

Quadrotor Control on $SU(2) \times R^3$ with SLAM Integration

Citation for published version (APA):

Greiff, M., Persson, P., Sun, Z., Åström, K., & Robertsson, A. (2021). Quadrotor Control on $SU(2) \times R^3$ with SLAM Integration. *arXiv*, 2021, Article 2110.01099. <https://doi.org/10.48550/arXiv.2110.01099>

DOI:

[10.48550/arXiv.2110.01099](https://doi.org/10.48550/arXiv.2110.01099)

Document status and date:

Published: 03/10/2021

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

QUADROTOR CONTROL ON $SU(2) \times \mathbb{R}^3$ WITH SLAM INTEGRATION

A PREPRINT

Marcus Greiff, Patrik Persson, Zhiyong Sun, Karl Åström, and Anders Robertsson

October 5, 2021

ABSTRACT

We present a trajectory tracking controller for a quadrotor unmanned aerial vehicle (UAV) configured on $SU(2) \times \mathbb{R}^3$, and relate this result to a family of geometric tracking controllers on $SO(3) \times \mathbb{R}^3$. The theoretical results are complemented by simulation examples, and the controller is subsequently implemented in practice and integrated with a simultaneous localization and mapping (SLAM) system through an extended Kalman filter (EKF). This facilitates the operation of the UAV without external motion capture systems, and we demonstrate that the proposed control system can be used for inventorying tasks in a supermarket environment without external positioning systems.

1 INTRODUCTION

The UAV is quickly becoming a ubiquitous tool in modern society. There already exist commercially available products that are capable of autonomous flight through narrow forest paths while filming downhill mountain-bikers [1]. These robots have the potential to completely disrupt and revolutionize transportation and logistics [2]. Due to the high margin pressure associated with these sectors, initial applications are likely to be found in inventorying and data collection using small UAVs, where the cost can be kept low and the solutions can be implemented under contemporary legislation. This motivates the development of control systems similar to [1] that can operate without external motion capture systems and perform simple inventorying tasks.

For this purpose, we present a nonlinear tracking controller for the UAV dynamics, which yields uniform local exponential stability (ULES) properties on a smaller domain of attraction, and asymptotic attractiveness on a much larger domain. Similar methods have been developed in the quaternion formalism [3, 4, 5], but the approach taken in this paper extends the results in [6] using a special distance on $SU(2)$. This permits a stability proof analogous to that of the geometric tracking controller on $SO(3)$ in [7, 8], which allows further generalization to the robust and globally stabilizing controllers in [9, 10]. When compared to controllers based on the model predictive control (MPC) method, such as the learning-based MPC in [11] or the perception-aware MPC in [12], the geometric controllers are capable of similar feats of agility while requiring significantly less computational resources. This becomes particularly relevant in the context of small UAVs, which operate under computational constraints, where a large part of the processing power needs to be allocated to the on-board sensor fusion.

As the proposed controller relies on full-state information, a specialized system for simultaneous localization and mapping (SLAM) is implemented to generate real-time pose (position and attitude) estimates of the UAV based on video feed streamed from a camera mounted on the UAV. For this purpose, a solution is implemented with ORB features [13] and the pre-integration proposed in [14], permitting real-time estimation of the UAV state conditioned on the inertial measurement unit (IMU) measurements and pose estimates from the SLAM system. We emphasize that this solution enables safe autonomous flights without external motion capture systems, and demonstrate the theory with experiments using the Crazyflie 2.0 UAV [15] (see Fig. 1).

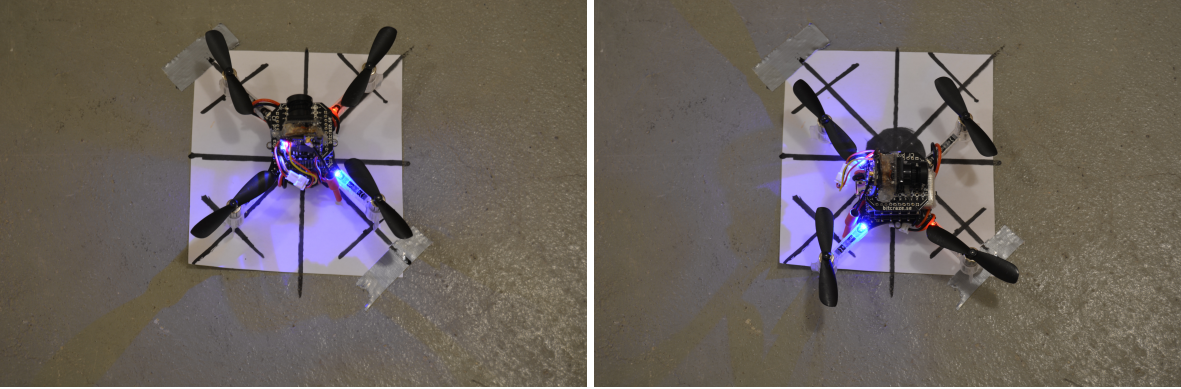


Figure 1: The Crazyflie 2.0 used in the experiments with a camera attached. *Left:* Initial configuration at $t = t_0$. *Right:* Terminal configuration at $t = t_f$.

1.1 Contributions

The contributions of this paper are threefold, and can be summarized as follows:

- A tracking geometric controller for UAV dynamics configured on $SU(2) \times \mathbb{R}^3$ is proposed, using the approach in [16] based on the attitude control result in [6]. The controller differs to that in [16] in several ways that have meaningful consequences, and its stability proof further motivates the use of the results presented in [5].
- A real-time compatible stack for SLAM is presented. The system tracks and matches ORB features [13] from a continuous video feed, utilizing the pre-integration proposed in [14] to leverage the sampled IMU-data.
- A demonstration of the controller in real-time is given, fusing estimates from the SLAM system and with other sensory data in an on-board extended Kalman filter (EKF), rendering the resulting control system capable of safely executing complex maneuvers that are relevant to supermarket inventorying.

1.2 Notation

Matrices and vectors are written in bold font, with entries of a vector \mathbf{u} as u_i , and the entries of a matrix \mathbf{A} are denoted by $[\mathbf{A}]_{ij}$. The smallest eigenvalue of a real symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is denoted by $\lambda_m(\mathbf{A})$, and its largest by $\lambda_M(\mathbf{A})$. The two-norm of $\mathbf{u} \in \mathbb{R}^n$ is denoted by $\|\mathbf{u}\| = \sqrt{\mathbf{u}^\top \mathbf{u}}$, and $\|\mathbf{u}\|_M = \sqrt{\mathbf{u}^\top \mathbf{M} \mathbf{u}}$ with a positive definite real matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$. The trace of $\mathbf{A} \in \mathbb{C}^{n \times n}$ is written $\text{Tr}(\mathbf{A}) = \sum_{i=1}^n A_{ii}$, and $\mathcal{U}(\Omega)$ denotes a uniform distribution over Ω , and implies sampling every element of Ω with equal probability. Let $\mathbf{S} : \mathbb{R}^3 \mapsto \mathbb{R}^{3 \times 3}$ such that for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$, where

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (1)$$

1.3 Structure

The mathematical preliminaries are given in Section 2, introducing a result on attitude control and identities pertaining to $SU(2)$. This is followed by a presentation of the problem formulation and the main result in Section 3, where attitude controller is used to derive a nonlinear tracking feedback law for a UAV configured on $SU(2) \times \mathbb{R}^3$. The SLAM system and its integration are discussed in Section 4, followed by simulation and experimental results in Section 5. Finally, the conclusion and outlook in Section 6 close the paper.

2 PRELIMINARIES

In the mathematical preliminaries, we start by defining the configuration manifolds, and relate elements of $SO(3)$ to elements $SU(2)$ through a carefully constructed embedding.

Definition 1. Let $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}$, with an associated Lie algebra $\mathfrak{so}(3) = \{\mathbf{L}_1 \omega_1 + \mathbf{L}_2 \omega_2 + \mathbf{L}_3 \omega_3 \in \mathbb{R}^{3 \times 3} \mid \boldsymbol{\omega} \in \mathbb{R}^3\}$ spanned by

$$\mathbf{L}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Definition 2. Let $SU(2) = \{\mathbf{X} \in \mathbb{C}^{2 \times 2} \mid \mathbf{X}^* \mathbf{X} = \mathbf{I}, \det(\mathbf{X}) = 1\}$, with an associated Lie algebra $\mathfrak{su}(2) = \{\mathbf{L}_1 \omega_1 + \mathbf{L}_2 \omega_2 + \mathbf{L}_3 \omega_3 \in \mathbb{C}^{2 \times 2} \mid \boldsymbol{\omega} \in \mathbb{R}^3\}$ spanned by

$$\mathbf{L}_1 = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{L}_3 = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}.$$

Here, we note that for any $\mathbf{X} \in SU(2)$, both \mathbf{X} and $-\mathbf{X}$ map to the same element on $SO(3)$. To see this, parametrize $SU(2)$ by a unit vector $\mathbf{q} = (q_1, q_2, q_3, q_4)^\top$, as

$$\mathbf{X} = \begin{pmatrix} q_1 + iq_4 & -q_3 + iq_2 \\ q_3 + iq_2 & q_1 - iq_4 \end{pmatrix} \in SU(2), \quad (2)$$

which encompasses all of $SU(2)$ by Definition 2. Furthermore, we embed elements $\mathbf{X} \in SU(2)$ into $\mathbf{R} \in SO(3)$ by

$$\mathbf{R} = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 - q_1 q_4) & 2(q_2 q_4 + q_1 q_3) \\ 2(q_2 q_3 + q_1 q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3 q_4 - q_1 q_2) \\ 2(q_2 q_4 - q_1 q_3) & 2(q_3 q_4 + q_1 q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \in SO(3). \quad (3)$$

Definition 3 (Lie Maps). Let G be any of the above defined Lie groups, with algebra \mathfrak{g} . The hat map is denoted $[\cdot]_G^\wedge : \mathbb{R}^3 \mapsto \mathfrak{g}$, the vee map is denoted $[\cdot]_G^\vee : \mathfrak{g} \mapsto \mathbb{R}^3$, and the associated exponential and logarithmic maps are denoted by $Exp_G : \mathfrak{g} \mapsto G$ and $Log_G : G \mapsto \mathfrak{g}$ respectively [17, 6].

To simplify any implementation of the controllers, the

Definition 4 (Hat and vee maps of $SO(3)$). From Definitions 1 and 3, it follows that if $\mathbf{K} = [\boldsymbol{\omega}]_{SO(3)}^\wedge \in \mathfrak{so}(3)$, then

$$[\boldsymbol{\omega}]_{SO(3)}^\wedge = \mathbf{S}(\boldsymbol{\omega}), \quad [\mathbf{K}]_{SO(3)}^\vee = \begin{bmatrix} [\mathbf{K}]_{3,2} \\ [\mathbf{K}]_{1,3} \\ [\mathbf{K}]_{2,1} \end{bmatrix}.$$

Definition 5 (Hat and vee maps of $SU(2)$). From Definitions 1 and 3, it follows that if $\mathbf{K} = [\boldsymbol{\omega}]_{SU(2)}^\wedge \in \mathfrak{su}(2)$, then

$$[\boldsymbol{\omega}]_{SU(2)}^\wedge = \begin{bmatrix} i\omega_3 & -\omega_2 + i\omega_1 \\ \omega_2 + i\omega_1 & -i\omega_3 \end{bmatrix}, \quad [\mathbf{K}]_{SU(2)}^\vee = \frac{1}{2} \begin{bmatrix} \Im([\mathbf{K}]_{1,2} + [\mathbf{K}]_{2,1}) \\ \Re([\mathbf{K}]_{2,1} - [\mathbf{K}]_{1,2}) \\ \Im([\mathbf{K}]_{1,1} - [\mathbf{K}]_{2,2}) \end{bmatrix}.$$

Remark 1. Here, we note that if $\mathbf{X} \in SU(2)$ and $\mathbf{R} \in SO(3)$ represent the same attitude on $SO(3)$, using the embedding in (2) and (3), then, for any $\boldsymbol{\omega} = \theta \mathbf{u} \in \mathbb{R}^3$,

$$Exp_{SO(3)}([\boldsymbol{\omega}]_{SO(3)}^\wedge) \in SO(3), \quad Exp_{SU(2)}([\boldsymbol{\omega}/2]_{SU(2)}^\wedge) \in SU(2),$$

both represent the same attitude on $SO(3)$, corresponding to a rotation of θ about a unit vector $\mathbf{u} \in \mathbb{R}^3$. For any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ with $\|\mathbf{a}\| = \|\mathbf{b}\|$, we have the rotational composition

$$\mathbf{a} = \mathbf{R}\mathbf{b}, \quad \mathbf{a} = [\mathbf{X}[\mathbf{b}]_{SU(2)}^\wedge \mathbf{X}^*]_{SU(2)}^\vee. \quad (4)$$

Definition 6. In the following, we define a global frame $\{\mathcal{G}\}$ spanned by three unit vectors \mathbf{e}_i with the i th element set to 1, and a body-fixed frame $\{\mathcal{B}\}$ spanned by three unit vectors \mathbf{b}_i , which are related to the global frame by

$$\mathbf{I} = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] = \mathbf{R}^\top [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3], \quad (5)$$

where \mathbf{R} rotates a vector from $\{\mathcal{B}\}$ to $\{\mathcal{G}\}$ (see Fig. 2).



Figure 2: *Left:* Depiction of the considered UAV geometry. *Right:* The laboratory environment with $\{G\}$.

In the following, we will at times refer to a *reference rotation*, with $\mathbf{R}_r \in SO(3)$ or $\mathbf{X}_r \in SU(2)$, with an associated set of body basis vectors \mathbf{b}_{ri} , and a *desired rotation* $\mathbf{R}_d \in SO(3)$ or $\mathbf{X}_d \in SU(2)$ with an associated set of body basis vectors \mathbf{b}_{di} (for $i = 1, 2, 3$). We also make frequent use of two closely related distances, here defined as follows.

Definition 7. Let $\Psi : SO(3) \times SO(3) \mapsto [0, 2]$, where

$$\Psi(\mathbf{R}_1, \mathbf{R}_2) = \frac{1}{2} \text{Tr}(\mathbf{I} - \mathbf{R}_1^\top \mathbf{R}_2). \quad (6)$$

Definition 8. Let $\Gamma : SU(2) \times SU(2) \mapsto [0, 2]$, where

$$\Gamma(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{2} \text{Tr}(\mathbf{I} - \mathbf{X}_1^* \mathbf{X}_2). \quad (7)$$

These distances facilitate elegant and powerful controller developments. The UAV dynamics are taken to be configured on $SU(2) \times \mathbb{R}^3$, with a state $(\mathbf{p}, \mathbf{v}, \mathbf{X}, \boldsymbol{\omega}) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SU(2) \times \mathbb{R}^3$, and are to be driven along a reference $(\mathbf{p}_r, \mathbf{v}_r, \mathbf{X}_r, \boldsymbol{\omega}_r) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SU(2) \times \mathbb{R}^3$, evolving in time by

$$\dot{\mathbf{p}}_r = \mathbf{v}_r \quad \dot{\mathbf{p}} = \mathbf{v} \quad (8a)$$

$$m\dot{\mathbf{v}}_r = f_r \mathbf{R}_r \mathbf{e}_3 - m g \mathbf{e}_3 \quad m\dot{\mathbf{v}} = f \mathbf{R} \mathbf{e}_3 - m g \mathbf{e}_3 \quad (8b)$$

$$\dot{\mathbf{X}}_r = \mathbf{X}_r [\boldsymbol{\omega}_r / 2]_{SU(2)}^\wedge \quad \dot{\mathbf{X}} = \mathbf{X} [\boldsymbol{\omega} / 2]_{SU(2)}^\wedge \quad (8c)$$

$$\mathbf{J} \dot{\boldsymbol{\omega}}_r = \mathbf{S}(\mathbf{J} \boldsymbol{\omega}_r) \boldsymbol{\omega}_r + \boldsymbol{\tau}_r \quad \mathbf{J} \dot{\boldsymbol{\omega}} = \mathbf{S}(\mathbf{J} \boldsymbol{\omega}) \boldsymbol{\omega} + \boldsymbol{\tau} \quad (8d)$$

where \mathbf{R} and \mathbf{R}_r are computed by (4) from \mathbf{X} and \mathbf{X}_r , respectively. In this notation, $\mathbf{p} \in \mathbb{R}^3$ [m] defines the position of the UAV in $\{G\}$; $\mathbf{v} \in \mathbb{R}^3$ [m/s] defines the velocity of the UAV in $\{B\}$; $\mathbf{X} \in SU(2)$ defines the attitude of the UAV; $\boldsymbol{\omega} \in \mathbb{R}^3$ [rad/s] denotes the usual attitude rates defined in $\{B\}$; $f > 0$ [N] defines the thrust generated by the rotors; $\boldsymbol{\tau}$ [N.m] denotes the torques defined in $\{B\}$ (see Fig. 2). The model is parameterized by a positive definite symmetric inertia matrix $\mathbf{J} \in \mathbb{R}^{3 \times 3}$, a positive mass $m > 0$ [kg], and a constant positive gravitation acceleration $g > 0$ [m/s²].

For the purposes of this paper, we recall the attitude controller in [6, Proposition 1], developed for the attitude subsystem characterized by (8c) and (8d), used to control $(\mathbf{X}, \boldsymbol{\omega})$ along the reference trajectory $(\mathbf{X}_r, \boldsymbol{\omega}_r)$.

Proposition 1. Let $\mathbf{X}_e = \mathbf{X}_r^* \mathbf{X} \in SU(2)$, and define

$$\mathbf{e}_X = \frac{1}{2} [\mathbf{X}_e - \text{Tr}(\mathbf{X}_e) \mathbf{I} / 2]_{SU(2)}^\vee \in \mathbb{R}^3, \quad \mathbf{e}_\omega = \boldsymbol{\omega} - [(\mathbf{X}_e)^* [\boldsymbol{\omega}_r]_{SU(2)}^\wedge (\mathbf{X}_e)]_{SU(2)}^\vee \in \mathbb{R}^3, \quad (10)$$

and let $\mathbf{z} = (\|\mathbf{e}_X\|, \|\mathbf{e}_\omega\|)^\top \in \mathbb{R}_{\geq 0}^2$. Take a set of gains $k_X, k_\omega, k_c > 0$ such that the matrices

$$\mathbf{W}^{aa} = \begin{bmatrix} \frac{k_c k_X}{\lambda_M(\mathbf{J})} & -\frac{k_c k_\omega}{2\lambda_m(\mathbf{J})} \\ -\frac{k_c k_\omega}{2\lambda_m(\mathbf{J})} & k_\omega - \frac{k_c}{4} \end{bmatrix} \succ \mathbf{0}, \quad \mathbf{M}_1^{aa} = \frac{1}{2} \begin{bmatrix} 4k_X & -k_c \\ -k_c & \lambda_m(\mathbf{J}) \end{bmatrix} \succ \mathbf{0}, \quad \mathbf{M}_2^{aa} = \frac{1}{2} \begin{bmatrix} \frac{8k_X}{2-\phi} & k_c \\ k_c & \lambda_M(\mathbf{J}) \end{bmatrix} \succ \mathbf{0}. \quad (11)$$

Then, for any initial error on the domain

$$D^a = \left\{ \begin{bmatrix} \mathbf{e}_X \\ \mathbf{e}_\omega \end{bmatrix} \in \mathbb{R}^6 \mid \begin{array}{l} \Gamma(\mathbf{X}_r(t_o), \mathbf{X}(t_o)) \leq \phi < 2, \\ \mathbf{z}(t_o)^\top \mathbf{M}_2^{aa} \mathbf{z}(t_o) \leq k_X \phi \end{array} \right\}, \quad (12)$$

driving the system (8d) with a full-state feedback

$$\boldsymbol{\tau} = -k_X \mathbf{e}_X - k_\omega \mathbf{e}_\omega - \mathbf{S}(\mathbf{J}\boldsymbol{\omega})\boldsymbol{\omega} + \mathbf{J}[-\mathbf{e}_\omega/2]_{\widehat{SU}(2)}^* \mathbf{X}_e^* [\boldsymbol{\omega}_r]_{\widehat{SU}(2)} \mathbf{X}_e + \mathbf{X}_e^* [\dot{\boldsymbol{\omega}}_r]_{\widehat{SU}(2)} \mathbf{X}_e + \mathbf{X}_e^* [\boldsymbol{\omega}_r]_{\widehat{SU}(2)} \mathbf{X}_e [\mathbf{e}_\omega/2]_{\widehat{SU}(2)}^{\vee} \mathbf{X}_e \quad (13a)$$

yields an equilibrium point $\mathbf{z} = \mathbf{0} \Rightarrow (\mathbf{e}_X, \mathbf{e}_\omega) = (\mathbf{0}, \mathbf{0}) \Rightarrow (\mathbf{X}, \boldsymbol{\omega}) = (\mathbf{X}_r, \boldsymbol{\omega}_r)$, which is UES on D^a .

Proof. The proof is given in [6, Proposition 1], and follows by the definition of a Lyapunov function candidate

$$\mathcal{V}^a = k_X \Gamma(\mathbf{X}_r, \mathbf{X}) + c_a \mathbf{e}_\omega \cdot \mathbf{e}_X + \frac{1}{2} \mathbf{e}_\omega \cdot \mathbf{J} \mathbf{e}_\omega, \quad (14)$$

where all solutions remain on D^a as $\mathcal{V}^a|_{c_a=0}$ is non-increasing on D^a , also ensuring that $\Gamma(\mathbf{X}_r, \mathbf{X}) < \phi$ for all $t \geq t_o$. As $\|\mathbf{z}\|_{\mathbf{M}_1^{aa}}^2 \leq \mathcal{V}^a \leq \|\mathbf{z}\|_{\mathbf{M}_2^{aa}}^2$ and $\dot{\mathcal{V}}^a \leq -\|\mathbf{z}\|_{\mathbf{W}^{aa}}^2$, a proof of UES on D^a follows by [18, Theorem 4.10]. \square

With these preliminaries, we proceed by posing and solving the control problem for the full UAV dynamics.

3 THE UAV CONTROL PROBLEM

We start by defining the control problem in Section 3.1. To solve this problem, intuition regarding the problem geometry and the controller design is given in Section 3.2 and 3.3, respectively. Based on these ideas, and using the geometric controller in [6] as a starting point, an analogous continuous feedback to that in [7] is derived for the system in (8) configured on $SU(2) \times \mathbb{R}^3$. This main theoretical result is given in Section 3.4.

3.1 Control Problem

Problem 1. Consider a system with a state $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \mathbf{X}, \boldsymbol{\omega}) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SU(2) \times \mathbb{R}^3$, with an associated reference trajectory $\mathbf{x}_r = (\mathbf{p}_r, \mathbf{v}_r, \mathbf{X}_r, \boldsymbol{\omega}_r) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SU(2) \times \mathbb{R}^3$, driven by $(f, \boldsymbol{\tau}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^3$ and $(f_r, \boldsymbol{\tau}_r) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^3$, respectively, evolving by the UAV dynamics in (8). Assume that the state \mathbf{x} is known and design a full state feedback law such that $\mathbf{x} \rightarrow \mathbf{x}_r$ as $t \rightarrow \infty$, and characterize the stability properties of the closed-loop system.

3.2 Geometric Intuition

To provide some intuition regarding the distance in (7) and aid the developments, consider two elements $\mathbf{X}_1, \mathbf{X}_2 \in SU(2)$. Take the conjugate product $\mathbf{X}_e = \mathbf{X}_1^* \mathbf{X}_2$ to be a rotation of θ about a unit vector $\mathbf{u} = (u_1, u_2, u_3)^\top$, as

$$\mathbf{X}_e \triangleq \begin{pmatrix} \cos(\frac{\theta}{2}) + iu_3 \sin(\frac{\theta}{2}) & -u_2 \sin(\frac{\theta}{2}) + iu_1 \sin(\frac{\theta}{2}) \\ u_2 \sin(\frac{\theta}{2}) + iu_1 \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) - iu_3 \sin(\frac{\theta}{2}) \end{pmatrix},$$

then $\Gamma(\mathbf{X}_1, \mathbf{X}_2) = 1 - \cos(\frac{\theta}{2})$, and $\forall \Gamma(\mathbf{X}_1, \mathbf{X}_2) \leq \phi < 2$,

$$\frac{1}{2} \sin^2(\frac{\theta}{2}) \leq \Gamma(\mathbf{X}_1, \mathbf{X}_2) \leq \frac{1}{2 - \phi} \sin^2(\frac{\theta}{2}). \quad (15)$$

for any ϕ upper bounding $\Gamma(\mathbf{X}_1, \mathbf{X}_2)$ (see Figure 3). This will be used in the stability analysis of the controllers on $SU(2)$. Specifically, if $\mathbf{e}_X = \frac{1}{2}[\mathbf{X}_e - \text{Tr}(\mathbf{X}_e)\mathbf{I}/2]_{\widehat{SU}(2)}^{\vee} = \frac{1}{2} \sin(\frac{\theta}{2}) \mathbf{u}$ with $\mathbf{X}_e = \mathbf{X}_1^* \mathbf{X}_2$, $\|\mathbf{e}_X\|^2 = \frac{1}{4} \sin^2(\frac{\theta}{2})$, then (15) becomes

$$2\|\mathbf{e}_X\|^2 \leq \Gamma(\mathbf{X}_1, \mathbf{X}_2) \leq \frac{4}{2 - \phi} \|\mathbf{e}_X\|^2. \quad (16)$$

This observation leads to several insightful geometric relationships with respect to the sine and cosine of the angle of the eigen-axis rotation between two vectors. For future reference, we first make some preliminary geometric observations with respect to the vectors $\mathbf{b}_{d3} = \mathbf{R}_d \mathbf{e}_3$ and $\mathbf{b}_3 = \mathbf{R} \mathbf{e}_3$, and their cosine angle $\mathbf{b}_{d3} \cdot \mathbf{b}_3 = \cos(\theta)$.

- Firstly, we note that

$$\Gamma(\mathbf{X}_d, \mathbf{X}) \leq \phi < 1 - 1/\sqrt{2} \Rightarrow \cos(\theta) > 0. \quad (17)$$

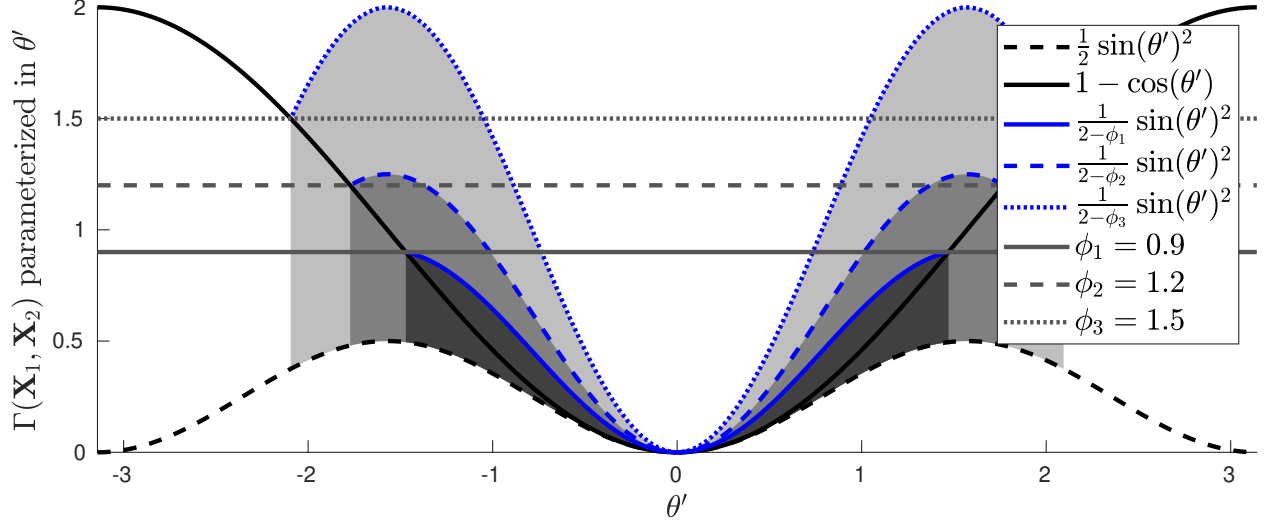


Figure 3: Illustration of the distance Γ with the element $\mathbf{X}_1, \mathbf{X}_2$ representing a rotation of θ about an arbitrary unit axis, with $\theta' = \theta/2$. Illustrates the upper and lower bounds in (15) for three different values of $\phi \in \{0.9, 1.2, 1.5\}$.

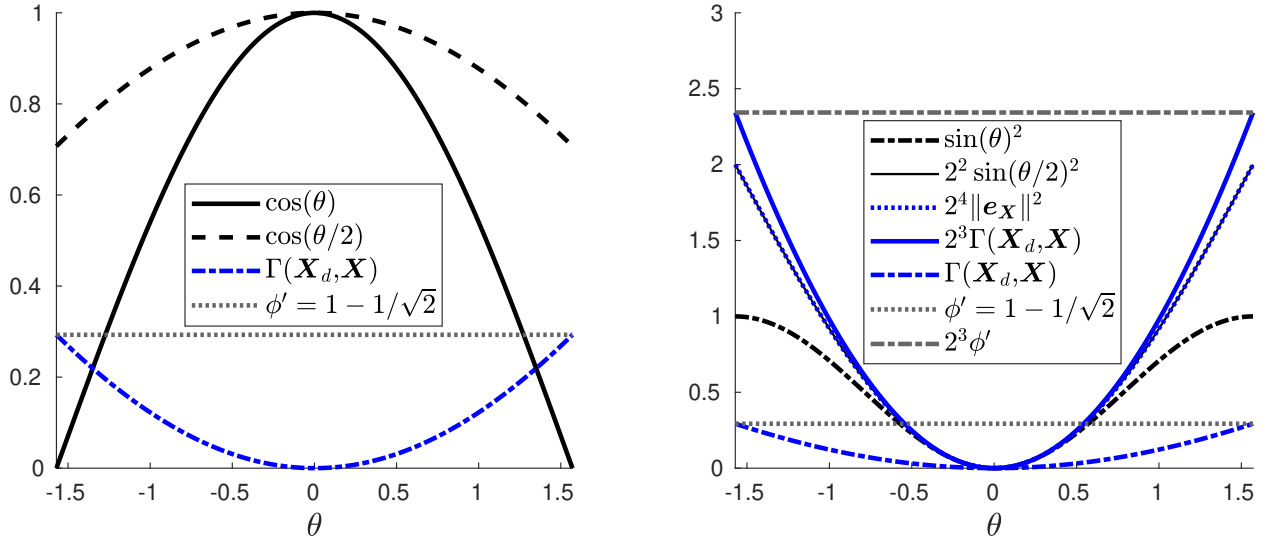


Figure 4: Illustration of the geometric relationships that facilitate the stability proof. *Left*: Expressions relating to the cosine of the rotation angle θ . *Right*: Expressions relating to the squared sine of the rotation angle θ .

- Secondly, as $\sin^2(\theta) \leq 4 \sin^2(\theta/2)$ for all $\cos(\theta) > 0$,

$$\sin^2(\theta) \leq 2^4 \|\mathbf{e}_X\|^2 \leq 2^3 \Gamma(\mathbf{X}_d, \mathbf{X}) \leq 2^3 \phi \triangleq \alpha^2. \quad (18)$$

These geometric relationships are illustrated in Figure 4, and from these observations, it is clear that for any

$$\Gamma(\mathbf{X}_d, \mathbf{X}) \leq \phi < 2^{-3} \Rightarrow \cos(\theta) > 0 \Rightarrow \alpha < 1 \quad (19)$$

Remark 2. The condition on ϕ may seem overly restrictive, but it is worth noting that if $\mathbf{X}_e \in SU(2)$ is parameterized in the more common ZYX Tait-Bryan rotation angles (pitch, roll, and yaw), as $\boldsymbol{\eta} \in \mathbb{R}^3$, then $\|\boldsymbol{\eta}\|^2 \approx 4\Psi(\mathbf{R}_d, \mathbf{R}) \approx 8\Gamma(\mathbf{X}_d, \mathbf{X})$. As such, this condition in (18) is more restrictive than $\Psi(\mathbf{R}_d, \mathbf{R}) \leq 1$ in [16] (by approximately a factor of four). But (18) still permits significant attitude errors when considered in the Tait-Bryan rotation angles.

3.3 Controller Intuition

A solution to Problem 1 when considered on $SO(3) \times \mathbb{R}^3$ is given in [7], with more detail in [16]. The developments here follow a similar idea, according to the observations in Section 3.2. Consider a set of translation control errors, defined as $e_p = p - p_r \in \mathbb{R}^3$ and $e_v = v - v_r \in \mathbb{R}^3$. Combined with the reference acceleration, these errors can be used to compute a desired force in the frame $\{\mathcal{G}\}$, as

$$\mathbf{f}_d = -k_p e_p - k_v e_v + m g e_3 + m \ddot{p}_r. \quad (20)$$

As the controlled system in (8) is only capable of generating forces along the \mathbf{b}_3 -direction, its attitude needs to be controlled to a desired attitude, $\mathbf{R}_d(t) \in SO(3)$, which transiently may differ from $\mathbf{R}_r(t) \in SO(3)$ when correcting for the errors in e_p and e_v . Consider expressing this attitude in terms of a set of desired body basis vectors $\mathbf{b}_{d,i} \in \mathbb{R}^3$, forming a desired body frame $\{\mathcal{B}_d\}$. It is clear that $\mathbf{b}_{d,3} = \mathbf{f}_d / \|\mathbf{f}_d\|$, but the final degree of freedom can be fixed in many ways. Three such examples are given below, where:

- (i) the vector $\mathbf{b}_{d,1}$ is provided explicitly;
- (ii) the vector $\mathbf{b}_{d,1}$ is computed from \mathbf{R}_r ;
- (iii) the vector $\mathbf{b}_{d,1}$ is defined with respect to $\mathbf{b}_{d,3}$.

In the case (ii), the desired body direction can be computed through a sequence of projections as outlined in [7], where

$$\mathbf{b}_{d,1} = -\frac{1}{\|\mathbf{b}_{d,3} \times \mathbf{b}_{r,1}\|} (\mathbf{b}_{d,3} \times (\mathbf{b}_{d,3} \times \mathbf{b}_{r,1})). \quad (21)$$

This permits a construction of the desired rotation, attitude rate and attitude rate time-derivative for case (i) and (ii), as

$$\mathbf{R}_d = [\mathbf{b}_{d,1} \quad (\mathbf{b}_{d,3} \times \mathbf{b}_{d,1}) \quad \mathbf{b}_{d,3}] \in SO(3), \quad (22a)$$

$$\boldsymbol{\omega}_d = [\mathbf{R}_d^\top \dot{\mathbf{R}}_d]_{SO(3)}^\vee \in \mathbb{R}^3, \quad (22b)$$

$$\dot{\boldsymbol{\omega}}_d = [\dot{\mathbf{R}}_d^\top \dot{\mathbf{R}}_d + \mathbf{R}_d^\top \ddot{\mathbf{R}}_d]_{SO(3)}^\vee \in \mathbb{R}^3. \quad (22c)$$

In the case (iii), the desired attitude can be formed by computing an angle $\beta \triangleq \text{atan2}(\sqrt{f_{d1}^2 + f_{d2}^2}, f_{d3})$, and defining

$$\mathbf{n} = (f_{d1}^2 + f_{d2}^2)^{-1/2} (-f_{d2}, f_{d1}, 0)^\top \quad (23a)$$

$$\mathbf{X}_A = \text{Exp}_{SU(2)}([\beta \mathbf{n} / 2]_{SU(2)}^\wedge) \quad (23b)$$

$$\mathbf{X}_B = \text{Exp}_{SU(2)}([\psi_r \mathbf{e}_3 / 2]_{SU(2)}^\wedge) \quad (23c)$$

where ψ_r parameterizes a desired rotation about the body $\mathbf{b}_{d,3}$ vector. With these definitions, the desired attitude can be defined as the composition $\mathbf{X}_d = \mathbf{X}_A \mathbf{X}_B$, computing the desired rates and accelerations through the inverse kinematics, similar to (22b) and (22c), or by numerical differentiation. If considering the continuous attitude controller in Proposition 1, this desired attitude needs to be chosen with care such that \mathbf{X}_d is continuous in time to avoid dynamical unwinding [19]. This can be done using the ideas in [20], or directly using the distance in Definition 8.

Remark 3. In a controller implementation running at a time-step of h [s], enforcing continuity in $\mathbf{X}_d(t)$ can be done at a time t by computing one of the two elements $\bar{\mathbf{X}}_d(t) \in SU(2)$ associated with $\mathbf{R}_d(t) \in SO(3)$, taking

$$\mathbf{X}_d(t) = \begin{cases} +\bar{\mathbf{X}}_d(t), & \text{if } \Gamma(\bar{\mathbf{X}}_d(t), \mathbf{X}_d(t-h)) < 1 \\ -\bar{\mathbf{X}}_d(t), & \text{otherwise} \end{cases}.$$

The computation of $\bar{\mathbf{X}}_d$ from \mathbf{R}_d can be done though (4).

3.4 Continuous feedback on $SU(2) \times \mathbb{R}^3$

With this intuition, the main result is stated as follows.

Proposition 2 (Geometric Control on $SU(2) \times \mathbb{R}^3$). Consider the dynamics (8) controlled by a feedback where:

- the torques, $\boldsymbol{\tau}$, are computed by the controller Proposition 1 implemented to track a trajectory $(\mathbf{X}_d, \boldsymbol{\omega}_d, \dot{\boldsymbol{\omega}}_d)$;
- the desired attitude reference trajectory is formed by (22), expanding \mathbf{R}_d or \mathbf{q}_d into $\mathbf{X}_d \in SU(2)$, and enforcing continuity of the desired reference on $SU(2)$;

- the actuating force is computed as $f = \mathbf{f}_d \cdot \mathbf{R}e_3$, with the desired force computed as described in (20).

Assume that, for all $t \geq t_o$:

(A1) the desired reference $(\mathbf{X}_d, \boldsymbol{\omega}_d, \dot{\boldsymbol{\omega}}_d)$ is well defined;

(A2) there exists a bound $\|mge_3 + m\ddot{\mathbf{p}}_r\| \leq B_f$;

(A3) and $\|\mathbf{e}_p(t_o)\| < B_p$ for a fixed $B_p > 0$;

(A4) the initial errors satisfy $\Gamma(\mathbf{X}_d(t_o), \mathbf{X}(t_o)) \leq \phi < 2^{-3}$;

(A5) the controller parameters $(k_p, k_v, k_X, k_\omega, c_a, c_p) \in \mathbb{R}_{>0}^6$ are chosen such that for $\alpha = 2\sqrt{2\phi}$, the matrices

$$\mathbf{M}_1^{pp} \triangleq \frac{1}{2} \begin{bmatrix} k_p & -c_p \\ \star & m \end{bmatrix}, \quad \mathbf{M}_2^{pp} \triangleq \frac{1}{2} \begin{bmatrix} k_p & c_p \\ \star & m \end{bmatrix}, \quad \mathbf{W}^{pp} \triangleq \begin{bmatrix} \frac{c_p k_p}{m}(1-\alpha) & -\frac{c_p k_v}{2m}(1+\alpha) \\ \star & k_v(1-\alpha) - c_p \end{bmatrix}, \quad (24)$$

are all positive definite, and there exist a matrix

$$\mathbf{W}^{pa} \triangleq 4 \begin{bmatrix} \frac{B_f c_p}{m} & 0 \\ B_f + k_p B_p & 0 \end{bmatrix}, \quad (25)$$

such that $B_z = 4\lambda_m(\mathbf{W}^{aa})\lambda_m(\mathbf{W}^{pp}) - \|\mathbf{W}^{pa}\|^2 > 0$.

Consider a domain

$$D = \left\{ \begin{bmatrix} \mathbf{e}_p(t_o) \\ \mathbf{e}_v(t_o) \\ \mathbf{e}_X(t_o) \\ \mathbf{e}_\omega(t_o) \end{bmatrix} \in \mathbb{R}^{12} \left| \begin{array}{l} \Gamma(\mathbf{X}_d(t_o), \mathbf{X}(t_o)) = \phi_o \leq \phi < 2^{-3}, \\ \|\mathbf{e}_\omega(t_o)\|^2 \leq \frac{2}{\lambda_M(\mathbf{J})} k_X (\phi - \phi_o), \\ \lambda_M(\mathbf{M}_2^{aa}) \|\mathbf{z}_a(t_o)\|^2 + \\ \lambda_M(\mathbf{M}_2^{pp}) \|\mathbf{z}_p(t_o)\|^2 \leq \frac{1}{2} k_p B_p^2 \end{array} \right. \right\}, \quad (26)$$

where $\mathbf{z}_p = (\|\mathbf{e}_p\|; \|\mathbf{e}_v\|) \in \mathbb{R}_{\geq 0}^2$, $\mathbf{z}_a = (\|\mathbf{e}_X\|; \|\mathbf{e}_\omega\|) \in \mathbb{R}_{\geq 0}^2$. Given the assumptions (A1)-(A5), the equilibrium point $(\mathbf{e}_p, \mathbf{e}_v, \mathbf{e}_X, \mathbf{e}_\omega) = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$ is UES on D .

Proof. The proof is given in the Appendix, with a sketch provided here. Similar to [7, Proposition 2], it follows by defining a Lyapunov function candidate

$$\bar{V} = \frac{1}{2} k_p \|\mathbf{e}_p\|^2 + \frac{1}{2} m \|\mathbf{e}_v\|^2 + c_p \mathbf{e}_p \cdot \mathbf{e}_v + k_X \Gamma(\mathbf{X}_d, \mathbf{X}) + c_a \mathbf{e}_X \cdot \mathbf{e}_\omega + \frac{1}{2} \mathbf{e}_\omega \cdot \mathbf{J} \mathbf{e}_\omega. \quad (27a)$$

Given the assumptions (A1)-(A5), it is shown all solutions initialized on D remain on this domain for all $t \geq t_o$. Furthermore, it is shown \bar{V} is continuously differentiable, and there exist constants $c_1, c_2, c_3 > 0$ expressed in the matrices in (24) and (25), such that

$$c_1 \|\bar{\mathbf{z}}\|^2 \leq \bar{V} \leq c_2 \|\bar{\mathbf{z}}\|^2, \quad (d/dt)\bar{V} \leq -c_3 \|\bar{\mathbf{z}}\|^2, \quad (28)$$

where $\bar{\mathbf{z}} = (\|\mathbf{e}_p\|; \|\mathbf{e}_v\|; \|\mathbf{e}_X\|; \|\mathbf{e}_\omega\|)$. This holds for all solutions of the error dynamics on D . Applying [18, Theorem 10] shows UES of $\bar{\mathbf{z}} = \mathbf{0}$ on D . \square

3.5 Comments on the Assumptions

3.5.1 Assumption (A1)

The assumption is generally difficult to guarantee, as the denominators in (21) and (23a) depend on the control errors, and are defined with respect to the reference trajectory and not the desired reference trajectory. There may exist solutions both initially and transiently where these are ill defined. However, as the full-state information of the UAV is available, such cases can easily be detected and handled in the controller implementation (see, e.g., [21]).

3.5.2 Assumption (A4)

It may seem as though Assumption (A4) is restrictive, as it only permits small attitude errors. It can be relaxed slightly; given the characterization of the domain of exponential convergence in Proposition 1, the following holds.

Proposition 3. Consider the system in (8) in closed-loop feedback with Proposition 2, but instead of Assumption (A4), assume that the initial errors satisfy

$$\Gamma(\mathbf{X}_d(t_o), \mathbf{X}(t_o)) \leq \phi < 2, \quad (29a)$$

$$\|e_\omega(t_o)\|^2 \leq \frac{2}{\lambda_M(\mathbf{J})} k_X (\phi - \Gamma(\mathbf{X}_d(t_o), \mathbf{X}(t_o))). \quad (29b)$$

Under these conditions, the origin $(e_p, e_v, e_x, e_\omega) = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$ is asymptotically attractive.

Proof. This becomes completely analogous to the proof in [7, Appendix E], therefore omitted for brevity. It follows by showing boundedness of solutions on $t \in [t_o, t^*]$, before the errors approach D as defined in (26) at a finite time t^* , after which the errors decay exponentially to the origin. \square

It is worth noting that the UAV system with the attitude controller in Proposition 1 can be shown to be almost globally asymptotically stable (AGAS), in the sense that all initial conditions converge to a set $\Gamma(\mathbf{X}_d(t_o), \mathbf{X}(t_o)) \in \{0, 2\}$ with $e_\omega = \mathbf{0}$, corresponding to a stable point interior of (29), or an unstable point on the boundary of the domain of exponential attraction in (29). As such, the solutions associated with *almost all* initial conditions asymptotically converge to (29), with subsequent convergence of the errors to D in (26).

3.5.3 Assumption (A5)

As pointed out in [6], for small c_a , the matrices $M_1^{aa}, M_2^{aa}, \mathbf{W}^{aa}$ are positive definite. Similarly, for sufficiently small c_p , the matrices $M_1^{pp}, M_2^{pp}, \mathbf{W}^{pp}$ are positive definite. Specifically, sufficient conditions are

$$c_a < \min \left\{ 4k_\omega, \frac{4k_\omega k_X \lambda_m(\mathbf{J})^2}{\lambda_M(\mathbf{J})k_w^2 + \lambda_m(\mathbf{J})^2 k_X}, 2\sqrt{k_X \lambda_m(\mathbf{J})} \right\}$$

$$c_p < \min \left\{ k_v(1 - \alpha), \frac{4mk_p k_v (1 - \alpha)^2}{k_v^2(1 + \alpha)^2 + 4mk_p(1 - \alpha)}, \sqrt{k_x m} \right\}$$

Similarly, it is clear that the last condition in Assumption (A5) can be satisfied by decreasing ϕ and α and/or increasing the tuning parameters k_X and k_ω in relation to the B_f and B_p . It should also be noted that it can be replaced by a less restrictive condition $\mathbf{W}^{pp} - \mathbf{W}^{pa}(\mathbf{W}^{aa})^{-1}(\mathbf{W}^{pa})^\top \succ \mathbf{0}$.

4 THE ESTIMATION PROBLEM

In this section, we present a system for simultaneously localization and mapping (SLAM), which is used to generate real-time pose estimates of the UAV from a monocular video feed. The system is feature based, relying on extracted ORB features [13] for tracking and matching. Additionally it utilizes accelerometer data and gyroscopic data from an inertial measurement unit (IMU) to constrain the scale and two rotational components, the pitch and the roll. These are otherwise ambiguous if only images are used. This is important since the controller assumes positions in meters and rotations relative to the gravity direction. The system uses the method proposed in [14] to accumulate IMU data into so-called deltas, containing information about the metric relative transformation between consecutive frames. These deltas are then used together with feature matches between images in a large non-linear optimization problem called bundle adjustment [22], to solve for the camera poses and three-dimensional structure up to a metric solution in $\{\mathcal{G}\}$.

The SLAM system is divided into modules, each performing a specific task concurrently with the others and communicates with the other modules using message passing. The main modules in the system are described in order:

- 1 The initialization module (see Section 4.1);
- 2 The tracking module (see Section 4.2);
- 3 The re-localization module (see Section 4.3);
- 4 The triangulation module (see Section 4.4);
- 5 The mapping module (see Section 4.5).

Furthermore, the integration of the SLAM system with the EKF on the Crazyflie is described in Section 4.6.

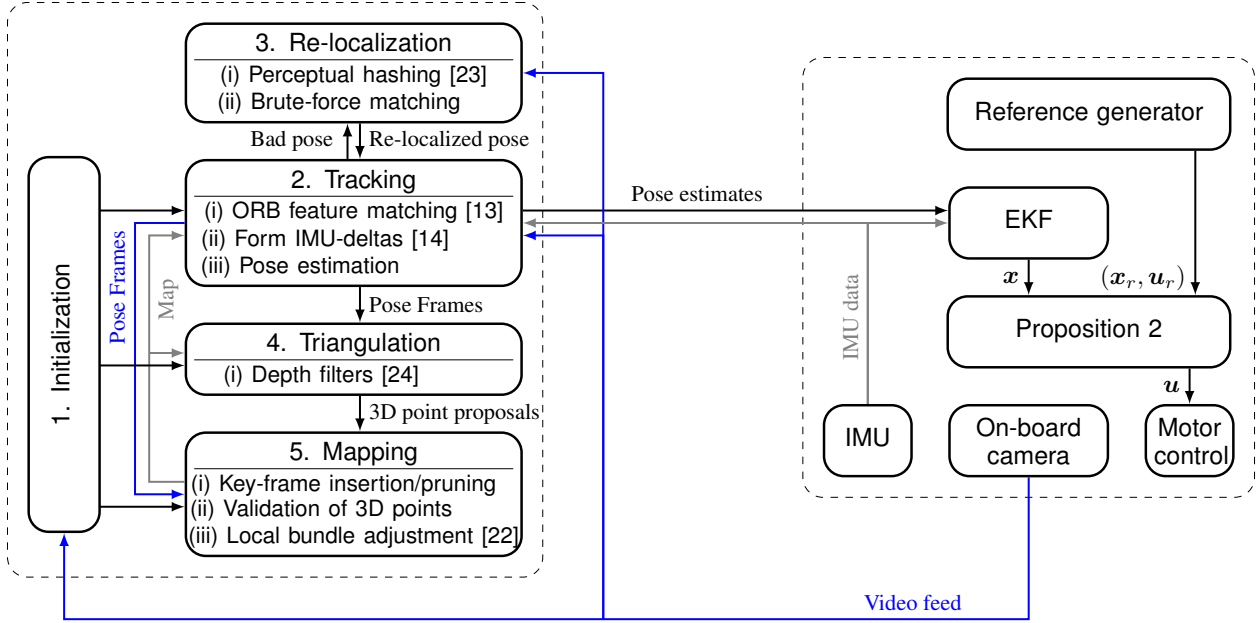


Figure 5: Holistic view of the control system modules with the main information flow. The SLAM system (left) is run on an external computer, and the modules implemented in the Crazyflie firmware (right) are run on its ARM processor. The communication between the dashed boxes is facilitated by two independent radios.

4.1 Initialization

The system has three phases: the initialization phase, the non-metric phase, and the metric phase (the metric phase being the operational phase). In the first phase an initial two-frame solution is found by selecting a reference frame and then as new frames arrive, estimating an essential matrix between the frames using the five point solver in [25] and the RANSAC framework in [26]. From the essential matrix the pose and 3D structure is extracted, and if the set of 3D points that satisfy a re-projection threshold is sufficiently large and the median depth is sufficiently low (indicating adequate parallax), the solution is accepted and is then sent to the mapping module. This module in turn bundles the solution and sends it to the tracking module. If the initialization fails at any step the solution is rejected and the next frame is tried. If it fails too many times a new reference frame is selected.

This moves the system to the non-metric phase where the solution is defined up to similarity transformation, and no IMU data is used to constrain the scale and rotation of the system. Once the system has collected five keyframes (to be defined in Section 4.5), the solution is upgraded to a metric solution using the method in [27], by solving for the gravity direction and scale. Now the system enters the operational phase, where the UAV can use the estimated pose.

4.2 Tracking

The tracking module tracks the pose of consecutive frames relative to the current map. It uses the previous frame (and the IMU data when in the metric phase), to generate a proposition of the pose of the current frame. By using the 3D points seen in the previous frame, a new set of potentially visible 3D points are selected from the map that are co-visible with the previous points. These are projected into the current frame to facilitate a guided search for feature matches. The pose is then optimized using the matched features and in the case of metric phase, together with the past 3 poses and their corresponding IMU deltas. This process is repeated with the optimized pose and a smaller search window in the guided search. Next, matches with large re-projection error are discarded, and if the number of matches are sufficiently large and the pose is consistent with the IMU data, the pose is sent to the other modules. If the tracking fails, a re-localization request is sent to the re-localization module, which then returns with the pose of the current frame if possible.

4.3 Re-localization

The re-localization module uses perceptual hashing in [23] to turn images into hashes, where similar images have similar hashes. When performing re-localization of a query frame, the frame is converted to a hash. The hash can then

efficiently be compared to the hashes of the keyframes in the map to find frames that have observed the same view. The most similar frames are then selected as candidates. For each such frame, the observed 3D points are matched to the query frame features using brute-force matching together with a three point pose solver in [28] and RANSAC in [26]. Any camera pose with a sufficient amount of inlier matches is then sent back to the tracking module as a candidate pose.

4.4 Triangulation

To triangulate new 3D points and extend the current map, a two-step approach is used. Firstly, for each keyframe, the inverse depth filters in [24] are used to features that have not already been associated with 3D points. The filter maintains a Gaussian distribution over inverse depth of the 3D point from the keyframe’s point of view, as well as a beta distribution over measurement update inlier ratio. For each new pose frame sent from the tracker, the co-visible filters are updated. If the inlier ratio of a filter becomes lower than a threshold, it is restarted. If the uncertainty of the inverse depth becomes sufficiently low, the inverse depth is considered to have converged and is sent from the triangulation module to the mapping module. In the next step, the inverse depth is converted to a 3D point distribution using the inverse depth uncertainty from the filter and an assumed pixel noise of one pixel. This distribution is then projected into co-visible keyframes and features close to the projection, with a sufficiently low descriptor error, are considered to be potential matches. In the next step, a two-point DLT based triangulation RANSAC procedure in [29] is used to find the 3D point with the largest inlier set. If this 3D point has been seen by four keyframes, then it is accepted and is given a life of 5. Each time the 3D point is seen by the tracker, the life is increased. Each time it is predicted but not seen, its life is decreased. If the life reaches zero, the 3D point is removed.

4.5 Mapping

The mapping module is responsible of updating, optimizing and pruning the map, as well as sending the changes to the rest of the modules. Using all of the frames in the video would quickly make the optimization problem prohibitively large and additionally many of the frames would be near identical and unable to provide much information. Therefore, a small subset is selected, here called keyframes, that accurately represent the solution. The map object contains the 3D points, the IMU-deltas, and the keyframes.

Keyframes are added in real-time as the map expands, and must satisfy certain conditions to be added: (i) compared to the closest keyframe, the camera must have moved at least 2.5% of the current mean depth; (ii) the mutual overlap with the most overlapping keyframe must be less than 90 %, or the average uncertainty of the projection of the 3D points must be higher than four pixels; and (iii) the number of tracked points must be larger than ten. This leads to a generous keyframe insertion policy, quickly expanding the 3D point set, but introducing redundancy in the keyframes.

A redundant keyframe can be characterized as observing the same 3D points as many of the other keyframes. To detect and remove redundant keyframes, we keep track of the amount of 3D points each keyframe observes that have also been observed by at least seven other keyframes. If the fraction of these 3D points is higher than 90%, the keyframe is marked as redundant. From the set of redundant keyframes, the frame that is the closest to another keyframe is removed.

Every time a new keyframe is added, a local bundle adjustment is performed where keyframes and 3D points that are co-visible with the new frame are adjusted and the remaining are kept fixed. This is done to prevent the map from diverging, and only a local portion of the map is optimized to keep the computational time bounded. Whenever the map has been updated, either by adding a new keyframe, by adding a new 3D point, or by performing a pruning, the difference between the previous map and the updated map is extracted and is then sent to the rest of the modules.

4.6 Integration

To fuse SLAM estimates with the IMU-data and generate a full state estimate of the UAV given the dynamics in (8), we consider the IMU-driven multiplicative extended Kalman filter (EKF) proposed in [30], with a first order attitude reset in [31]. The filter assumes a non-linear UAV model corresponding to the dynamics in (8), but with the velocities \mathbf{v} expressed in $\{\mathcal{B}\}$, and the attitude parameterized as a first-order attitude error, $\boldsymbol{\delta} \in \mathbb{R}^3$. This attitude error relates to the estimate $\hat{\mathbf{R}}(t) \in \text{SO}(3)$ as $\hat{\mathbf{R}}(t) = \hat{\mathbf{R}}(t_k)(\mathbf{I} + \mathbf{S}(\boldsymbol{\delta}(t)))$, and is reset to zero when it exceeds a predefined threshold with t_k denoting the most recent reset. As such, the state of the EKF is given by $\mathbf{z} = (\mathbf{p}; \mathbf{v}; \boldsymbol{\delta}) \in \mathbb{R}^9$, and the dynamics in (8) are expressed in \mathbf{z} . This estimate is subsequently externalized into the full state of the UAV, $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \mathbf{X}, \boldsymbol{\omega})$, where the attitude rates are computed by averaging the gyroscopic measurements between each EKF prediction. In the experiments, a scalar update version of the filter is used (see, e.g., [32, Chapter 6]), incorporating the measurements consecutively as they arrive. For additional details on the filter implementation, refer to [30, 31, 33].

The IMU-data from the Crazyflie are transferred to a PC using a version of the robot operating system (ROS) driver in [34], and the video is streamed over a different radio and processed directly by the SLAM system. This stack subsequently outputs an estimate of the UAV pose at 50 [Hz], which is communicated back to the UAV via radio through the ROS driver. The positional part of the estimate is queued into the EKF, which fuses the positional information with the IMU measurements at a rate of 100 [Hz]. The reference trajectory is also communicated to the UAV via ROS as a set of splines in the flat output space of the UAV, and the expansion of this trajectory into the states of the reference dynamics in (8) and the controller in Proposition 2, both run at a rate of 500 [Hz] (see Figure 5).

5 NUMERICAL EXAMPLES AND EXPERIMENTS

In this section, we start by giving a simulation example in Section 5.1 demonstrating the tracking properties of the controller in Proposition 2 for a circular maneuver with large initial errors. We then present an experiment in Section 5.2 where the controller is integrated with the SLAM system, while also removing the desired attitude accelerations (by letting $\dot{\omega}_d \triangleq 0$) to avoid differentiating the control errors.

5.1 Simulation Example

As the proposed controller is asymptotically attractive and ULES for all parameters $m > 0$, $\mathbf{J} = \mathbf{J}^\top \succ \mathbf{0}$ and all initial errors, we take $m = 0.1$ with $g = 10$, and sample a random inertia matrix (here chosen such that $\lambda_m(\mathbf{J}) = 0.05$ and $\lambda_M(\mathbf{J}) = 0.1$). The initial conditions of the system are sampled from $\mathbf{v}(t_o), \boldsymbol{\omega}(t_o) \sim \mathcal{N}(\mathbf{0}, 5\mathbf{I})$, with $\mathbf{p}(t_o) \sim \mathcal{N}((0, 0, -2)^\top, \mathbf{I})$, and $\mathbf{R}(t_o) \sim \mathcal{U}(\text{SO}(3))$. From this random initial state, the system is controlled along

$$\mathbf{p}_r(t) = (3 \sin(t), 3 \cos(t), 0)^\top, \quad \mathbf{b}_{1r}(t) = \mathbf{v}_r(t) / \|\mathbf{v}_r(t)\|.$$

This trajectory can be expanded into the full state trajectory of the UAV in (8) using the property of differential flatness (see, e.g., [32, Chapter 3]). In one particular realization,

$$\mathbf{R}(t_o) = \begin{bmatrix} 0.51 & -0.05 & -0.86 \\ -0.78 & 0.41 & -0.48 \\ 0.37 & 0.91 & 0.17 \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0.08 & 0.01 & 0.02 \\ 0.01 & 0.07 & 0.01 \\ 0.02 & 0.01 & 0.07 \end{bmatrix}, \quad (30)$$

$$\mathbf{p}(t_o) = \begin{bmatrix} 0.08 \\ -0.16 \\ -1.63 \end{bmatrix}, \quad \mathbf{v}(t_o) = \begin{bmatrix} -0.59 \\ 0.76 \\ -0.95 \end{bmatrix}, \quad \boldsymbol{\omega}(t_o) = \begin{bmatrix} -1.81 \\ 1.80 \\ 2.81 \end{bmatrix}. \quad (31)$$

For this realization of the UAV parameters and the initial errors, the control signals, reference trajectory and system response are depicted in Fig. 6. Here, we note that the Lyapunov function in (27a) quickly decays to a small value (here shown in the 10-logarithm), and that it satisfies the associated quadratic bounds at all times. This holds despite the system being initialized outside of D . As such, we here rely on the (almost global) asymptotic attractiveness properties in Proposition (3) before reaching the domain of exponential attraction. The system configurations are depicted in time in Fig. 7, with color coding of $\{\mathcal{B}\}$ corresponding to Fig. 2. Given that a new set of initial conditions and system parameters can be sampled, similar convergence properties were verified in a total of 10^3 realizations of the parameters and initial errors in (31).

5.2 Real-Time Example

In the second example, an inventorying experiment is conducted in real-time with a Crazyflie 2.0 [15] with respect to a set of shelves in the corner of a room (see Fig. 2). The SLAM system is integrated with the stock multiplicative EKF of the Crazyflie [33] as described in Section 4, using the positional estimates from the SLAM system in combination with the IMU-measurements to generate a full-state estimate. These estimates are subsequently used by the controller in Proposition 2 to actuate the UAV, but employing the discontinuous version of the attitude controller in [6].

The shelf geometry is assumed to be known, and a reference trajectory is planned consisting of linear splines such that the UAV traverses each segment at a velocity of $\|\mathbf{v}_r(t)\| = 1$ [m/s] approximately 0.4 [m] from the shelves. In order to keep the shelves within camera view, this implies performing a turn at $t \in [24.6, 26.3]$, also defined by linear splines. As such, the reference trajectory cannot be followed perfectly at the spline end-points, where the velocities are discontinuous, resulting in slight transients in the errors. The tracking errors are shown in Fig. 8, with the expanded reference trajectory and positional estimates logged from the UAV, and the attitude error computed between the estimated rotation in the EKF and the rotation in the SLAM system (which is not incorporated in the EKF). To highlight that the attitude error does not correspond to the distance between the estimate of the EKF and the desired

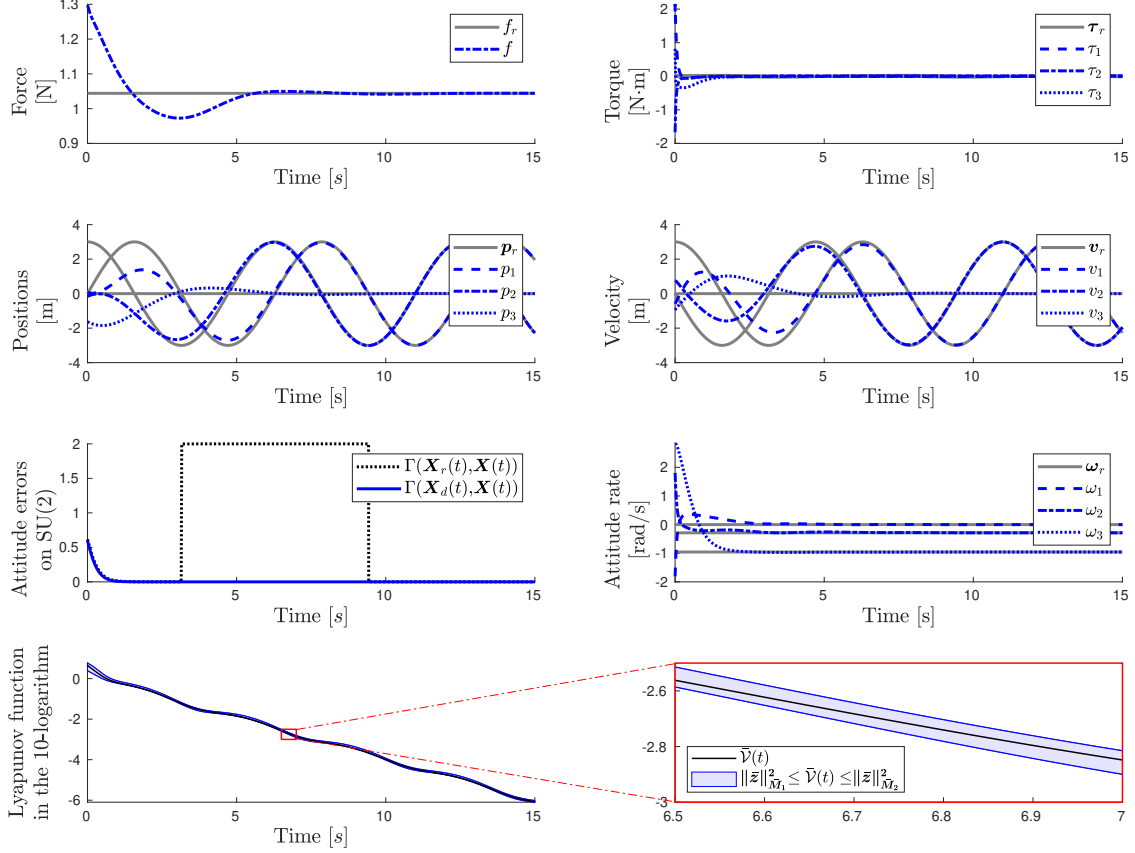


Figure 6: System response and control signals in time with references in gray and signals of the controlled system in blue. *Top, left*: Reference- and controlled force. *Top, right*: Reference- and controlled torque. *Top center, left*: Reference- and controlled position. *Top center, right*: Reference- and controlled velocity. *Bottom center, left*: Distance to the reference attitude (black) and desired attitude (blue). *Bottom center, right*: Reference- and controlled attitude rate. *Bottom*: The Lyapunov function depicted with the associated quadratic bounds in blue, depicted over $t \in [0, 15]$ to the left, with a zoom indicated in red on $t \in [6.5, 7]$ to the right.

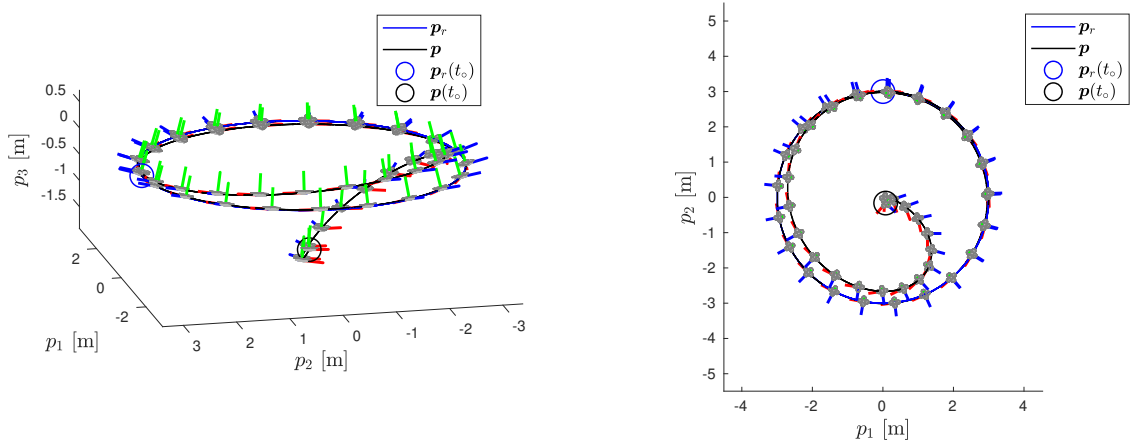


Figure 7: Configurations of the UAV $(\mathbf{p}(t), \mathbf{X}(t)) \in \mathbb{R}^3 \times SU(2)$ when controlled using Proposition 2.

attitude (i.e., the control error $\mathbf{X}_d^* \mathbf{X}$ in the UAV), the attitude error is depicted in the Ψ -distance, with \mathbf{R}_r sampled from the UAV reference trajectory and \mathbf{R} estimated from the SLAM system.

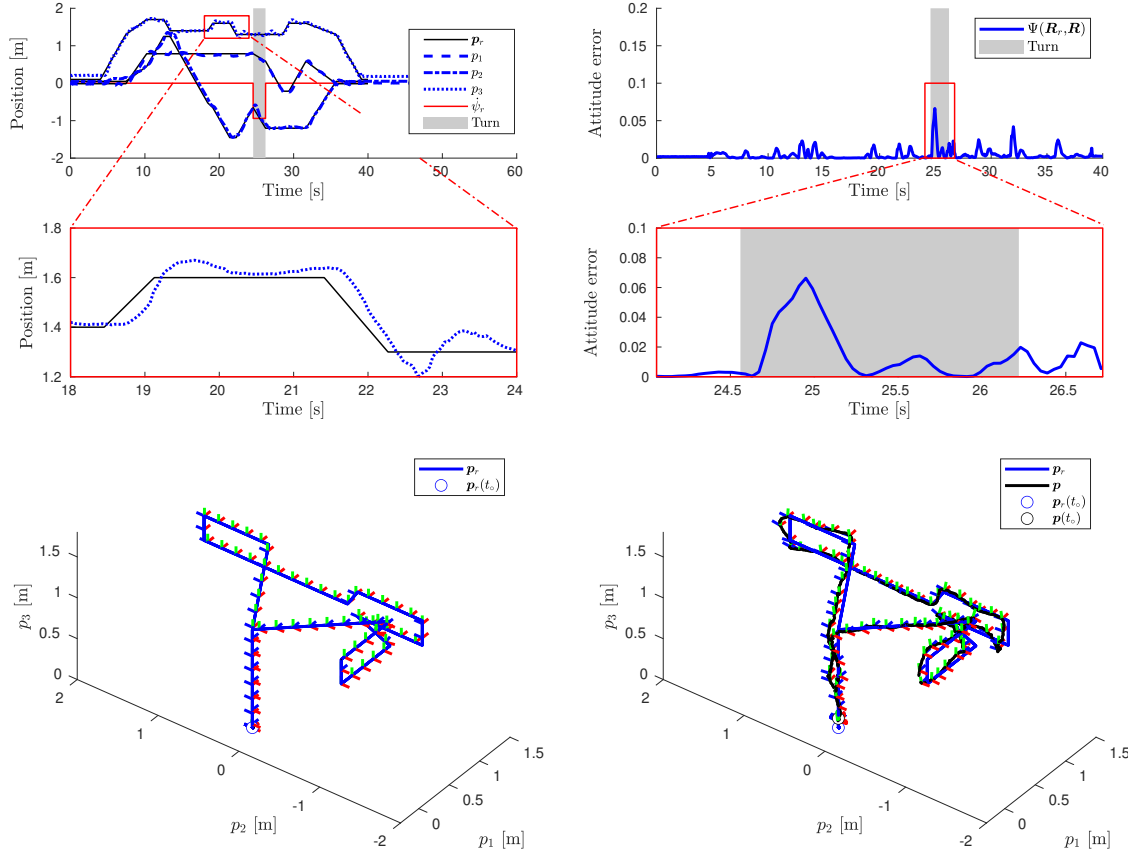


Figure 8: *Top, left*: The positional reference p_r (black) and the response p (blue), along with the time-derivative of the yaw angle reference (red) indicating the time-interval during which the system is turning (gray). *Top, right*: The tracking attitude error in the distance on $SO(3)$, slightly increasing during the turn, but otherwise small. *Center, left*: Zoom on the positional response in the elevation just before the turn, showing slight delays in the response and an overshoot. *Center, right*: Zoom on the attitude error during the turn. *Bottom, left*: Reference configurations trajectory in \mathbb{R}^3 in time. *Bottom, right*: System response in \mathbb{R}^3 in time, showing the reference trajectory, positional response, and measured rotation.

The UAV successfully scans the shelves in rapid succession, following the reference trajectory down to the expected tracking errors induced by a lack of continuity when switching between splines. We also note that the attitude error is relatively small throughout the experiment, with slight increases when switching between splines, and a larger error during the start of the turning maneuver. Again, we emphasize that this is the attitude error between the reference trajectory and the rotation estimated in the SLAM system, which is not explicitly used in the controller. Finally, to get a sense of the accuracy of the slam system, the initial and terminal configuration of the UAV is depicted in Fig. 1. This demonstrates that without any external motion capture, the controlled system navigates back to a point that differs from the initial position by a few centimeters, despite facing a different wall while landing. Supporting videos of the simulation as well as the experiment are published in [35].

6 CONCLUSIONS

In this paper, we have presented a geometric tracking controller for quadrotor UAVs configured on $SU(2) \times \mathbb{R}^3$, that is analogous to the geometric tracking controller on $SO(3) \times \mathbb{R}^3$ in [7] – yet distinctly different in ways that have meaningful consequences. In addition, a SLAM system was implemented based on ORB features to process monocular video to a sequence of a pose estimates that were subsequently fused with IMU-data in an on-board EKF. The controller in Proposition 2 was demonstrated in simulation, before being applied in practice to a real-time inventorying scenario. We emphasize that the proposed control system is capable of actuating the UAV along the desired reference trajectory without any external motion capture system. As such, a UAV equipped with this control system is easily deployed and is a low-cost alternative for supermarket inventorying.

References

- [1] Skydio, “Skydio product homepage,” last accessed at 2021-09-01. [Online]. Available: <https://www.skydio.com/>
- [2] T. M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, and P. Fraga-Lamas, “Towards an autonomous industry 4.0 warehouse: A UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management,” *Sensors*, vol. 19, no. 10, p. 2394, 2019.
- [3] E. Fresk and G. Nikolakopoulos, “Full quaternion based attitude control for a quadrotor,” in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 3864–3869.
- [4] D. Brescianini, M. Hehn, and R. D’Andrea, “Nonlinear quadrocopter attitude control: Technical report,” ETH Zurich, Tech. Rep., 2013.
- [5] D. Brescianini and R. D’Andrea, “Tilt-prioritized quadrocopter attitude control,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 376–387, 2018.
- [6] M. Greiff, Z. Sun, and A. Robertsson, “Attitude Control on $SU(2)$: Stability, Robustness, and Similarities,” *IEEE Control Systems Letters*, vol. 6, pp. 73–78, 2021.
- [7] T. Lee, M. Leok, and N. H. McClamroch, “Geometric tracking control of a quadrotor UAV on $SE(3)$,” in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5420–5425.
- [8] F. Goodarzi, D. Lee, and T. Lee, “Geometric nonlinear PID control of a quadrotor UAV on $SE(3)$,” *2013 European control conference (ECC)*, pp. 3845–3850, 2013.
- [9] T. Lee, M. Leok, and N. H. McClamroch, “Nonlinear robust tracking control of a quadrotor UAV on $SE(3)$,” *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2013.
- [10] T. Lee, “Global Exponential Attitude Tracking Controls on $SO(3)$,” *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2837–2842, 2015.
- [11] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, “Deep Drone Acrobatics,” 2020. [Online]. Available: arxiv.org/abs/2006.05768
- [12] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, “PAMPC: Perception-aware model predictive control for quadrotors,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.
- [13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [14] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual-inertial odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [15] Bitcraze, “Crazyflie 2.0,” 2021, last accessed: 10-4-2021. [Online]. Available: www.bitcraze.io/products/old-products/crazyflie-2-0
- [16] T. Lee, M. Leok, and N. H. McClamroch, “Control of complex maneuvers for a quadrotor UAV using geometric methods on $SE(3)$,” 2011. [Online]. Available: arxiv.org/abs/1003.2005
- [17] B. Hall, *Lie groups, Lie algebras, and representations: an elementary introduction*. Springer International Publishing, Switzerland, 2015, vol. 222.
- [18] H. Khalil, *Nonlinear systems*, 3rd ed. Prentice hall Upper Saddle River, New Jersey, USA, 2002.
- [19] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, “On quaternion-based attitude control and the unwinding phenomenon,” in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 299–304.
- [20] ———, “On path-lifting mechanisms and unwinding in quaternion-based attitude control,” *IEEE Transactions on Automatic Control*, vol. 58, no. 5, pp. 1179–1191, 2012.
- [21] M. Greiff and A. Robertsson, “Incremental Reference Generation for Nonsingular Control on $SE(3)$,” in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 132–137.
- [22] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–3288.
- [23] C. Zauner, “Implementation and benchmarking of perceptual image hash functions,” Ph.D. dissertation, 2010, ISBN: 1446144429.
- [24] G. Vogiatzis and C. Hernández, “Video-based, real-time multi-view stereo,” *Image and Vision Computing*, vol. 29, no. 7, pp. 434–441, 2011.

- [25] D. Nister, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [26] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the Association for Computing Machinery (ACM)*, vol. 24, pp. 381–395, 1981.
- [27] R. Mur-Artal and J. D. Tardós, “Visual-Inertial Monocular SLAM With Map Reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [28] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, “Complete solution classification for the perspective-three-point problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.
- [29] R. I. Hartley and P. Sturm, “Triangulation,” *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.
- [30] M. W. Mueller, M. Hamer, and R. D’Andrea, “Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1730–1736.
- [31] M. W. Mueller, M. Hehn, and R. D’Andrea, “Covariance Correction Step for Kalman Filtering with an Attitude,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2301–2306, 2017.
- [32] M. Greiff, “Modelling and control of the Crazyflie quadrotor for aggressive and autonomous flight by optical flow driven state estimation,” Master’s thesis, Lund University, 2017.
- [33] Bitcraze, “MEKF in the Crazyflie,” 2021, last accessed: 10-4-2021. [Online]. Available: github.com/bitcraze/crazyflie-firmware/blob/master/src/modules/src/kalman_core/kalman_core.c
- [34] W. Hönig and N. Ayanian, *Flying Multiple UAVs Using ROS*. Springer International Publishing, 2017, pp. 83–118.
- [35] M. Greiff, “Inventorying with the crazyflie 2.0,” last accessed at 2021-10-03. [Online]. Available: <https://youtu.be/JIZ2KM-rYjk>

Appendix

Proof of Proposition 2. This proof is strikingly similar to the proof for the controller on $SO(3) \times \mathbb{R}^3$ in [7, Appendix D], and a sketch is given here with some key intermediary expressions. Define a Lyapunov function comprised of two parts, one relating the attitude subsystem, \mathcal{V}_a , and one relating to the translation subsystem, \mathcal{V}_p . The former is defined as in (14), and the latter is defined analogously, as

$$\mathcal{V}^a \stackrel{\text{def}}{=} k_X \Gamma(\mathbf{X}_d, \mathbf{X}) + c_a \mathbf{e}_\omega \cdot \mathbf{e}_X + \frac{1}{2} \mathbf{e}_\omega \cdot \mathbf{J} \mathbf{e}_\omega, \quad (32a)$$

$$\mathcal{V}^p \stackrel{\text{def}}{=} \frac{1}{2} k_p \|\mathbf{e}_p\|^2 + \frac{1}{2} m \|\mathbf{e}_v\|^2 + c_p \mathbf{e}_p \cdot \mathbf{e}_v, \quad (32b)$$

respectively. As the attitude dynamics are actuated along this trajectory by Proposition 1, consider initial attitude errors on

$$(\mathbf{X}_e(t_o), \mathbf{e}_\omega(t_o)) \in \{(\mathbf{X}_e, \mathbf{e}_\omega) \in \mathcal{L}_\phi \times \mathbb{R}^3 \mid \mathcal{V}^a|_{c_a=0} \leq k_X \phi\}, \quad (33)$$

where

$$\mathcal{L}_\phi = \{\mathbf{X}_d^* \mathbf{X} \in SU(2) \mid \Gamma(\mathbf{X}_d, \mathbf{X}) \leq \phi < 2\}. \quad (34)$$

From the result in Proposition 1, it follows that $\mathbf{X}_e(t) \in \mathcal{L}_\phi$ for all $t \geq t_o$ if $\mathcal{V}^a(t_o)|_{c_a=0} \leq k_X \phi$, and that the errors converge exponentially to $(\mathbf{X}_e, \mathbf{e}_\omega) = (\mathbf{I}, \mathbf{0})$. As such, the main idea of the proof is to conduct the stability analysis on a domain $D = \{(\mathbf{e}_p, \mathbf{e}_v, \mathbf{X}_e, \mathbf{e}_\omega) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathcal{L}_\phi \times \mathbb{R}^3 \mid \|\mathbf{e}_p\| \leq B_p\}$, restricting the domain by making ϕ and B_p sufficiently small such that all solutions remain on D .

In the following, we consider performing the stability analysis on the domain characterized by D with $\phi = 2^{-3}$ with a Lyapunov function candidate $\mathcal{V} = \mathcal{V}^a + \mathcal{V}^p$.

Translation error dynamics

by plugging in the proposed feedback law, the translation error dynamics can be written

$$m \dot{\mathbf{e}}_v = m \ddot{\mathbf{p}} - m \ddot{\mathbf{p}}_r = -m g \mathbf{e}_3 + \mathbf{f} \mathbf{R} \mathbf{e}_3 - m \ddot{\mathbf{p}}_r. \quad (35)$$

Note that $\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3 = \mathbf{b}_{d3} \cdot \mathbf{b}_3 > 0$ if we can ensure that (17) holds. If so, the term $\frac{f}{\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3} \mathbf{R}_d \mathbf{e}_3$ is well defined, and as such, (35) can be expressed

$$m \dot{\mathbf{e}}_v = -m g \mathbf{e}_3 - m \ddot{\mathbf{p}}_r + \frac{f}{\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3} \mathbf{R}_d \mathbf{e}_3 + \bar{\mathbf{f}}, \quad (36)$$

where

$$\bar{\mathbf{f}} = \frac{f}{\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3} \mathbf{R}_d \mathbf{e}_3 \left((\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3) \mathbf{R} \mathbf{e}_3 - \mathbf{R}_d \mathbf{e}_3 \right). \quad (37)$$

In addition, recall that the actuating force is computed by

$$\mathbf{f}_d = -k_p \mathbf{e}_p - k_v \mathbf{e}_v + m g \mathbf{e}_3 + m \ddot{\mathbf{p}}_r. \quad (38)$$

As $\mathbf{b}_{d3} = \mathbf{f}_d \|\mathbf{f}_d\|^{-1}$, $\mathbf{f}_d = \|\mathbf{f}_d\| \mathbf{b}_{d3} = \|\mathbf{f}_d\| \mathbf{R}_d \mathbf{e}_3$, and

$$\frac{f}{\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3} \mathbf{R}_d \mathbf{e}_3 = \frac{\mathbf{f}_d \cdot \mathbf{R} \mathbf{e}_3}{\mathbf{e}_3 \mathbf{R}_d^\top \mathbf{R} \mathbf{e}_3} \mathbf{R}_d \mathbf{e}_3 = \mathbf{f}_d. \quad (39)$$

Insertion of this expression in (36) yields

$$m \dot{\mathbf{e}}_v = -k_p \mathbf{e}_p - k_v \mathbf{e}_v + \bar{\mathbf{f}}. \quad (40)$$

To proceed, we start by bounding $\bar{\mathbf{f}}$ in the control errors using the assumptions of the theorem, but first, we derive the expression for the time-derivative of the part of the Lyapunov function associated with the translation errors.

Translation Lyapunov function candidate

Differentiation of the Lyapunov function associated with the translation dynamics along the solutions of the controlled system yields

$$\begin{aligned} \dot{\mathcal{V}}^p &= -(k_v - c_p)\|e_v\|^2 - \frac{c_p k_p}{m}\|e_p\|^2 - \frac{c_p k_v}{m}(e_p \cdot e_v) + \\ &\quad \bar{\mathbf{f}} \cdot (c_p m^{-1} e_p + e_v). \end{aligned} \quad (41)$$

Furthermore, by (39), we have that

$$\|\bar{\mathbf{f}}\| \leq \|\mathbf{f}_d\| \|(e_3 \mathbf{R}_d^\top \mathbf{R} e_3) \mathbf{R} e_3 - \mathbf{R}_d e_3\|, \quad (42)$$

where the second term is recognized as the sine angle of the eigenaxis rotation angle between \mathbf{b}_{d3} and \mathbf{b}_3 , as pointed out in [7]. As such, we can utilize the fact that this rotation angle is bounded in the control errors on D , as per (17). By this simple observation, we obtain

$$\begin{aligned} \|\bar{\mathbf{f}}\| &\leq \|\mathbf{f}_d\| \|(e_3 \mathbf{R}_d^\top \mathbf{R} e_3) \mathbf{R} e_3 - \mathbf{R}_d e_3\| \\ &\leq (k_p \|e_p\| + k_v \|e_v\| + B) \|(e_3 \mathbf{R}_d^\top \mathbf{R} e_3) \mathbf{R} e_3 - \mathbf{R}_d e_3\| \\ &\leq (k_p \|e_p\| + k_v \|e_v\| + B) 4 \|e_X\| \\ &\leq (k_p \|e_p\| + k_v \|e_v\| + B) \alpha, \end{aligned}$$

where the second inequality follows from assumption (A3), and the third and fourth hold for all trajectories on D from the observation regarding the sine angle in (18). Insertion of this bound in (43) yields

$$\begin{aligned} \dot{\mathcal{V}}^p &= -(k_v - c_p)\|e_v\|^2 - \frac{c_p k_p}{m}\|e_p\|^2 - \frac{c_p k_v}{m}(e_p \cdot e_v) + \bar{\mathbf{f}} \cdot \left(\frac{c_p}{m} e_p + e_v\right) \\ &= -(k_v(1 - \alpha) - c_p)\|e_v\|^2 - \frac{c_p k_p}{m}(1 - \alpha)\|e_p\|^2 \\ &\quad + \frac{c_p k_v}{m}(1 + \alpha)\|e_p\| \|e_v\| \\ &\quad + 4 \|e_X\| \left(B \left(\frac{c_p}{m} \|e_p\| + \|e_v\| \right) + k_p \|e_p\| \|e_v\| \right) \\ &\leq -z_p^\top \mathbf{W}^{pp} z_p + z_p^\top \mathbf{W}^{pa} z_a, \end{aligned} \quad (43)$$

with \mathbf{W}^{pp} and \mathbf{W}^{pa} defined as in the proposition statement, in (24) and (25), respectively. In addition, we note that

$$z_p^\top \mathbf{M}_1^{pp} z_p \leq \mathcal{V}^p \leq z_p^\top \mathbf{M}_2^{pp} z_p, \quad (44)$$

for \mathbf{M}_1^{pp} and \mathbf{M}_2^{pp} defined as in (24) of the proposition.

Complete Lyapunov function candidate

In addition to these definitions, take \mathbf{M}_1^{aa} , \mathbf{M}_2^{aa} , \mathbf{W}^{aa} , to be the matrices in (12) associated with the controller in Theorem 1, also given in (24). In addition, define the matrices

$$\bar{\mathbf{M}}_1 = \text{diag}(\mathbf{M}_1^{pp}, \mathbf{M}_1^{aa}), \quad \bar{\mathbf{M}}_2 = \text{diag}(\mathbf{M}_2^{pp}, \mathbf{M}_2^{aa}), \quad \bar{\mathbf{W}} = \begin{bmatrix} \mathbf{W}^{pp} & -\frac{1}{2} \mathbf{W}^{pa} \\ \star & \mathbf{W}^{aa} \end{bmatrix}. \quad (45)$$

where $\bar{\mathbf{M}}_i \succ \mathbf{0}$ if $\mathbf{M}_i^{jj} \succ \mathbf{0}$. For the combined Lyapunov function candidate $\mathcal{V} = \mathcal{V}^p + \mathcal{V}^a$, we find that

$$\bar{\mathbf{z}}^\top \bar{\mathbf{M}}_1 \bar{\mathbf{z}} \leq \mathcal{V} \leq \bar{\mathbf{z}}^\top \bar{\mathbf{M}}_2 \bar{\mathbf{z}}. \quad (46)$$

Differentiating \mathcal{V} along the closed-loop solutions on D yields

$$\dot{\mathcal{V}} = \dot{\mathcal{V}}^p + \dot{\mathcal{V}}^a \leq -\bar{\mathbf{z}}^\top \bar{\mathbf{W}} \bar{\mathbf{z}}. \quad (47)$$

By assumption (A5), we have that

$$\begin{aligned} \dot{\mathcal{V}} &\leq -\bar{\mathbf{z}}^\top \bar{\mathbf{W}} \bar{\mathbf{z}} \\ &\leq -\lambda_m(\mathbf{W}^{pp}) \|z_p\|^2 + \|\mathbf{W}^{pa}\| \|z_p\| \|z_a\| - \lambda_m(\mathbf{W}^{aa}) \|z_a\|^2 \\ &\leq - \begin{bmatrix} \|z_p\| \\ \|z_a\| \end{bmatrix} \begin{bmatrix} \lambda_m(\mathbf{W}^{pp}) & -\frac{1}{2} \|\mathbf{W}^{pa}\| \\ \star & \lambda_m(\mathbf{W}^{aa}) \end{bmatrix} \begin{bmatrix} \|z_p\| \\ \|z_a\| \end{bmatrix} \\ &\leq -B_z (\|z_p\|^2 + \|z_a\|^2). \end{aligned} \quad (48)$$

Consequently, the Lyapunov function time-derivative is negative definite in $\bar{\mathbf{z}}$ along the solutions of the error dynamics on D . By (48), it also follows that \mathcal{V} is continuously differentiable on D as all of the signals constituting $\dot{\mathcal{V}}$ are bounded in the initial errors. As such, [18, Theorem 4.10] yields that the origin $\bar{\mathbf{z}} = \mathbf{0}$ is UES on the domain D . \square