 \exp(1/2 \log(x_2^\top x_4))$
3: calculate a sequence of curves recursively as in (2) given the ordered points $\{x_0, x_1, x_2, x_3, x_4\}$ with $n = 4$
4: **return** $c_{x_0,x_2,x_4} \leftarrow x_4^4$

Noticing that $V_0 = \log(x_0^\top x_1) = 1/2 \log(x_0^\top x_2), V_1 = \log(x_1^\top x_2) = 1/2 \log(x_0^\top x_2)$, we have $V_0 = V_1$. Similarly, $V_2 = V_3$. From Lemma 3, we can easily check that $c_{x_0,x_2,x_4}(0) = x_0, c_{x_0,x_2,x_4}(1) = x_4$. The velocities at the boundary point are $\frac{d}{d\tau}c_{x_0,x_2,x_4}(0) = 2x_0 \log(x_0^\top x_2), \frac{d}{d\tau}c_{x_0,x_2,x_4}(1) = 2x_4 \log(x_2^\top x_4)$, and the geometric accelerations are given by $\frac{D^2}{d\tau^2}c_{x_0,x_2,x_4}(0) = 12x_0\Upsilon_0^{-1}(V_1 - V_0) = 0, \frac{D^2}{d\tau^2}c_{x_0,x_2,x_4}(1) = 12x_n\Upsilon_1^{-1}(V_3 - V_2) = 0$, noticing that $V_1 - V_0 = V_3 - V_2 = 0$ and $\Upsilon_0^{-1}, \Upsilon_1^{-1}$ being linear transformations.

In addition to these explicitly expressed velocities and geometric accelerations at the endpoints, we have another nice property of the constructed curve c_{x_0,x_2,x_4}.

Proposition 1. *Given arbitrary $n+1$ ordered points $\{x_0, x_1, x_2, \cdots, x_n\}$ such that $x_i \in S$, $i = 0, 1, \cdots, n$, where S is a ball region in $SO(3)$ with radius $\theta \in (0, \pi/2)$. The Bézier curve $x_n^n(\tau)$ generated from* (3) *always exists and evolves in S, i.e., $x_n^n(\tau) \in S, 0 \le \tau \le 1$.*

A conclusion is that the curve c_{x_0,x_2,x_4} constructed from Algorithm 1 is well-defined and evolves within the cell.

C. Bézier curve construction in a set of cells

Now we apply Algorithm 1 to generate a curve evolving among a set of cells. In the following, we use the notation $c_{x_0,x_2,x_4}(\tau) : [0, 1] \to SO(3)$ to denote the curve generated from Algorithm 1 given the three points x_0, x_2, x_4.

Proposition 2. *Assume that R_0, R_f are the initial and target orientations, respectively, $R_0 \in S_1$, $R_f \in S_m$, and assume there exists a sequence of cells $S_1S_2 \cdots S_m$ such that S_iS_{i+1} are adjacent for $i = 1, 2, \cdots, m-1$. Then, a curve $c : \mathbb{R} \supset [0, m] \to SO(3)$ defined as*

$$c(\tau) = \begin{cases} c_{R_0,R_1,R_{1,2}}(\tau), & \tau \in [0, 1), \\ c_{R_{i-1,i},R_i,R_{i,i+1}}(\tau - i + 1), & \tau \in [i-1, i), \\ & i \in \{2, 3, \cdots, m-1\}, \\ c_{R_{m-1,m},R_m,R_f}(\tau - m + 1), & \tau \in [m-1, m], \end{cases} \tag{4}$$

where R_i is the center of cell S_i, $R_{i,i+1} := R_i \exp(1/2 \log(R_i^\top R_{i+1}))$, has the following properties: i) $c(0) = R_0, c(m) = R_f$; ii) $c(\tau)$ is a C^2 curve; iii) $c(\tau) \in \cup_{i=1}^m S_i$ for $\tau \in [0, m]$.

D. Time re-parameterization

In order to obtain a reference trajectory that solves Problem 1, let τ be a smooth function of time, i.e., $\tau : \mathbb{R}_{\ge} \to [0, m]$ that rescales the trajectory $c : [0, m] \to SO(3)$ to the time domain $\gamma := c \circ \tau : \mathbb{R}_{\ge} \to SO(3)$.

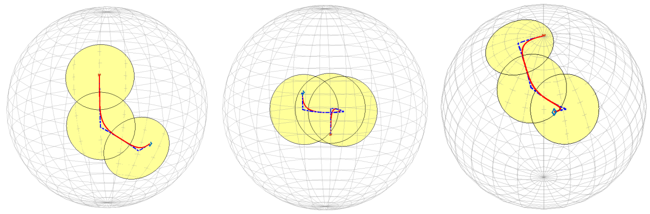
Here we choose a smooth transition function

$$s(x) = \begin{cases} 0 & x \in (-\infty, 0), \\ \frac{\rho(x)}{\rho(x)+\rho(1-x)} & x \in [0, 1), \\ 1 & x \in [1, \infty) \end{cases} \tag{5}$$

with $\rho(x) := (1/x)e^{-1/x}$.

Theorem 1. *Given a sequence of cells $S_1, S_2, \cdots, S_m$ such that $R_0 \in S_1, R_f \in S_m, S_iS_{i+1}$ are adjacent. Choose $c : [0, m] \to SO(3)$ defined in* (4) *and $\tau(t) := ms(t/T)$ with $s(\cdot)$ in* (5)*. The curve $\gamma := c \circ \tau : \mathbb{R}_{\ge} \to SO(3)$ is continuously differentiable, and satisfies $\gamma(0) = R_0, \gamma(T) = R_f, d\gamma/dt(0) = d\gamma/dt(T) = 0, D^2\gamma/dt^2(0) = D^2\gamma/dt^2(T) = 0, \gamma(t) \in \cup_{i\in\mathcal{N}} S_i$ for $t \ge 0$.*

Remark 2. Although in this work we set the initial and terminal velocities to be zero, the presented method can be extended to problems with non-zero velocity boundary conditions by manipulating the controlling points in the cells



(a) Trajectory of x-axis. (b) Trajectory of y-axis. (c) Trajectory of z-axis.

Fig. 1: Comparison of the trajectories of body-fixed axes: $c \circ \tau$ in red and the one from [11] in blue.

S_0, S_m and the time-reparametrization function $s(\cdot)$. This is a straightforward extension and details are omitted here.

We demonstrate in Fig. 1 the constructed reference curve $c \circ \tau$ (red line) and the curve from [11] (blue dash line) for comparison. The data is given in the simulation section. It is seen that a smoother attitude maneuver is obtained compared to that of [11]. The trajectory in [11] needs to reach zero velocities at intermediate points, which is avoided in the new construction.

IV. Control barrier function design

In this section, we present the procedure to construct a zeroing control barrier function that guarantees the actual attitude trajectory evolves within $\cup_{i\in\mathcal{N}} S_i$.

We start the barrier function design from one cell. For an arbitrary cell S_i, define a function $r_i : SO(3) \to \mathbb{R}$

$$r_i(R) = \epsilon - \|R_i - R\|_F^2/2, \tag{6}$$

where constant $\epsilon := 4\sin^2(\theta/2)$, R_i, θ are the cell center and radius of cell S_i, respectively. It is easy to show that $r_i(R) > 0$ if and only if $R \in S_i$, in view of the fact that $\|v - w\|_F = 2\sqrt{2}\sin(\mathrm{d}(v,w)/2)$ holds for $v, w \in SO(3)$. If we need to constrain the trajectory in cell S_i, $r_i(R)$ is a natural candidate as a zeroing control barrier function as it indicates how far the state is from the cell boundary. Note that there are many alternatives $r_i(R)$ to (6), for example, $r_i(R) = \theta - \mathrm{d}(R, R_i)$. The reason we choose $r_i(R)$ as in (6) is merely to simplify the expression of its derivatives.

To ensure the actual attitude trajectory evolves within $\cup_{i\in\mathcal{N}} S_i$, we need for every time instant $t \geq 0$, there exists at least one $i \in \mathcal{N}$ that $R(t) \in S_i$, i.e.,

$$\max_{i\in\mathcal{N}}(r_i(R(t))) > 0, \quad \text{for } t \geq 0. \tag{7}$$

This max operation would lead to nonsmooth analysis and a complex formulation [12]. In the following, we will show how to circumvent the nonsmooth analysis.

Define

$$h(R) = \sum_{i\in\mathcal{N}} s(r_i(R)/\epsilon) - \delta, \tag{8}$$

where $\delta > 0$ is a user-defined constant, and $s(\cdot)$ is given in (5). The associated constrained set is thus $C_h^R = \{R \in SO(3) : h(R) \geq 0\}$. Since $\cup_{i\in\mathcal{N}} S_i = \{R : h(R) > -\delta\}$, it is straightforward that $C_h^R \subset \cup_{i\in\mathcal{N}} S_i$, and the constant δ represents the safety margin. The conservativeness is illustrated in Fig. 2 in the planar case. For any given C^2 curve $c \circ \tau : \mathbb{R}_{\geq} \to \cup_{i\in\mathcal{N}} S_i$, we can find a safety margin (i.e., δ) such that the curve $c \circ \tau$ evolves within C_h^R. In the following, we thus assume $c \circ \tau(t) \in C_h^R$ for $t \geq 0$.

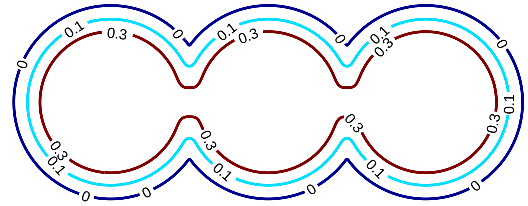


Fig. 2: Illustration of C_h in the planar case with different conservativeness δ's.

We embed the attitude dynamics in (1) in a higher dimensional Euclidean space as

$$\dot{x} := f(x) + gu, \tag{9}$$

where $x = (r_{11}, r_{12}, \cdots, r_{33}, \omega_1, \omega_2, \omega_3) \in \mathbb{R}^{12}, f(x) = (r_{12}\omega_3 - r_{13}\omega_2; r_{13}\omega_1 - r_{11}\omega_3; r_{11}\omega_2 - r_{12}\omega_1; r_{22}\omega_3 - r_{23}\omega_2; r_{23}\omega_1 - r_{21}\omega_3; r_{21}\omega_2 - r_{22}\omega_1; r_{32}\omega_3 - r_{33}\omega_2; r_{33}\omega_1 - r_{31}\omega_3; r_{31}\omega_2 - r_{32}\omega_1; J^{-1}(-[\omega]_\times J\omega)) \in \mathbb{R}^{12}, g = \begin{pmatrix} 0_{9\times 3} \\ J^{-1} \end{pmatrix}$. This is equivalent to (1) by rewriting the attitude dynamics in a vectorized manner. Note that for $r_{11}, r_{12}, \cdots, r_{33}$, there exist 6 implicit equality constraints since they are elements of a rotation matrix. We denote the corresponding 6−dimensional submanifold $C_{TSO(3)} := \{x \in \mathbb{R}^{12} : \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix} \in SO(3)\}$. Moreover, if $x(0) \in C_{TSO(3)}$, then given any control signal u which is Lipschitz continuous in x, the solution of the dynamical system (9) satisfies $x(t) \in C_{TSO(3)}$ for $t \geq 0$. This fact can be easily obtained considering that (9) and (1) are equivalent. h in (8) is thus a function of the system states x, in particular, of the states $(x_1, x_2, \cdots, x_9)$. The associate constrained set is $C_h := \{x \in C_{TSO(3)} : h(x) \geq 0\}$.

For all $x \in C_h$, we obtain $L_g h = 0$, and $L_g L_f h(x)$ may vanish at some points in C_h (see Appendix for derivations). Here we note that the higher-order control barrier function design developed in [13], [14] is not directly applicable as a result. A key observation regarding the singularity set $\mathcal{D} = \{x \in C_h : L_g L_f h(x) = 0\}$ is given below:

Proposition 3. *Let* $\mathcal{D} = \{x \in C_h : L_g L_f h(x) = 0\}$. *Then, there exists a constant* $\xi > 0$ *such that* $\inf_{x\in\mathcal{D}} h(x) \geq \xi$.

In the following a remedy is derived to handle the singularity set $\mathcal{D} = \{x \in C_h : L_g L_f h(x) = 0\}$ effectively. Denote the associated set $C_{h,\xi} = \{x \in C_{TSO(3)} : h(x) \geq \xi\}$. Let $\chi(\cdot)$ be a twice differentiable function satisfying the following properties: $\chi(0) = 0, \chi(a) = 1$ for $a \geq 1, \frac{d\chi}{d\iota}(a) > 0$ for $a < 1$. Then we smoothly truncate $h(x)$ by the upper bound ξ, i.e.,

$$b(x) = \chi(h(x)/\xi). \tag{10}$$

It is easy to verify that $C_h = C_b := \{x \in C_{TSO(3)} : b(x) \geq$

$0\}$. In the following, the forward invariance of the set C_b is shown instead.

We adopt the procedure of the higher-order control barrier function design as in [13], [14]. The idea is briefly presented here: from Brezis version of Nagumo's theorem [20, Theorem 4], the forward invariance of the set C_b is guaranteed by the condition $\dot{b}(x) \geq -\alpha(b(x))$ for all $x \in C_b$, where α is a continuously differentiable, extended class $\mathcal{K}$ function. Note that $L_g b = 0$, then $b_1(x) := \dot{b}(x) + \alpha(b(x)) = L_f b + \alpha(b(x))$ is still a function of the state x. To guarantee the forward invariance of the set $C_{b1} := \{x \in C_{TSO(3)} : b_1(x) \geq 0\}$, using Brezis version of Nagumo's theorem again, we have the new condition $\dot{b}_1(x) \geq -\beta(b_1(x))$ for all $x \in C_b \cap C_{b1}$, where β is a continuously differentiable, extended class $\mathcal{K}$ function. Thus, the condition we will enforce in real-time is given as $L_g b_1(x) u + L_f b_1(x) + \beta(b_1(x)) \geq 0$ for all $x \in C_b \cap C_{b1}$. The feasibility result is as follows:

Proposition 4. *The inequality condition on* $u \in \mathbb{R}^3$

$$L_g b_1(x) u + L_f b_1(x) + \beta(b_1(x)) \geq 0 \tag{11}$$

is feasible for all $x \in C_b \cap C_{b1}$.

Suppose a nominal bounded control input $u_{nom}(x)$, Lipschitz continuous in x, has been designed for the attitude dynamics and the closed-loop solution tracks the constructed reference trajectory γ. We modify the control input online to account for the safety constraints. Concretely, the controller is given by the quadratic program below:

$$\begin{aligned} u(x) &= \arg\min_{u \in \mathbb{R}^3} \|u - u_{nom}\|^2 \\ \text{s.t.} \quad & L_g b_1(x) u + L_f b_1(x) + \beta(b_1(x)) \geq 0. \end{aligned} \tag{12}$$

This formulation reflects that the safety constraint has priority over the tracking mission.

Theorem 2. *For the attitude control system in* (1)*, the controller* (12) *renders the set* $C_b \cap C_{b1}$ *forward invariant.*

Remark 3. Compared to the nonsmooth barrier function design in [12], we formulate a smooth control barrier function and thus avoid the nonsmooth analysis. This formulation comes at the cost of conservativeness in terms of the set difference between $\cup_{i \in \mathcal{N}} S_i$ and C_h^R. Note that the conservativeness can be explicitly adjusted by choosing a proper δ and serves as a safety margin in many robotic applications.

Remark 4. Although [21] has studied the application of barrier functions in constrained attitude control problems, the proposed framework in this paper is generally more advantageous as 1)zeroing instead of reciprocal barrier function is used, which is well-defined even outside of the safety set and is guaranteed to be robust to small model perturbations [16]; 2)here we deal with safety regions of arbitrary shape and the feasibility to the online optimization is guaranteed.

Remark 5. In Theorem 2 we guarantee the forward invariance of the set $C_b \cap C_{b1}$ instead of C_h. This does not cause conservativeness. For any $h(x(0)) > 0$, or equivalently, $b(x(0)) > 0$, there always exists an extended class $\mathcal{K}$ function $\alpha(\cdot)$ such that $b_1(x(0)) = L_f b(x(0)) + \alpha(b(x(0))) > 0$. In this way, C_{b1} is constructed such that $x(0) \in C_b \cap C_{b1}$.

V. Simulations

In this section, we demonstrate the favorable properties of the constructed reference trajectory and the designed zeroing control barrier function. The scenario is given as follows: the inertia matrix of the rigid body is given by $J = \begin{bmatrix} 5.5 & 0.06 & -0.03 \\ 0.06 & 5.5 & 0.01 \\ -0.03 & 0.01 & 0.1 \end{bmatrix} \mathrm{kg} \cdot \mathrm{m}^2$. The target attitude is set as $R_f = I$, and the center points of the sampling cells are given as $R_3 = \exp(15°/180° \times \pi[e_1]_\times)$, $R_2 = \exp(30°/180° \times \pi[e_2]_\times) R_3$, $R_1 = \exp(30°/180° \times \pi[0, 0.447, 0.894]_\times) R_2$, and the initial attitude $R_0 = \exp(10°/180° \times \pi[e_1]_\times) R_1$. The radius of the cells is set as $\theta = 0.3491$ rad $(20°)$. The settling time is $T = 40s$. Based on these data, we show the constructed reference trajectory in red in Fig. 1.

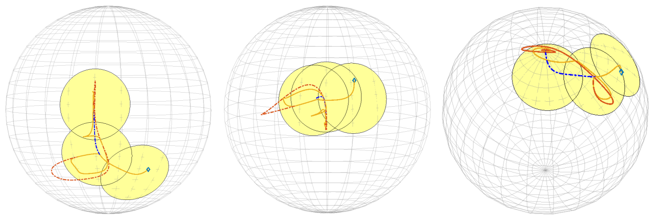
In what follows, we use the saturated controller from [2] as the nominal controller in (12): $u_{nom} = J\tilde{R}^\top \dot{\omega}_r + [\tilde{R}^\top \omega_r]_\times J \tilde{R}^\top \omega_r - k_1 (\tilde{R} - \tilde{R}^\top)^\vee - k_2 \tanh(\tilde{\omega})$, where $\tilde{R} = R_r^\top R$, $\tilde{\omega}(t) = \omega - \tilde{R}^\top \omega_r$, R_r, ω_r are the reference orientation and reference angular velocity obtained from the constructed trajectory γ, respectively, $k_1, k_2 > 0$ are tuning gains and $\tanh(\cdot)$ is the element-wise hyperbolic tangent function.

In the simulations, we augment the nominal control signal with an additive signal $u_{add} = 0.3 * \big(\sin(2\pi \frac{t-20}{5}), \sin(\pi \frac{t-20}{5}), -\sin(\pi \frac{t-20}{5})\big)$ for the time interval $t \in [20, 25]$. This control signal simulates, for example, a human input to the system that could lead to a deviation from the reference trajectory and may even drive the states out of the safe cells. The controller parameters are set as $k_1 = 0.2, k_2 = 0.2$. The parameters in the control barrier function are chosen as $\delta = 0.1, \xi = 0.7, \alpha(x) = \beta(x) = x, \chi(x) = \begin{cases} (x-1)^3+1, & \text{if } x \leq 1; \\ 1, & \text{if } x > 1. \end{cases}$

Fig. 3 shows the trajectories in three cases: 1) no additive signal is applied and the control barrier function exists (in blue); 2) additive signal is applied and control barrier function does not exist (in dark red); 3) additive signal is applied and control barrier function exists (in yellow). It is shown that without the additive control signal, the system trajectory is similar to the reference trajectory in Fig. 1. However, when the additive signal exists and only the nominal controller is applied, the state deviates from the previous trajectory and runs out of the safety cells. Once the control barrier formulation is applied, the resulting trajectory remains in the safety set.

VI. Conclusion

In this paper, we construct a C^2 reference trajectory on $SO(3)$ and develop a safety certificate utilizing the control barrier function formulation for constrained attitude control problems, following the framework of our previous work in [11]. To construct the reference trajectory, we first design the controlling points for Bézier curve generation on $SO(3)$, which is then time re-parametrized to satisfy boundary conditions. The reference trajectory is shown to be C^2 continuous, connect the initial and target orientations,



(a) Trajectory of x-axis. (b) Trajectory of y-axis. (c) Trajectory of z-axis.

Fig. 3: Comparison of the trajectories of body-fixed axes in three cases.

and evolve within the predefined safe regions. Moreover, a smooth control barrier function is designed over a set of overlapping cells to circumvent the non-smooth analysis in previous works. The safety certificate is given as a linear constraint on the control input. This paper also provides a remedy to handle the states when the singularity of the linear constraint occurs.

APPENDIX

We collect in this appendix all the results supporting the derivations and claims of the main part of the paper.

From simplicity, denote the auxiliary variables $q = (x_1, x_2, \cdots, x_9)$, $\omega = (\omega_1, \omega_2, \omega_3)$ and the elements in R_i as $\begin{pmatrix} x_1^i & x_2^i & x_3^i \\ x_4^i & x_5^i & x_6^i \\ x_7^i & x_8^i & x_9^i \end{pmatrix}$, and let $[A]_{i,j}$ be the (i,j)th element of matrix A. Thus the state variable is rewritten as $x = (q, \omega)$. As q is the stacked vector of the rotation matrix R, we use q and R interchangeably to denote the attitude state in the following. We obtain that, for $r_i(q)$ defined in (6), $\left[\frac{\partial r_i}{\partial x}\right]_j = \begin{cases} -x_j + x_j^i, & j=1,2,\cdots,9, \\ 0, & j=10,11,12. \end{cases}$ From $h(q)$ given in (8), we further have $\frac{\partial h}{\partial x} = (\frac{\partial h}{\partial q}, \frac{\partial h}{\partial \omega})$ with $\frac{\partial h}{\partial q} = \sum_{i\in\mathcal{N}} \frac{\partial s(r_i(q)/\epsilon)}{\partial q} := \frac{1}{\epsilon}\sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i} \frac{\partial (r_i(q))}{\partial q}$, $\frac{\partial h}{\partial \omega} = \sum_{i\in\mathcal{N}} \frac{\partial s(r_i(q)/\epsilon)}{\partial \omega} = 0$, where $\eta_i(q) := r_i(q)/\epsilon$ for brevity.

With f in (9), we further obtain $L_f h = \frac{\partial h}{\partial x} \cdot f = \frac{1}{\epsilon}\sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i}\frac{\partial (r_i(x))}{\partial x} \cdot f = \frac{1}{\epsilon}\sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i}(e_{32}^i - e_{23}^i, e_{13}^i - e_{31}^i, e_{21}^i - e_{12}^i) \cdot \omega := \frac{1}{\epsilon}\sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i}\omega^\top e^i(q)$, where $e_{j,k}^i(q) = [R^\top R_i]_{j,k}$ for $j,k = 1,2,3$. Similarly, we have $L_g h = \frac{\partial h}{\partial x} \cdot g$. Noticing that g in (9) and $\frac{\partial h}{\partial \omega} = 0$, we obtain $L_g h = \frac{\partial h}{\partial q} \cdot 0_{9\times 3} + \frac{\partial h}{\partial \omega} \cdot J^{-1} = 0$. Moreover, we can calculate that $L_g L_f h = \frac{\partial L_f h}{\partial x} \cdot g = \frac{\partial L_f h}{\partial q} \cdot 0_{9\times 3} + \frac{\partial L_f h}{\partial \omega} \cdot J^{-1} = \frac{1}{\epsilon}\frac{\partial \sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i}\omega^\top e^i(q)}{\partial \omega} \cdot J^{-1}$. Note that $\eta_i(q)$ only relies on q, and thus $L_g L_f h = \frac{1}{\epsilon}\sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i}\frac{\partial \omega^\top e^i(q)}{\partial \omega} \cdot J^{-1} = \frac{1}{\epsilon}\sum_{i\in\mathcal{N}} \frac{ds}{d\eta_i} e^i(q)^\top J^{-1}$. From (10), we obtain $L_f b = \frac{d\chi}{d\iota}(\frac{h(x)}{\xi})\frac{\partial h/\xi}{\partial x} \cdot f = \frac{1}{\xi}\frac{d\chi}{d\iota} L_f h$ and $L_g b = \frac{1}{\xi}\frac{d\chi}{d\iota} L_g h = 0$, where $\iota := h(x)/\xi$ for brevity.

As $b_1(x) = L_f b + \alpha(b(x))$, we obtain $L_f b_1 = \frac{1}{\xi}\left(\frac{1}{\xi}\frac{d^2\chi}{d\iota^2}(L_f h)^2 + \frac{d\chi}{d\iota} L_f^2 h\right) + \frac{d\alpha}{db}(b(x)) L_f b$ with $L_f^2 h = \frac{1}{\epsilon}\sum_{i\in\mathcal{N}}\left(\frac{1}{\epsilon}\frac{d^2 s}{d\eta_i^2}(\omega^\top e^i(q))^2 + \frac{ds}{d\eta_i}\frac{\partial \omega^\top e^i(q)}{\partial x} \cdot f\right)$ and $L_g b_1 = \frac{1}{\xi}\frac{d\chi}{d\iota} L_g L_f h$.

ACKNOWLEDGMENT

The authors thank Prof. Fatima Silva Leite for the inspiring discussion on the De Casteljau algorithm on $SO(3)$.

REFERENCES

[1] D. E. Koditschek, "Application of a new Lyapunov function to global adaptive attitude tracking," in *Proceedings of the 27th IEEE Conference on Decision and Control,*, 1988, pp. 63–68.

[2] T. Lee, "Robust adaptive attitude tracking on $SO(3)$ with an application to a quadrotor uav," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1924–1930, 2012.

[3] S. Berkane, A. Abdessameud, and A. Tayebi, "Hybrid global exponential stabilization on $SO(3)$," *Automatica*, vol. 81, pp. 279–285, 2017.

[4] H. B. Hablani, "Attitude commands avoiding bright objects and maintaining communication with ground station," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 6, pp. 759–767, 1999.

[5] 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, pp. 2578–2592, 2014.

[6] S. Kulumani and T. Lee, "Constrained geometric attitude control on $SO(3)$," *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2796–2809, 2017.

[7] Q. Hu, B. Chi, and M. R. Akella, "Anti-unwinding attitude control of spacecraft with forbidden pointing constraints," *Journal of Guidance, Control, and Dynamics*, pp. 1–13, 2018.

[8] E. Frazzoli, M. Dahleh, E. Feron, and R. Kornfeld, "A randomized attitude slew planning algorithm for autonomous spacecraft," in *AIAA Guidance, Navigation, and Control Conference*, 2001.

[9] H. C. Kjellberg and E. G. Lightsey, "Discretized quaternion constrained attitude pathfinding," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 11, pp. 713–718, 2015.

[10] J. D. Biggs and L. Colley, "Geometric attitude motion planning for spacecraft with pointing and actuator constraints," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 7, pp. 1672–1677, 2016.

[11] X. Tan, S. Berkane, and D. V. Dimarogonas, "Constrained attitude maneuvers on SO(3) : Rotation space sampling, planning and low-level control," *Automatica*, vol. 112, p. 108659, 2020.

[12] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.

[13] W. S. Cortez and D. V. Dimarogonas, "Correct-by-design control barrier functions for Euler-Lagrange systems with input constraints," in *2020 American Control Conference (ACC)*, 2020.

[14] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 27–34.

[15] X. Tan and D. V. Dimarogonas, "Construction of control barrier function and C^2 reference trajectory for constrained attitude maneuvers," *arXiv*, 2020. [Online]. Available: http://arxiv.org/abs/2008.08921

[16] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[17] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Springer Science & Business Media, 2004, vol. 49.

[18] F. Park and B. Ravani, "Bézier curves in Riemannian manifolds and lie groups with kinematics applications," *Journal of mechanical designs*, vol. 117, no. 1, pp. 36–40, 1995.

[19] P. Crouch, G. Kun, and F. S. Leite, "The De Casteljau algorithm on Lie groups and spheres," *Journal of Dynamical and Control Systems*, vol. 5, no. 3, pp. 397–429, 1999.

[20] R. Redheffer, "The theorems of Bony and Brezis on flow-invariant sets," *The American Mathematical Monthly*, vol. 79, no. 7, pp. 740–747, 1972.

[21] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control Lyapunov and control barrier functions for systems evolving on manifolds," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2038–2044.