Run Time Assurance for Spacecraft Attitude Control Under Nondeterministic Assumptions

Matthew Abate, *Student Member, IEEE*, Mark Mote, Mehregan Dor, Corbin Klett, Sean Phillips, Kendra Lang, Panagiotis Tsiotras, *Fellow, IEEE*, Eric Feron, *Member, IEEE*, and Samuel Coogan, *Senior Member, IEEE*

***Abstract*— This paper presents a comprehensive development and testing of a Run Time Assurance (RTA) filter for a torque-controlled spacecraft in free rotational motion with torque actuation limits for which the objective is to enforce a line-of-sight constraint. A nondeterministic dynamical model is considered for the spacecraft that accounts for disturbance torques, and a guaranteed safe RTA filter is constructed using recent results from mixed monotone systems theory for reachable set overapproximations and optimization-based computation of invariant sets. The RTA filter ensures that the system is always within reach of an a priori safe terminal set by computing reachable sets of the dynamics online at run time. The approach is demonstrated on the Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS) platform at the Georgia Institute of Technology. In the experiment, potentially unsafe inputs are provided by a human, and the RTA filter overrides the human-commanded inputs when necessary to guarantee safety. The controller update rate for the ASTROS platform is about 10Hz, while the RTA filter requires about 1 millisecond of computation time per controller update.**

***Index Terms*— Aerospace safety, Collision avoidance, Mixed monotone systems, Optimization methods**

This work was partially supported by the Air Force Office of Scientific Research under grant FA9550-19-1-0015, and by the Department of Defense under STTR contract FA9453-19-C-0621. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense or of the United States Air Force.

Approved for public release; distribution is unlimited. Public Affairs approval #AFRL-2022-4826

M. Abate is with the School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA matt.abate@gatech.edu.

M. Mote is founder CEO of Pytheia, Atlanta, GA, 30308, USA mark@pytheia.com.

M. Dor is with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, 30332, USA mdor3@gatech.edu.

C. Klett is with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA corbin@gatech.edu.

S. Phillips is with the Space Control Branch, Air Force Research Laboratory, Kirtland AFB, NM, 87117, USA.

K. Lang is with Verus Research; Albuquerque, NM 87110 USA kendra.lang@verusresearch.net.

P. Tsiotras is with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA tsiotras@gatech.edu.

E. Feron is with the School of Electrical and Computer Engineering, King Abdullah University of Science and Technology, Thuwal, 23955, Saudi Arabia eric.feron@kaust.edu.sa.

S. Coogan is with the School of Electrical and Computer Engineering and the School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, 30332, GA, USA sam.coogan@gatech.edu.

I. Introduction

THE recent proliferation of commercial and government space missions, particularly in low-Earth orbit, has motivated the need for increased autonomy capabilities for space vehicles. Such advancements in autonomy can save on costs and risks associated with human-in-the-loop operations while improving vehicle reliability and performance. The term *assured autonomy* has been applied in this context to describe a vehicle or system that has the ability to safely perform a complex mission or a set of missions in the presence of uncertainty and without human intervention [1], [2].

Run time assurance (RTA) provides an elegant and highly-adaptable methodology for assured autonomy, with current research and applications in the domain of spacecraft systems [2]–[4]. The approach is associated with the system architecture depicted in Figure 1, where an assurance mechanism is employed between the plant and primary controller. Given a desired control policy, an assurance mechanism will *filter* online the desired input in such a way that preserves system safety, while also ensuring that the desired control input is passed to the system when it is safe to do so. In this way, the addition of an RTA works to decouple the task of enforcing safety constraints from all other objectives of the controller, if any, and allows the designer to sidestep the common trade-off between performance and safety. Well-known examples of RTA mechanisms include the Simplex architecture [5]–[8], which switches to a backup control scheme when necessary (see Figure 1), and control barrier functions (CBFs) [9]–[11], which adjust the desired control actions in a minimally invasive way to ensure forward invariance of a predetermined safe subset of the state space.

This paper presents a comprehensive development and testing of an RTA filter for a torque-controlled spacecraft in free rotational motion subject to a line-of-sight constraint. Our RTA filter uses mixed monotone systems theory to compute reachable set overapproximations within the control loop, and system safety is enforced online using a precomputed but conservative invariant safe set. Our method accommodates torque disturbances and guarantees safety for the spacecraft when the disturbance input is bounded within a given range. Further, we design a novel Sum-of-Squares optimization program to compute a robust invariant safe set given the system's safety constraint and the disturbance input bounds."

The proposed algorithm can be summarized as follows. We

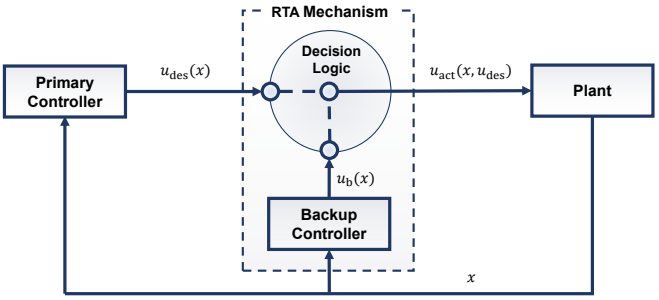

Fig. 1: Run time assured control system architecture. Control inputs $\mathbf{u}_{\text{des}}$ are suggested by a potentially unsafe primary controller, possibly a human operator. When necessary to preserve system safety, the run time assurance (RTA) mechanism applies, instead, a known safe backup control input $\mathbf{u}_{\text{b}}$ to the plant. The backup controller is certified to be safe via the explicit knowledge of a forward invariant set and, in this way, system safety is ensured for all time.

first assume the availability of a backup feedback control policy that is verified *a priori* to render a given subset of the state space robustly forward invariant. Motivated by applications where such subsets are generally conservative, our objective is to allow the system to safely evolve beyond this initial verified subset. To do so, we propose computing an overapproximation of the system's reachable set under the backup control policy. This is used to guarantee safety of states outside the verified subset by proving that the backup controller provides a safe trajectory that returns the system to the verified subset. Specifically, if the reachable set becomes fully contained within the verified safe subset at some time along the prediction horizon, and it does not intersect with the set of states where the line-of-sight constraint is violated before that time, then a safe return trajectory is ensured. We construct this assurance mechanism in the following way.

We begin by computing a safe backup control policy and the corresponding safe subset of the state space in Section III. The backup controller is designed to drive trajectories to the backup set while saturating to adhere to the input constraints. These actuation constraints on the backup controller make it challenging to explicitly obtain a large, robust, and safe forward invariant set *a priori*, further motivating the approach proposed in this study where a conservative-but-verified safe subset is coupled with reachable set computations to achieve safety beyond the verified subset.

To compute reachable sets within the control loop, we propose using the theory of mixed monotone dynamical systems, which provides an efficient technique for overapproximating reachable sets using hyperrectangles, and which has previously been demonstrated in online safety applications [12]–[15]. A dynamical system, possibly subject to a disturbance input, is mixed monotone when there exists a related decomposition function that separates the initial system dynamics into *cooperative* and *competitive* state interactions [16]. Mixed monotonicity applies to continuous-time systems [17]–[21], discrete-time systems [22], [23], controlled systems [16], [24], and systems with disturbances [23]–[25]. This is the main point of study in Section IV, where we show that closed-loop spacecraft dynamics under the backup controller are mixed monotone, and we explicitly compute a decomposition function for this system. An example is provided in the same section where we demonstrate how the decomposition function is applied for the efficient computation of reachable sets. In summary, mixed monotonicity enables our run time assurance approach in three main ways:

- Mixed monotonicity provides a computationally efficient approach for overapproximating the reachable set of the spacecraft dynamics.
- Mixed monotonicity readily allows for the incorporation of bounded uncertainties in the state and dynamics, as well as constraints on the control input.
- In the case of line-of-sight constraints for spacecraft systems, mixed monotonicity provides an efficient technique for checking whether the system's reachable set overapproximation in the space of line-of-sight angles is contained fully within the safe region, and this is true even though the line-of-sight angle is not considered to be a state of the system.

In Section V, we provide details of our hardware testbed, the Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS) platform, at the Georgia Institute of Technology. The experimental setup is shown later in Figure 6, and a video of the experiment is available at `https://youtu.be/g1-zMepDm1I`.

Notation

Given two vectors $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ we denote the vector concatenation of x and y by $(x, y) := [x^\top, y^\top]^\top \in \mathbb{R}^{m+n}$. To denote collections of elements within a vector or a matrix, we write $x_{i:j} \in \mathbb{R}^{j-i+1}$ to denote the i-through-j^{th} elements of $x \in \mathbb{R}^n$ and $A_{i,:} \in \mathbb{R}^{1\times m}$ to denote the i^{th} row of $A \in \mathbb{R}^{n\times m}$; that is,

$$\begin{aligned} x_{i:j} &= (x_i, x_{i+1}, \cdots, x_{j-1}, x_j) \\ A_{i,:} &= [A_{i,1}\,,\, A_{i,2}\,, \cdots,\, A_{i,m-1}\,,\, A_{i,m}]. \end{aligned} \tag{1}$$

Given a matrix $A \in \mathbb{R}^{n\times m}$, we denote by $[A]^+$ and $[A]^-$ the positive and negative parts of A, respectively, that is,

$$\begin{aligned} [A]^+_{i,j} &= \begin{cases} A_{i,j} & \text{if } A_{i,j} \geq 0, \\ 0 & \text{if } A_{i,j} < 0, \end{cases} \\ [A]^- &= A - [A]^+. \end{aligned} \tag{2}$$

Given $x, y \in \mathbb{R}^n$ we write $x \preceq y$ if and only if $x_i \leq y_i$ for all i. In the instance where $x \preceq y$, we let

$$[x, y] := \{z \in \mathbb{R}^n \mid x \preceq z \text{ and } z \preceq y\} \tag{3}$$

denote the hyperrectangle defined by the endpoints x and y and we let

$$\langle\!\langle x, y \rangle\!\rangle := \{z \in \mathbb{R}^n \mid z_i \in \{x_i, y_i\}\ \forall i = 1, \ldots, n\} \tag{4}$$

denote the finite set of 2^n corners of $[x, y]$. Finally, given $a = (x, y) \in \mathbb{R}^{2n}$ with $x \preceq y$, we denote by $[\![a]\!] := [x, y]$ the hyperrectangle formed by the first n and last n components of a. Note that for scalars $x, y \in \mathbb{R}$ with $x \leq y$, the set $[x, y] \subset \mathbb{R}$ is the interval of real numbers between x and y, inclusive.

II. Problem Statement and Proposed Solution for Obstacle Avoidance Case Study

We consider the problem of spacecraft attitude control in the presence of a safety constraint on the allowable line-of-sight angle of a spacecraft. A safety assured controller is one that is guaranteed to not violate the line-of-sight constraint. In this paper, the proposed solution for obtaining an assured controller is with an RTA mechanism that filters an unverified control input online in order to ensure system safety at run time. In this section, we first introduce the model for the spacecraft attitude dynamics. Next, we discuss the real-world system requirements that inform the construction of the RTA mechanism. Finally, we present an overview of the RTA algorithm that is further detailed in the following sections.

A. Spacecraft Attitude Dynamics

We study a rotating rigid body model of a spacecraft given by

$$\dot{\omega}(t) = J^{-1}(-\omega(t) \times J\omega(t) + u(t) + w(t)), \tag{5}$$

where $\omega(t) = (\omega_x(t), \omega_y(t), \omega_z(t)) \in \mathbb{R}^3$ is the vector of spacecraft angular rates at time t, $u(t) \in \mathcal{U} \subset \mathbb{R}^3$ is the vector of applied torque inputs, and $w(t) \in \mathcal{W} \subset \mathbb{R}^3$ is a vector of disturbance torques. The set $\mathcal{U}$ incorporates, for example, actuation constraints, and $\mathcal{W}$ accommodates known disturbance bounds. Further, the matrix $J \in \mathbb{R}^{3\times3}$ in (5) is the inertia matrix for the spacecraft, which is symmetric and positive-definite. Below, for time-varying functions, we generally omit the explicit dependence on time t.

The components of ω describe the rotation of the body-fixed reference frame $\mathcal{F}_\mathrm{B}$ with respect to the inertial frame $\mathcal{F}_\mathrm{I}$ and the vectors ω, u, and w are expressed in $\mathcal{F}_\mathrm{B}$; see Figure 2. We let $\mathcal{F}_\mathrm{I} := (\mathcal{O}, \widehat{I}, \widehat{J}, \widehat{K})$ and $\mathcal{F}_\mathrm{B} := (\mathcal{O}, \widehat{\imath}, \widehat{\jmath}, \widehat{k})$ so that, without loss of generality, the origin of the inertial frame $\mathcal{O}$ is the same as that of the body frame, and the unit vectors $\widehat{\imath}$, $\widehat{\jmath}$, $\widehat{k}$ and $\widehat{I}$, $\widehat{J}$, $\widehat{K}$ form right-handed orthogonal bases for $\mathcal{F}_\mathrm{B}$ and $\mathcal{F}_\mathrm{I}$, respectively, so that $\widehat{\imath} \times \widehat{\jmath} = \widehat{k}$ and $\widehat{I} \times \widehat{J} = \widehat{K}$.

There are many possible ways to represent the attitude (orientation) dynamics of a rotating body; see [26] for a short review. For reasons that will become clear in Section III, in this work, we choose a two-parameter representation of the attitude that describes only the orientation of the body-fixed $\widehat{k}$-axis with respect to the inertially fixed $\widehat{K}$-axis[1]. This two-parameter attitude description, originally introduced in [27], enforces a natural dimensionality reduction of the attitude that eliminates the irrelevant rotation about the pointing axis from the attitude description. Its utility for solving line-of-sight/pointing problems has been demonstrated, for example, in [28], [29]. In particular, we describe the spacecraft attitude using $\rho = (\rho_1, \rho_2) \in \mathbb{R}^2$, where the time-evolution of ρ is

[1]The general theory posited in this section for parameterizing attitude extends naturally to other orientation definitions, as well, so that one may choose to represent orientation using, e.g., the body-fixed $\widehat{\imath}$-axis and the inertially fixed $\widehat{J}$-axis. Moreover, the particular parameterization used in this work, which uses $\widehat{k}$ and $\widehat{K}$, is chosen without loss of generality and serves only for ease of exposition.

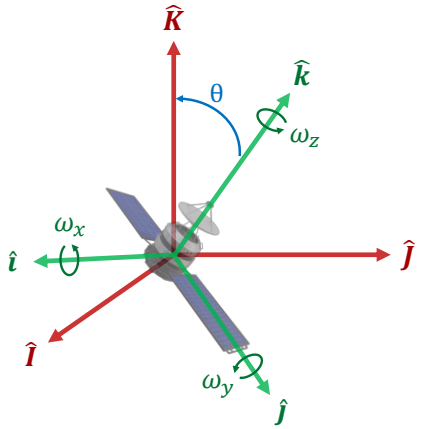

Fig. 2: Depiction of spacecraft system. The inertially-fixed reference frame $\mathcal{F}_\mathrm{I} := (\mathcal{O}, \widehat{I}, \widehat{J}, \widehat{K})$ is shown in red and the body-fixed reference frame $\mathcal{F}_\mathrm{B} := (\mathcal{O}, \widehat{\imath}, \widehat{\jmath}, \widehat{k})$ is shown in green. The line-of-sight angle, which is the angle made between the $\widehat{k}$ and $\widehat{K}$ directions is given by (7) and is shown in blue. Note that the vectors ω, u, and w in (5) are all taken with respect to $\mathcal{F}_\mathrm{B}$.

described by

$$\begin{aligned}\dot{\rho}_1 &= \omega_z\rho_2 + \omega_y\rho_1\rho_2 + \frac{\omega_x}{2}(1 + \rho_1^2 - \rho_2^2),\\ \dot{\rho}_2 &= -\omega_z\rho_1 + \omega_x\rho_1\rho_2 + \frac{\omega_y}{2}(1 + \rho_2^2 - \rho_1^2).\end{aligned} \tag{6}$$

Following the derivations of [27], observe that the *line-of-sight angle* of the spacecraft, which is the angle made between the $\widehat{k}$ and $\widehat{K}$ directions, is given by

$$\theta(\rho, \omega) := \arccos\left(\frac{1 - \rho_1^2 - \rho_2^2}{1 + \rho_1^2 + \rho_2^2}\right), \tag{7}$$

i.e., $\rho = 0$ when $\widehat{k}$ points in the $\widehat{K}$-direction. See Figure 2.

Combining the orientation dynamics (6) with the rotational velocity dynamics (5), we have that the full spacecraft dynamics are given by

$$\begin{bmatrix}\dot{\rho}_1\\ \dot{\rho}_2\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}\omega_z\rho_2 + \omega_y\rho_1\rho_2 + \frac{\omega_x}{2}(1 + \rho_1^2 - \rho_2^2)\\ -\omega_z\rho_1 + \omega_x\rho_1\rho_2 + \frac{\omega_y}{2}(1 + \rho_2^2 - \rho_1^2)\\ J^{-1}(-\omega \times J\omega + u + w)\end{bmatrix}, \tag{8}$$

which we hereafter denote as

$$\dot{x} = f(x, u, w), \tag{9}$$

with state $x = (\rho, \omega) \subset \mathcal{X} \in \mathbb{R}^5$ and where $u \in \mathcal{U} \subset \mathbb{R}^3$ and $w \in \mathcal{W} \subset \mathbb{R}^3$ retain their definitions from (5). In the following, we say that a control policy $\mathbf{u} : \mathbb{R} \times \mathcal{X} \to \mathbb{R}^3$ is *admissible on the set* $S \subset \mathcal{X}$ if $\mathbf{u}(t; x) \in \mathcal{U}$ for all $x \in S$ and for all $t \geq 0$.

Additionally, we denote by $\Phi(t; x, \mathbf{u}, \mathbf{w})$ the state of (9) reached at time t when starting from state $x \in \mathcal{X}$ at time 0 and evolving subject to the feedback control $\mathbf{u}(\,\cdot\,)$ and the disturbance $\mathbf{w}(\,\cdot\,)$. Throughout this paper, we assume that $\Phi(t; x, \mathbf{u}, \mathbf{w})$ is unique when it exists. The *time-t reachable set of* (9) *from* $S \subset \mathcal{X}$ *under* $\mathbf{u}$ is denoted as

$$R(t; S, \mathbf{u}) := \{\Phi(t; x, \mathbf{u}, \mathbf{w}) \in \mathcal{X} \,|\, x \in S \text{ for some } \mathbf{w} : [0, t] \to \mathcal{W}\}, \tag{10}$$

which is the set of states reachable from S at time $t \geq 0$ under some disturbance signal.

B. Safe Operational Behavior for Spacecraft Systems

Safety for the spacecraft system (9) is formalized via a line-of-sight constraint. Specifically, we say that a state $x = (\rho, \omega) \in \mathcal{X}$ is safe when $\theta(x) \leq \theta_{\max}$, where $\theta(x)$ is given by (7) and where $\theta_{\max}$ is a parameter describing the maximum allowable line-of-sight angle, which is assumed to be fixed *a priori*. To assess the safety of states, we employ a safety constraint function

$$\varphi(x) := (1 + \rho_1^2 + \rho_2^2)\left(\cos\left(\theta(x)\right) - \cos\left(\theta_{\max}\right)\right), \quad (11)$$

or, equivalently,

$$\varphi(x) := (1 - \cos(\theta_{\max})) - (1 + \cos\left(\theta_{\max}\right))\left(\rho_1^2 + \rho_2^2\right), \quad (12)$$

where we observe that $\varphi(x) \geq 0$ only when x is safe and $\varphi(x) < 0$ otherwise. The set of all states $x \in \mathcal{X}$ that satisfy $\varphi(x) \geq 0$ is referred to as the *constraint set* and is denoted by

$$\mathcal{C}_{\mathrm{A}} := \{x \in \mathcal{X} \,|\, \varphi(x) \geq 0\}. \quad (13)$$

Thus, the goal is to construct a controller $\mathbf{u}$ that ensures $\Phi(t; x_0, \mathbf{u}, \mathbf{w}) \in \mathcal{C}_{\mathrm{A}}$ for all $t \geq 0$ and for all disturbance signals $\mathbf{w} : [0, \infty) \to \mathcal{W}$.

C. Run Time Assurance Solution

Our goal is to design a feedback controller that ensures the satisfaction of the safety constraint (12) along trajectories of (9). One way to establish controller safety is through invariance.

Definition 1 (Robust Forward Invariance)**.** A set $S \subseteq \mathcal{X}$ is robustly forward invariant for (9) under the feedback control $\mathbf{u}$ if $\Phi(t; x, \mathbf{u}, \mathbf{w}) \in S$ for all $x \in S$, all $t \geq 0$ and all disturbance inputs $\mathbf{w} : [0, t] \to \mathcal{W}$, whenever $\Phi(t; x, \mathbf{u}, \mathbf{w})$ exists. ■

A feedback control policy $\mathbf{u}$ is said to be safe for (9) if there exists a set S so that S is forward invariant for (9) under $\mathbf{u}$ and $S \subseteq \mathcal{C}_{\mathrm{A}}$. Applying $\mathbf{u}$ to (9) then ensures $\theta(x) \leq \theta_{\max}$ for all time, so long as $x(0) \in S$. While forward invariance provides a theoretical foundation for assessing safety, for complex nonlinear systems, such as (9), with control input constraints and disturbances, there generally do not exist prescriptive formulas for generating safe controllers and forward invariant regions. The solution in this work is to employ a *run time assurance mechanism* (RTA) in-the-loop, which filters an unsafe desired control input at run time to ensure the existence of an implicitly defined forward invariant set in the statespace; see Figure 1. The RTA switches between two competing control policies: a performance-driven *desired* control policy and a verified *backup* control policy.

First, consider a desired control policy $\mathbf{u}_{\mathrm{des}}(t; x)$ for the system (9). The desired controller is assumed to satisfy some performance control objective but is not directly applicable to (9) due to the fact that (i) $\mathbf{u}_{\mathrm{des}}$ may not be safe, that is, applying $\mathbf{u}_{\mathrm{des}}$ may cause the system to leave $\mathcal{C}_{\mathrm{A}}$, and (ii) $\mathbf{u}_{\mathrm{des}}$ may not be admissible, that is, there may exist a state $x \in \mathcal{C}_{\mathrm{A}}$ so that $\mathbf{u}_{\mathrm{des}}(t; x) \notin \mathcal{U}$ at some time t.

Next, suppose we have knowledge of a backup control policy $\mathbf{u}_{\mathrm{b}}(x)$ for (9), which is certifiable safe via an explicitly defined, but potentially conservative, forward invariant *safe terminal set* $\mathcal{C}_{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}}$ and is also admissible, i.e., $\mathbf{u}_{\mathrm{b}}(x) \in \mathcal{U}$ for all $x \in \mathcal{C}_{\mathrm{b}}$. Applying $\mathbf{u}_{\mathrm{b}}(x)$ to (9) now ensures that all state trajectories remain in $\mathcal{C}_{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}}$ and guarantees system safety for all time. However, applying such a backup controller may not be preferable; indeed, backup controllers are typically designed without considering performance objectives and it is possible that, for instance, $\mathcal{C}_{\mathrm{b}}$ is too small to allow for satisfaction of performance criteria.

We now have the requisite tools to construct our RTA mechanism; see Figure 3 for a topological description of the algorithm. We assume the control input is updated with a time step of Δt. At every update time $t = k\Delta t$ for $k = 0, 1, 2, \ldots$, the RTA consists of the following steps:

Step 1) Receive as inputs:
- The current system state, $x_0 := x(t) = x(k\Delta t)$
- The desired control policy $\mathbf{u}_{\mathrm{des}}$, and
- The backup control policy $\mathbf{u}_{\mathrm{b}}$.

Step 2) Compute an overapproximation $\mathcal{X}_{\mathrm{p}} \supseteq R(t_{\mathrm{des}}; x_0, \mathbf{u}_{\mathrm{des}})$ for some constant $t_{\mathrm{des}} > 0$.

Step 3) Compute a sequence of overapproximations under the backup controller $\mathcal{X}_{\mathrm{b}}^k \supseteq R(kt_{\mathrm{b}}/k^*; \mathcal{X}_{\mathrm{p}}, \mathbf{u}_{\mathrm{b}})$ for $k = 1, \ldots, k^*$ for some k^*.

Step 4) If $\varphi(x) \geq 0$ for all $x \in \mathcal{X}_k^{\mathrm{b}}$ and all $k \in \{1, \cdots, k^*\}$, and $\mathcal{X}_{k^*}^{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{b}}$, the desired input $\mathbf{u}_{\mathrm{des}}(t, x_0)$ is allowed to pass to the system unaltered for the next controller update step of length Δt. Otherwise, when the above constraints are not met, the backup control input $\mathbf{u}_{\mathrm{b}}(x_0)$ is applied to the system.

We make the following remarks:

- In the above RTA algorithm, t_{des}, t_{b}, and k^* are tunable parameters. Note that we do not require t_{des} or t_{b} to be a multiple of Δt.
- While the controller is updated with discrete time step Δt, the reachable set overapproximations are computed in continuous-time. In principle, these sets are obtained by any method for obtaining overapproximations of reachable sets of continuous-time systems, but many reachability methods are not computationally efficient enough to be used in realtime in the control loop. Below, we propose a method for obtaining reachable set overapproximations with a single simulation of an ordinary differential equation representing an appropriately constructed embedding system.
- The RTA logic is executed at the beginning of each controller update step. We assume that the computation time required for evaluating the RTA logic is negligible compared to the time step Δt. In the hardware demonstration below, the RTA logic execution time is about one-hundredth of the controller update time.
- We assume that the entire desired control policy $\mathbf{u}_{\mathrm{des}}$ is available to the RTA, however, in practice, we only need the desired input for the current controller update

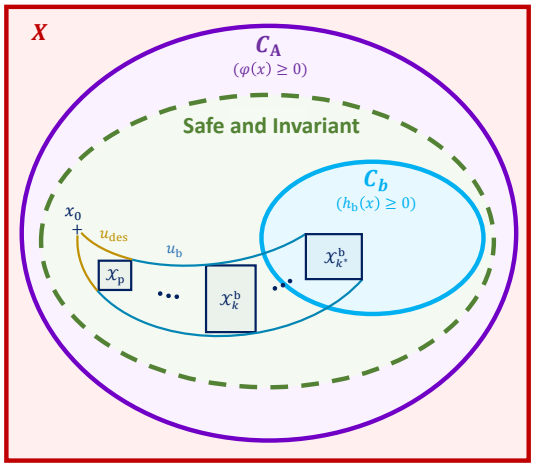

Fig. 3: Topological depiction of the RTA algorithm. The state space of (9) is $\mathcal{X} \subset \mathbb{R}^5$, and the set of states satisfying the safety constraint (12) is $\mathcal{C}_\mathrm{A} \subset \mathcal{X}$. The RTA is constructed via a backup controller $\mathbf{u}_\mathrm{b}$, and the set $\mathcal{C}_\mathrm{b}$ is forward invariant for (9) under $\mathbf{u}_\mathrm{b}$. At each time t, when (9) is at state $x(t)$, the RTA first computes an overapproximation of the time-t_des reachable set of (9) from $x(t)$ under $\mathbf{u}_\mathrm{des}$, and this overapproximation is denoted $\mathcal{X}_\mathrm{p}$. Next, for all $t \in [t_\mathrm{p}, t_\mathrm{p} + t_\mathrm{b}]$, the RTA computes an overapproximation of the time-t reachable set of (9) from $\mathcal{X}_\mathrm{p}$ under $\mathbf{u}_\mathrm{b}$. In instances where (i) there exists a time $T \in [0, t_\mathrm{p}+t_\mathrm{b}]$ for which $R(T; x(t)) \subset \mathcal{C}_\mathrm{b}$ and (ii) for all $\tau \in [0, T]$ we have $R(\tau; x(t)) \subset \mathcal{C}_\mathrm{A}$, the desired control input $\mathbf{u}_\mathrm{des}(x)$ is applied to (9). Otherwise, the backup control input $\mathbf{u}_\mathrm{b}(x)$ is applied. In this way, we guarantee that $x(t) \in \mathcal{C}_\mathrm{A}$ for all $t \geq 0$, that is, the control input produced ensures the satisfaction of the safety constraint (12).

step, and then it is sufficient to have an approximation of the desired policy from $t + \Delta t$ to t_p, which is required to compute $\mathcal{X}_\mathrm{p}$ in Step 2. This approximation can be obtained by, e.g., assuming a zero-order hold on the desired input. This is done in the example of Section V, where the desired control inputs are generated by a human with a joystick.

The RTA mechanism acts to *filter* the desired control input online, and this approach guarantees system safety while also ensuring that the desired control input is passed to the system when it is safe to do so. A benefit of this approach, as we shall see later on, is that it allows system trajectories to leave the backup set $\mathcal{C}_\mathrm{b}$ when it is verified online that the current system state can safely return to $\mathcal{C}_\mathrm{b}$ on a finite time horizon. In this way, employing an RTA mechanism eases the task of computing a large forward invariant set $\mathcal{C}_\mathrm{b}$ a priori and offloads computation to online reachability analysis.

We begin by computing a safe backup control policy $\mathbf{u}_\mathrm{b}$ and the corresponding safe set $\mathcal{C}_\mathrm{b}$ for (9) in Section III. The backup controller is designed to cancel nonlinearities in the dynamics but saturates in order to remain admissible. This saturation contributes to the difficulty of computing an explicit, large robust forward invariant set *a priori*, further motivating the approach proposed in this paper in which a conservative-but-verified safe subset is coupled with reachable set computations to achieve safety beyond $\mathcal{C}_\mathrm{b}$. Computing reachable sets in the presence of disturbances can be computationally costly and not suitable for run time implementation. Nonetheless, we show how the theory of mixed monotone dynamical systems can be leveraged for the efficient computation of robust reachable sets for (9). This is the main point of study in Section IV where we show that the closed-loop spacecraft dynamics (9) under a particularly constructed $\mathbf{u}_\mathrm{b}$ is a mixed monotone system and where we compute a decomposition function for this system. An example is provided in the same section where we demonstrate how the decomposition function is applied for the efficient computation of reachable sets.

III. Constructing a Robustly Forward Invariant Safe Terminal Set

In this section we construct a safe terminal set $\mathcal{C}_\mathrm{b}$ that is robustly forward invariant under some terminal control policy $\mathbf{u}_\mathrm{t}$ constructed jointly with $\mathcal{C}_\mathrm{b}$. Such a set and controller is generally computed offline, either analytically or via optimization techniques, and inserted into the RTA filtering mechanism as an assurance that the backup controller will lead to safety.

Definition 2 (Safe Terminal Set and Terminal Control Policy)**.** A nonempty set $\mathcal{C}_\mathrm{b}$ with accompanying feedback control policy $\mathbf{u}_\mathrm{t}$ is a *safe terminal set* and a *terminal control policy* if:

i) $\mathcal{C}_\mathrm{b} \subseteq \mathcal{C}_\mathrm{A}$,
ii) The control policy is admissible, i.e. $\mathbf{u}_\mathrm{t}(x) \in \mathcal{U}$ for all $x \in \mathcal{C}_\mathrm{b}$, and
iii) $\mathcal{C}_\mathrm{b}$ is robustly forward invariant for (9).

■

We aim to construct a safe terminal set $\mathcal{C}_\mathrm{b}$ as a zero superlevel set of some function $h_\mathrm{b}(x)$, *i.e.*, $\mathcal{C}_\mathrm{b} = \{x \in \mathcal{X} \,|\, h_\mathrm{b}(x) \geq 0\}$ for some continuously differentiable function $h_\mathrm{b} : \mathcal{X} \to \mathbb{R}$.

Proposition 1. *Given a continuously differentiable function $h_\mathrm{b}(x)$ and a control policy $\mathbf{u}_\mathrm{t}$, the set $\mathcal{C}_\mathrm{b} = \{x \in \mathcal{X} \,|\, h_\mathrm{b}(x) \geq 0\}$ is a safe terminal set, and $\mathbf{u}_\mathrm{t}$ a corresponding terminal control policy, if the following two conditions hold:*

i) For all x such that $h_\mathrm{b}(x) \geq 0$, $\varphi(x) \geq 0$ and $u_\mathrm{t}(x) \in \mathcal{U}$.
ii) For all x such that $h_\mathrm{b}(x) = 0$ and for all $w \in \mathcal{W}$, $\nabla h_\mathrm{b}(x) \neq 0$ and

$$\nabla h_\mathrm{b}(x) f(x, u_\mathrm{T}(x), w) \geq 0. \tag{14}$$

Remark 1. The above conditions imply that $\mathcal{C}_\mathrm{b} \subseteq \mathcal{C}_\mathrm{A}$, $u_\mathrm{t}(t)$ is admissible , and (14) ensures that $\mathcal{C}_\mathrm{b}$ is robustly forward invariant [30]. ■

The set $\mathcal{C}_\mathrm{b}$ should be as large as possible to give the primary controller the flexibility to operate over a wide range of states. In other words, a large $\mathcal{C}_\mathrm{b}$ will ensure that a safe backup strategy will almost always be available, and only infrequently will the primary controller find itself approaching a state where a future safe backup policy may not exist. Furthermore, a large $\mathcal{C}_\mathrm{b}$ can ease the computational workload of the RTA because a larger set generally reduces the time-horizon which must be searched to show that a safe backup policy exists.

Several strategies are available to find a set that satisfies the conditions in Proposition 1. Such strategies generally resemble a multi-objective non-convex optimization problem that can be broken into convex subproblems; iterating through these subproblems can yield a set $\mathcal{C}_\mathrm{b}$ which both satisfies Proposition 1 and allows for the system to perform satisfactorily with the RTA in the loop. An example of two competing objectives are the need to maximize the size (volume) of the set while maintaining the robustness requirement. These two objectives compete because a low-gain controller is necessary to ensure that $u(x) \in \mathcal{U}$ when $\theta(x)$ is far from the origin, but a high-gain controller is needed to produce a set that is forward invariant for all $w \in \mathcal{W}$.

Since the attitude dynamics of the spacecraft are polynomial functions, we propose using the *sum-of-squares* (SOS) optimization framework to obtain a safe terminal set $\mathcal{C}_\mathrm{b}$ that certifiably satisfies the positivity requirements given in Proposition 1. A comprehensive description of the theory of SOS optimization and its applications to control systems is given in [31], [32]. The search for an SOS polynomial can be written as a semi-definite programming (SDP) problem, for which many proven solvers exist. A related approach uses the same underlying machinery to solve a similar spacecraft control verification problem by framing it as a generalized moment problem [33]. In the following, we assume a general knowledge of SOS optimization and the tools used to translate the requirements of Proposition 1 into an SOS search problem, such as the S-Procedure [34].

A procedure to compute $\mathcal{C}_\mathrm{b}$ is now outlined. We say that a real polynomial $s(x) \in$ SOS if there exist some real polynomials $p_i(x)$ such that $s(x) = \sum_i p_i^2(x)$.

1) *Stabilize to Safety.* We first hypothesize a control policy $\mathbf{u}_\mathrm{t}(x)$ that will create a forward invariant safe region for the spacecraft system (9). To do this, we exploit structural properties of the system and design a controller with a Lyapunov certificate of stability on a bounded safe subset of the statespace, assuming the disturbance is not present. In general, any stabilizing controller can be used with the main results of this work and Step 2) below; see [35], for example, for a stabilizing linear controller for (9), which is inertia-free and in terms of the state variables ρ_1, ρ_2 and ω, and where stability is certified via a quadratic Lyapunov function. For experimental convenience, in this section, we focus specifically on the linearizing controller
$$\mathbf{u}_\mathrm{t}(x) = \omega \times J\omega + Kx, \tag{15}$$
which can tuned via the gain matrix $K \in \mathbb{R}^{3\times 5}$ in order to ensure the stability of the closed-loop dynamics of (9). To prove local stability we use a linearization of the closed-loop dynamics $\dot{x} = Ax$. We then compute a positive definite matrix P which satisfies the Lyapunov equation $A^\top P + PA + Q = 0$ for some positive definite matrix Q. In this way, $V(x) = x^\top Px$ is a Lyapunov function for the linearized system, which implies that the nonlinear dynamics in (9) are also locally exponentially stable to the origin. Furthermore, this local Lyapunov function $V(x)$ for (9) can be used to construct an $h_\mathrm{b}(x)$ which satisfies Proposition 1. In practice, the linearizing controller will be applied to the system via a monotonically increasing saturation function $\mathbf{u}_\mathrm{b}(x) = \phi(\omega \times J\omega + Kx)$ where the saturation function $\phi : \mathbb{R}^3 \to \mathcal{U}$ ensures the admissibility of $\mathbf{u}_\mathrm{b}$. In the following, we use knowledge of the input constraints $\mathcal{U}$ to construct controller gains and verify a safe controlled invariant set for (9) in the presence of disturbances using the Lyapunov function V.
2) *Specify Candidate* $\mathcal{C}_\mathrm{b}$. In this step, we establish that $\varphi(x) \geq 0$ and $\mathbf{u}_\mathrm{t}(x) \in \mathcal{U}$ are satisfied by all states in the sublevel set $V(x) \leq \gamma$ for an appropriately chosen γ. To that end, we assume $\mathcal{U}$ is of the form $[-u_{1,\max}, u_{1,\max}] \times [-u_{2,\max}, u_{2,\max}] \times [-u_{3,\max}, u_{3,\max}]$, although it is straightforward to incorporate other semialgebraic set characterizations for $\mathcal{U}$. We then perform a line search over γ in order to find the largest value of γ such that there exists $s_i(x) \in$ SOS and $\varepsilon_i \geq 0$ that satisfy
$$-\varepsilon_i - c_i(x) + s_i(x)(\gamma - V(x)) \in \text{SOS} \tag{16}$$
for all $i = 1, \ldots, 7$ where
$$\begin{aligned}&\{c_1(x), \ldots, c_7(x)\}\\ &= \{\varphi(x),\ u_{j,\max} - u_{\mathrm{T},j}(x),\ u_{j,\max} + u_{\mathrm{T},j}(x)\}\end{aligned}$$
is a set of constraints encoding the safety and control constraints. Then, we have $h(x) = \gamma - V(x)$ as a candidate level set function for defining $\mathcal{C}_\mathrm{b}$.
3) *Certify Robust Forward Invariance.* In this step, we establish that (14) is satisfied by finding a $\lambda \geq 0$, a polynomial $p(x, w)$, and SOS polynomials $s_1(x, w)$, $s_2(x, w)$, and $s_3(x, w)$ satisfying
$$\begin{aligned}&-\lambda + \dot{h}(x, w) - p(x, w)h(x) - s_1(x, w)(w_{\max}^2 - w_x^2)\\ &- s_2(x, w)(w_{\max}^2 - w_y^2) - s_3(x, w)(w_{\max}^2 - w_z^2) \in \text{SOS}.\end{aligned}$$
Then we set the resulting $h_\mathrm{b}(x)$ as $h(x)$ and take $\mathcal{C}_\mathrm{b} = \{x \in \mathcal{X} \mid h_\mathrm{b}(x) \geq 0\}$.
4) *Measure Set and Iterate.* Comparison functions can be constructed in order to describe the "volume" of the set $\mathcal{C}_\mathrm{b}$. Then, various control policies $\mathbf{u}_\mathrm{t}(x)$ can be tested to find a $\mathcal{C}_\mathrm{b}$ of a size and shape which allows for satisfactory performance of the system with RTA in-the-loop while satisfying the safety and control constraints. Examples of such an iterative procedure are given in [36].

The above procedure computes certificates that are sufficient to verify that the conditions of Proposition 1 are satisfied. Note that if an optimization problem does not generate a feasible set of decision variables, it does not necessarily mean that the conditions of the proposition are not satisfied. In some cases, a set of higher-order polynomial constraints can be specified when the second-order polynomials used in this paper do not produce results; examples are given in [32]. Additionally, the parameters ε_i and λ provide a measure of the margin by which a condition is satisfied in the worst case, which may occur at some location on the boundary of $\mathcal{C}_\mathrm{b}$.

IV. Efficient Reachability Analysis via Mixed Monotonicity

The RTA architecture proposed in this work uses mixed monotone systems theory to efficiently compute reachable set

overapproximations online. We present the basic theory of mixed monotone systems in Section IV-A. Then, in Section IV-B, we show that closed-loop backup dynamics (9) under (15) are mixed monotone and we explicitly compute a decomposition function for this system. We present an example in Section IV-C, where we demonstrate how the decomposition function for (9) enables the efficient overapproximation of reachable sets for (9).

A. Preliminaries on Mixed Monotone Systems Theory

Consider a dynamical system

$$\dot{x} = F(x, w), \tag{17}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and $w \in \mathcal{W} := [\underline{w}, \overline{w}] \subset \mathbb{R}^m$ denote the system state and a bounded time-varying disturbance input.

Definition 3 (Mixed Monotonicity [16])**.** Given a locally Lipschitz continuous function $d : \mathcal{X} \times \mathcal{W} \times \mathcal{X} \times \mathcal{W} \to \mathbb{R}^n$, the system (17) is *mixed monotone with respect to* d if for all $x, \widehat{x} \in \mathcal{X}$ and all $w, \widehat{w} \in \mathcal{W}$ all of the following hold:

1) $d(x, w, x, w) = F(x, w)$.
2) $\frac{\partial d_i}{\partial x_j}(x, w, \widehat{x}, \widehat{w}) \geq 0$, for all $i, j \in \{1, \cdots, n\}$, with $i \neq j$, whenever the derivative exists.
3) $\frac{\partial d_i}{\partial \widehat{x}_j}(x, w, \widehat{x}, \widehat{w}) \leq 0$, for all $i, j \in \{1, \cdots, n\}$ whenever the derivative exists.
4) $\frac{\partial d_i}{\partial w_j}(x, w, \widehat{x}, \widehat{w}) \geq 0$ for all $i \in \{1, \cdots, n\}$ and all $j \in \{1, \cdots, p\}$ whenever the derivative exists.
5) $\frac{\partial d_i}{\partial \widehat{w}_j}(x, w, \widehat{x}, \widehat{w}) \leq 0$ for all $i \in \{1, \cdots, n\}$ and all $j \in \{1, \cdots, p\}$ whenever the derivative exists. ■

If (17) is mixed monotone with respect to d, then d is a *decomposition function* for (17) and

$$\begin{bmatrix} \dot{x} \\ \dot{\widehat{x}} \end{bmatrix} = E(x, \widehat{x}) := \begin{bmatrix} d(x, \underline{w}, \widehat{x}, \overline{w}) \\ d(\widehat{x}, \overline{w}, x, \underline{w}) \end{bmatrix} \tag{18}$$

is the *embedding system relative to* d. An important feature of mixed monotone systems is that overapproximations of reachable sets can be efficiently computed via a single simulation of the embedding system.

Proposition 2. *[16] Let* (17) *be mixed monotone with respect to* d *and choose* $S := [\underline{x}, \overline{x}] \subset \mathcal{X}$*. If* $\Phi^E(t; (\underline{x}, \overline{x})) \in \mathcal{X} \times \mathcal{X}$ *for all* $t \in [0, T]$ *then*

$$R(t; S) \subseteq [\![\Phi^E(t; (\underline{x}, \overline{x}))]\!], \tag{19}$$

where $\Phi^E(t; s)$ *is the state of the embedding system* (18) *at time* t *when beginning at state* $s \in \mathcal{X} \times \mathcal{X}$ *at time* 0*, and where* $R(t; S)$ *is the time-*t *reachable set of* (17)*.*

Proposition 2 provides an efficient algorithm for overapproximating reachable sets for (17): a simulation of the embedding system for time-horizon t, starting from state $(\underline{x}, \overline{x})$, identifies a hyperrectangular overapproximation of $R(t; [\underline{x}, \overline{x}])$ where the largest and smallest points in the rectangular approximation are taken to be the first n and last n coordinates of the simulation endpoint $\Phi^E(t; (\underline{x}, \overline{x}))$. The main challenge in this approach, however, is in identifying a suitable decomposition function for (17); generally, a mixed monotone system will be mixed monotone with respect to many decomposition functions, however, certain decomposition functions may provide more/less conservative approximations of reachable sets than others when used with Proposition 2. This is the main point of study in [37] where the authors show that all systems of the form (17) with a locally Lipshitz continuous vector field are mixed monotone with respect to a unique *tight decomposition function* that provides a tighter approximation of reachable sets than any other decomposition function for (17).

Proposition 3. *Any system* (17) *is mixed monotone with respect to decomposition function* d *constructed elementwise according to*

$$d_i(x, w, \widehat{x}, \widehat{w}) = \begin{cases} \min\limits_{\substack{y \in [x, \widehat{x}] \\ y_i = x_i \\ z \in [w, \widehat{w}]}} F_i(y, z), & \text{if } (x, w) \preceq (\widehat{x}, \widehat{w}), \\ \max\limits_{\substack{y \in [\widehat{x}, x] \\ y_i = x_i \\ z \in [\widehat{w}, w]}} F_i(y, z), & \text{if } (\widehat{x}, \widehat{w}) \preceq (x, w). \end{cases} \tag{20}$$

Moreover, for all other decomposition functions d' *for* (17) *and any initial set* $S = [\underline{x}, \overline{x}]$*,*

$$R(t; S) \subseteq [\![\Phi^E(t; (\underline{x}, \overline{x}))]\!] \subseteq [\![\Phi^{E'}(t; (\underline{x}, \overline{x}))]\!] \tag{21}$$

where Φ^E *and* $\Phi^{E'}$ *denote the state transition functions of the embedding systems constructed from* d *and* d'*, respectively.*

We refer to the unique decomposition function constructed in (20) as the *tight decomposition function* for (17). While it is sometimes possible to attain a tight decomposition function in closed-form, as posed in (20), computing a tight decomposition function generally requires solving a nonconvex optimization problem for each quadruple $(x, w, \widehat{x}, \widehat{w})$ and the computational infeasibility of (20) implies that it is of limited direct use. For this reason, computing decomposition functions using other means can be preferable; see [24], [38] for an algorithm for computing decomposition functions for systems with uniformly bounded Jacobian matrices and see [39] for an algorithm for computing decomposition functions for systems defined by polynomial vector fields.

Computing a decomposition function for (9) is the subject of the next section.

B. Decomposition Function Construction

In the setting of the spacecraft system (9), we construct a decomposition function by viewing the closed-loop backup dynamics as an interconnection between two subsystems; one describing the time-evolution of ρ and the other describing the time-evolution of $\Omega := J\omega$, where J is the inertia matrix in (9). Studying a linearly-transformed set of the dynamics in this way enables reduced conservatism in the approximation of reachable sets using Proposition 2—as studied further in [40] and [41]—and we demonstrate this assertion later through example in Section IV-C.

Proposition 4. *The closed-loop spacecraft dynamics* (9) *under* (15) *are mixed monotone. In particular, the orientation dynamics* (6) *in* (9) *are mixed monotone with respect to a tight decomposition function attainable in closed-form.*

We sketch the proof of Proposition 4 in the following two sections where we compute individual decomposition functions for the orientation and velocity dynamics, respectively.

1) Orientation dynamics: We first show that the orientation dynamics (6) are mixed monotone with a tight decomposition function attainable in closed-form. The proof of this result is sketched below through the derivation of d^{ρ_1}, the first entry of the tight decomposition function for (6) which later becomes the first entry of a decomposition function for (9). For the purpose of this section, we write the ρ_1 dynamics as

$$\dot{x}_1 = F^{\rho_1}(x) = x_2x_5 + x_1x_2x_4 + \frac{x_3}{2}(1 + x_1^2 - x_2^2), \quad (22)$$

where we recall that $x = (\rho, \omega) \in \mathbb{R}^5$, so that, e.g., $x_1 = \rho_1$. The derivation of d^{ρ_1} is given in Table I and we elaborate on the derivation below; in Table I we use $d^{\rho_1}(x, \widehat{x})$ to denote the decomposition function, where we now omit the 2^{nd} and 4^{th} arguments of $d^{\rho_1}(x, w, \widehat{x}, \widehat{w})$ to reflect the fact that (22) contains no disturbance input.

Computing a tight decomposition function for (22) requires solving the nonconvex optimization problem (20) in closed-form, where $F(x, w)$ in (20) is taken to be the vector field $F^{\rho_1}(x)$. This optimization problem is given by (23). A key observation in our approach is that (i) each constraint in (23) depends on only one state variable, and (ii) the vector field $F^{\rho_1}(x)$ which forms the objective function in (23) is linear in x_3, x_4 and x_5. This allows us to rewrite (20) as a nested optimization problem (24), where now the outer optimization problem in (24) is evaluated over the finite set and the inner optimization problem in (24) is evaluated over the single variable y_2.

The next step in the derivation involves replacing the inner optimization problem in (24) with a closed-form expression. To do this, consider a system with quadratic polynomial dynamics

$$\dot{x} = F^{\text{Quad}}(w) = aw^2 + bw + c \quad (26)$$

where $x \in \mathbb{R}$ is the state, $w \in \mathbb{R}$ is the input and $a, b, c \in \mathbb{R}$ are fixed parameters of the vector field. Observe that (26) is mixed monotone with a tight decomposition function given in closed-form by

$$\begin{aligned} &d^{\text{Quad}}\left(w, \widehat{w}, \begin{bmatrix} a \\ b \\ c \end{bmatrix}\right) = \\ &= \begin{cases} \min\limits_{z \in [w, \widehat{w}]} & az^2 + bz + c \quad \text{if } w \leq \widehat{w}, \\ \max\limits_{z \in [\widehat{w}, w]} & az^2 + bz + c \quad \text{if } \widehat{w} \leq w. \end{cases} \quad (27) \\ &= \begin{cases} \frac{-b^2+4ac}{4a} & \text{if } \frac{-b}{2} \in [aw, a\widehat{w}], \\ F^{\text{Quad}}(w) & \text{if } \frac{-b}{2} \notin [aw, a\widehat{w}] \text{ and } 0 \leq aw + a\widehat{w} + b, \\ F^{\text{Quad}}(\widehat{w}) & \text{if } \frac{-b}{2} \notin [aw, a\widehat{w}] \text{ and } 0 \geq aw + a\widehat{w} + b, \end{cases} \end{aligned}$$

where, we omit the second and fourth input in $d^{\text{Quad}}(x, w, \widehat{x}, \widehat{w})$ to reflect the fact that $F^{\text{Quad}}(w)$ depends only on w and we include (a, b, c) as an input to d^{Quad} so that the formulation is applicable across all choices of a, b, c. Observe that the optimization problem in (27) is the same as the inner optimization in (24), when $w, \widehat{w}, z, a, b$ and c in (27) are taken to be $x_2, \widehat{x}_2, y_2, -y_3/2, y_3 + x_1y_4$ and $y_3(1 + x_1^2)/2$ in (24), respectfully. Thus, we can replace the inner optimization problem in (24) with a single evaluation of d^{Quad}, which has a closed-form solution; see (25). Evaluating d^{ρ_1} involves computing a finite optimization problem over eight evaluations of d^{Quad} and, thus, we have arrived at a tractably computable representation of the tight decomposition function for (22).

2) Angular Velocity Dynamics: We next study the closed-loop angular velocity dynamics (5) under the backup controller (15), namely,

$$\dot{\omega} = J^{-1}\left(-\omega \times J\omega + \phi(\omega \times J\omega + K_\omega\omega + K_\rho\rho) + w\right) \quad (30)$$

where, as in (15), $\phi : \mathbb{R}^3 \to \mathcal{U}$ is a monotonically increasing saturation function and $K = [K_\omega, K_\rho] \in \mathbb{R}^{3\times3} \times \mathbb{R}^{3\times2}$ are linear controller gains. For reasons that will be made clear later, we consider a linear transformation on the state space of (30) $\Omega = J\omega$ so that the dynamics of Ω are given by

$$\begin{aligned} \dot{\Omega} = F^{\Omega}(x', w) &= -(J^{-1}\Omega) \times \Omega \\ &+ \phi\left((J^{-1}\Omega) \times \Omega + K_\omega J^{-1}\Omega + K_\rho\rho\right) + w \quad (31) \end{aligned}$$

with state $x' = (\rho, \Omega)$. The purpose of this section is to construct a decomposition function for (31), as in [40], where the authors discuss how considering a linear transformation of the system dynamics can allow for reduced conservatism in the approximation of reachable sets using Proposition 2.

We sketch the construction of a decomposition function for (31) through the derivation of d^{Ω_1}, the first entry. This derivation is provided in Table II.

The first entry of the dynamics (31) is given by (28), where F^{Quad} is given by (26) and where we use the shorthand notation

$$T^1_{\Omega_1} = \begin{bmatrix} J^{-1}_{3,2} \\ J^{-1}_{3,1}\Omega_1 \\ 0 \end{bmatrix}, \qquad T^2_{\Omega_1} = \begin{bmatrix} -J^{-1}_{2,3} \\ -J^{-1}_{2,1}\Omega_1 \\ 0 \end{bmatrix}, \quad (32)$$

$$T^3_{\Omega_1} = \begin{bmatrix} -J^{-1}_{3,2} \\ -J^{-1}_{3,1}\Omega_1 + J^{-1}_{1,2}K_{1,3} + J^{-1}_{2,2}K_{1,4} + J^{-1}_{3,2}K_{1,5} \\ (J^{-1}_{1,1}K_{1,3} + J^{-1}_{2,1}K_{1,4} + J^{-1}_{3,1}K_{1,4})/2 \end{bmatrix}, \quad (33)$$

$$T^4_{\Omega_1} = \begin{bmatrix} J^{-1}_{2,3} \\ J^{-1}_{2,1}\Omega_1 + J^{-1}_{1,3}K_{1,3} + J^{-1}_{2,3}K_{1,4} + J^{-1}_{3,3}K_{1,5} \\ (J^{-1}_{1,1}K_{1,3} + J^{-1}_{2,1}K_{1,4} + J^{-1}_{3,1}K_{1,4})/2 \end{bmatrix}, \quad (34)$$

$$c = J^{-1}_{2,2} - J^{-1}_{3,3}. \quad (35)$$

Observe that the terms $T^1_{\Omega_1}$, $T^2_{\Omega_1}$, $T^3_{\Omega_1}$, and $T^4_{\Omega_1}$ are functions of Ω_1. Since (28) is constructed as the sum of functions with known decompositions, we can compute a decomposition function for (28) as the sum of the decomposition functions for the individual elements of the sum; See (29) in Table II[2].

[2]Even though (29) is constructed from tight decomposition functions for (26) and (37), the resulting decomposition function is not tight for (28). Viewing the vector field as a *sum*, in this way, is known to lead to non-tight decomposition functions and, in general, it is preferable to minimize the number of sum elements, when possible, to attempt to minimize the conservatism. This is the main reason for considering a transformation of the dynamics, i.e., considering (31) rather than (5), as the transformed dynamics are constructed via sums of fewer elements and therefore yields a tighter decomposition function using this approach.

TABLE I: Derivation of the first entry of decomposition function

$$d^{\rho_1}(x,\widehat{x}) = \begin{cases} \min\limits_{\substack{y\in[x,\widehat{x}]\\ y_1=x_1}} \; y_2y_5 + y_1y_2y_4 + \frac{y_3}{2}(1+y_1^2-y_2^2) & \text{if } x \preceq \widehat{x}, \\ \max\limits_{\substack{y\in[\widehat{x},x]\\ y_1=x_1}} \; y_2y_5 + y_1y_2y_4 + \frac{y_3}{2}(1+y_1^2-y_2^2) & \text{if } \widehat{x} \preceq x. \end{cases} \tag{23}$$

$$= \begin{cases} \min\limits_{y_{3:5}\in\langle\!\langle x_{3:5},\widehat{x}_{3:5}\rangle\!\rangle} \; \min\limits_{y_2\in[x_2,\widehat{x}_2]} \; (-\frac{y_3}{2})y_2^2 + (y_5+x_1y_4)y_2 + \frac{y_3}{2}(1+x_1^2) & \text{if } x \preceq \widehat{x}, \\ \max\limits_{y_{3:5}\in\langle\!\langle \widehat{x}_{3:5},x_{3:5}\rangle\!\rangle} \; \max\limits_{y_2\in[\widehat{x}_2,x_2]} \; (-\frac{y_3}{2})y_2^2 + (y_5+x_1y_4)y_2 + \frac{y_3}{2}(1+x_1^2) & \text{if } \widehat{x} \preceq x. \end{cases} \tag{24}$$

$$= \begin{cases} \min\limits_{y_{3:5}\in\langle\!\langle x_{3:5},\widehat{x}_{3:5}\rangle\!\rangle} \; d^{\mathrm{Quad}}\left(y_2,\widehat{y}_2,\begin{bmatrix} -y_3/2 \\ y_5+x_1y_4 \\ y_3(1+x_1^2)/2 \end{bmatrix}\right) & \text{if } x \preceq \widehat{x}, \\ \max\limits_{y_{3:5}\in\langle\!\langle \widehat{x}_{3:5},x_{3:5}\rangle\!\rangle} \; d^{\mathrm{Quad}}\left(y_2,\widehat{y}_2,\begin{bmatrix} -y_3/2 \\ y_5+x_1y_4 \\ y_3(1+x_1^2)/2 \end{bmatrix}\right) & \text{if } \widehat{x} \preceq x. \end{cases} \tag{25}$$

TABLE II: Derivation of the third entry of decomposition function

$$\dot{\Omega}_1 = F^{\mathrm{Quad}}\left(\Omega_2, T^1_{\Omega_1}\right) + F^{\mathrm{Quad}}\left(\Omega_3, T^2_{\Omega_1}\right) - c\,\Omega_2\Omega_3 + \phi\left(F^{\mathrm{Quad}}\left(\Omega_2, T^3_{\Omega_1}\right) + F^{\mathrm{Quad}}\left(\Omega_3, T^4_{\Omega_1}\right) + c\,\Omega_2\Omega_3 + K^{\rho}_{1,:}\rho\right) + w_1 \tag{28}$$

$$d^{\Omega_1}(x,w,\widehat{x},\widehat{w}) = d^{\mathrm{Quad}}\left(\Omega_2,\widehat{\Omega}_2,T^1_{\Omega_1}\right) + d^{\mathrm{Quad}}\left(\Omega_3,\widehat{\Omega}_3,T^2_{\Omega_1}\right) + d^{\mathrm{Mult}}\left(\Omega_{2:3},\widehat{\Omega}_{2:3},-c\right) +$$
$$+ \phi\left(d^{\mathrm{Quad}}\left(\Omega_2,\widehat{\Omega}_2,T^3_{\Omega_1}\right) + d^{\mathrm{Quad}}\left(\Omega_3,\widehat{\Omega}_3,T^4_{\Omega_1}\right) + d^{\mathrm{Mult}}\left(\Omega_{2:3},\widehat{\Omega}_{2:3},c\right) + [K^{\rho}_{1,:}]^{+}\rho + [K^{\rho}_{1,:}]^{-}\widehat{\rho}\right) + w_1 \tag{29}$$

In Table II, we denote by

$$d^{\mathrm{Mult}}(w,\widehat{w},a) = \begin{cases} \min_{z\in\langle\!\langle w,\widehat{w}\rangle\!\rangle} a\,z_1z_2 & \text{if } w \preceq \widehat{w}, \\ \max_{z\in\langle\!\langle \widehat{w},w\rangle\!\rangle} a\,z_1z_2 & \text{if } \widehat{w} \preceq w, \end{cases} \tag{36}$$

the tight decomposition function for the mixed monotone system

$$\dot{x} = F^{\mathrm{Mult}}(w) = a\,w_1w_2, \tag{37}$$

where $x \in \mathbb{R}$ is the state, $w \in \mathbb{R}^2$ is the input, and a is a fixed parameter of the vector field. Moreover, since (29) contains only optimization problems over finite sets, we have arrived at a tractably computable decomposition function for (28).

C. Application of Mixed Monotonicity in Spacecraft Attitude Control

Combining the subsystem decomposition functions for (6) and (31) requires rectifying the fact that the two systems contain different state variables, that is, (6) depends on $x = (\rho,\omega)$ whereas (31) depends on $x' = (\rho,\Omega)$. For this reason, in our work, we construct a decomposition function for (9) using a slight modification of the decomposition function d^{ρ_1} from (25), modified to account for the fact that the state vector has changed to include Ω. Nonetheless, the resulting dynamics (6) are still linear in $\Omega = J\omega$ and thus the resulting decomposition function used in this work is still tight for (6).

Letting $d^{\rho}(x',\widehat{x}')$ denote this decomposition function, we have that $\dot{x}' = F(x',w)$ is mixed monotone with respect to

$$d(x',w,\widehat{x}',\widehat{w}) = \begin{bmatrix} d^{\rho}(x',\widehat{x}') \\ d^{\Omega}(x',w,\widehat{x}',\widehat{w}) \end{bmatrix}, \tag{38}$$

and Proposition 2 now implies that the reachable set of (6)–(31) from state $x' \in \mathbb{R}^5$ is efficiently overapproximated using

$$R(t;x',\mathbf{u}_{\mathrm{b}}) \subseteq [\![\Phi^E(t;(x',x'))]\!], \tag{39}$$

where Φ^E is the state transition function of the embedding system (18), taken with respect to d, and $R(t;x',\mathbf{u}_{\mathrm{b}})$ is the reachable set of (6)–(31) under $\mathbf{u}_{\mathrm{b}}$ as in (10). Equivalently, for all $\mathbf{w} : [0,t] \to \mathcal{W}$ we have that

$$(\rho, J\omega) \in [\![\Phi^E(t;(x,x))]\!] \tag{40}$$

where $(\rho,\omega) = \Phi(t;x,\mathbf{u}_{\mathrm{b}},\mathbf{w})$ and $x \in \mathcal{X}$ is the initial state. Furthermore, defining by

$$R^{\theta}(t;x',\mathbf{u}_{\mathrm{b}}) := \{\theta(y) \,|\, y \in R(t;x',\mathbf{u}_{\mathrm{b}})\}, \tag{41}$$

the system's reachable set in the space of line-of sight angles, we have that

$$R^{\theta}(t;x',\mathbf{u}_{\mathrm{b}}) \subseteq \Theta\left([\![\Phi^E(t;(x',x'))]\!]\right), \tag{42}$$

where $\Theta(\,\cdot\,)$ is an inclusion function for $\theta(\,\cdot\,)$ as defined in [42]; see also [41]. A demonstration is provided in Figure 4 where we apply (42) in order to overapproximate $R^{\theta}(t;x',\mathbf{u}_{\mathrm{b}})$ for $t \in [0,15]$ and where

$$x = \left(\frac{1}{\sqrt{3}}, 0, 0.1, 0.1, 0.1\right), \tag{43}$$

that is, $\theta(x) = 60°$ and $\omega_0 = (0.1, 0.1, 0.1)$ rad/s.

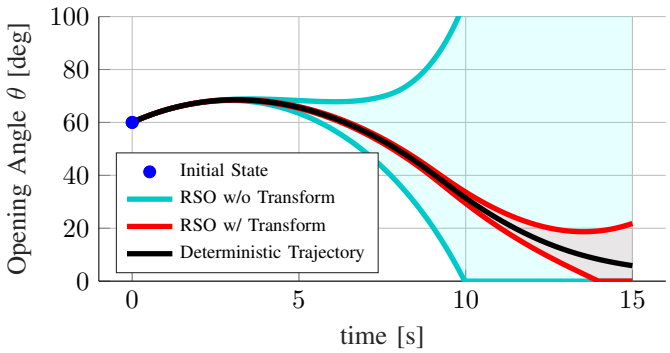

Fig. 4: Demonstration of the reachability procedure posited in (42). The initial state x given by (43) is shown in blue, and the deterministic trajectory $\theta(\Phi(t; x, \mathbf{u}_\mathrm{b}, 0))$, arising from the case with no disturbances, is shown in black. The over-approximation of $R^\theta(t; x', \mathbf{u}_\mathrm{b})$ attained from applying (42) with the decomposition function (38) for the transformed dynamics (6)–(31) is shown in red. As a comparison, we show also a similar reachable set approximation, attained from a decomposition function for the untransformed dynamics in (9). This second approximation, which is much more conservative, is shown in blue.

V. Hardware Demonstration of Assured Safety

We demonstrate the proposed framework for assured safety with an experiment at the **A**utonomous **S**pacecraf**t** **R**obotic **O**perations in **S**pace (ASTROS) experimental facility [43], located at the Dynamics and Control Systems Laboratory of the Georgia Institute of Technology.

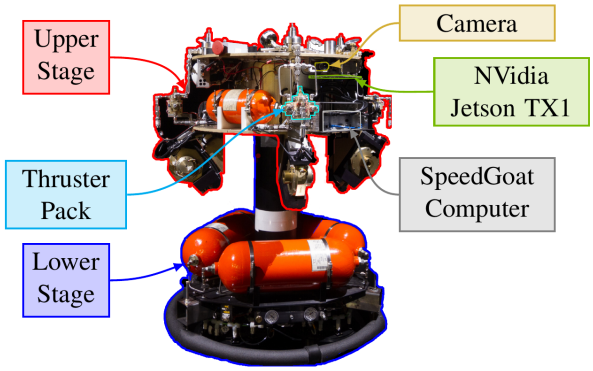

Fig. 5: Main Components of ASTROS Experimental Test-Bed

The ASTROS platform is composed of two structures, namely, the upper and lower stages. The motion of these two stages is restricted or rendered free by exploiting two pressurized-air bearing systems, allowing for frictionless motion in up to 5 degrees of freedom, 3 of which are of rotation and 2 of translation. A hemispherical air-bearing situated between the lower and upper stages allows for free rotation of the upper stage around two perpendicular horizontal axes. Additionally, a linear air-bearing system between the lower stage and the floor levitates its lower stage off the near-perfectly flat floor, providing two degrees of planar translation plus one degree of rotation (2+1 configuration), although this mode of motion was fixed during our experiment. The platform is fitted with 12 cold-air gas thrusters in a 3-3-3-3 pack configuration, which, when fired, generate forces and torques to allow it to actively maneuver in the test arena.

The ASTROS test-bed also possesses an inertial measurement unit and a rate gyro, which when paired with an extended Kalman filter (EKF), allow it to estimate the position, attitude, linear velocity and angular velocity of the upper stage. Additionally, a 12-camera VICON™ motion capture system provides accurate position and orientation measurements of the ASTROS upper stage at a rate of 100Hz, which are also incorporated into the EKF.

The actuation of the thrusters is performed by dedicated power electronics in response to control signals computed onboard using an embedded SpeedGoat™ computer. The computer compiles and executes a program derived from a prototyped Simulink™ model incorporating sensor measurement acquisition, control computation, actuator allocation and input-output communications with devices on the platform in real-time. A linear program leveraging GLPK [44] runs in real-time and allocates control to the 12 thrusters such that the resulting torque equals the requested torque control computed by our controller while minimizing pressurized air (fuel) usage. The 12 allocated control values are then emulated via a pulse width modulation (PWM) scheme, thus generating thruster on-off commands which are executed at cycle rate of 10Hz with a duty cycle resolution of 0.01s.

The experiments were conducted with an admissible control set $\mathcal{U} = [-2, 2] \times [-0.5, 0.5]^2$ Nm and considered disturbance torques in $\mathcal{W} = [-0.02, 0.02]^3$ Nm. Also, the inertia matrix of the spacecraft was considered to be

$$J = \begin{bmatrix} 17.5 & -0.8 & 0.3 \\ -0.8 & 14.9 & 0.4 \\ 0.3 & 0.4 & 20.8 \end{bmatrix} \ \mathrm{kg\,m^2}. \tag{44}$$

We first design a controlled forward invariant set via the procedure in Section III. We use the control policy (15) resulting from the solution to a Linear Quadratic Regulator (LQR) problem involving the linearized dynamics. In particular, we consider a backup controller

$$\mathbf{u}_\mathrm{b}(x) = \phi\left(\omega \times J\omega + Kx\right), \tag{45}$$

where the term $\phi : \mathbb{R}^3 \to \mathcal{U}$ is a saturation function, included in order to ensure the admissibility of $\mathbf{u}_\mathrm{b}$. An LQR problem is solved in order to penalize torque control effort along axes that have smaller actuation limits. This approach was iterated over using various cost functions in order to find a controller that would permit the largest possible set that is both robust and that does not saturate the actuators. The gain used is

$$K = 10^{-2} \times \begin{bmatrix} 100.0 & -0.4 & 467.3 & 3.5 & -1.4 \\ 0.3 & 70.6 & 5.1 & 330.8 & 4.2 \\ 1.4 & -5.9 & 4.0 & -20.0 & 99.8 \end{bmatrix}. \tag{46}$$

The resulting closed-loop linear system is then used to com-

pute a Lyapunov function $V(x) = x^\top P x$ with

$$P = \begin{bmatrix} 20.0 & 2.1 & 6.5 & 1.1 & -1.5 \\ 2.1 & 26.7 & 1.1 & 10.0 & -2.4 \\ 6.5 & 1.1 & 323.3 & 17.6 & -2.8 \\ 1.1 & 10.0 & 17.6 & 378.5 & -4.4 \\ -1.5 & -2.4 & -2.8 & -4.4 & 288.1 \end{bmatrix}. \quad (47)$$

The set $\mathcal{C}_\mathrm{b}$ was computed via the procedure outlined in Section III, and optimal sets of the decision variables were computed for $\gamma = 17.1$ using second-order polynomial and SOS multipliers. The set $\mathcal{C}_\mathrm{b}$ is therefore defined with $h_\mathrm{b}(x) = \gamma - V(x)$.

In the experiment, a 5° avoid cone is considered ($\theta_{\max} = 175°$). As shown in Figure 6, the avoid cone is constructed in such a way that prevents an onboard laserbeam from entering a ring placed approximately 18 feet away from the vehicle's center of rotation. The desired control input to the vehicle is provided by a human-controlled pilot joystick. The user attempts to orient the vehicle in such a way that the laserbeam enters the ring, however, the RTA mechanism is effective at preventing the safety constraint from being violated. That is, the user has full control of the vehicle when the RTA is inactive, and the RTA activates only as necessary to prevent violations of the line-of-sight constraint (12).

In particular, unsafe primary control actions are chosen by a human operator via a joystick with a controller update rate of 10 Hz. A $t_\mathrm{p} = 0.6$ sec reachable set computation is conducted assuming a zero-order hold for this duration on the human operator's command control action, followed by a $t_\mathrm{b} = 15$ sec reachability computation under the backup controller. In the reachability computations, we explicitly compute reachable set overapproximations at each increment of 0.1 sec along the system's trajectory; that is, six reachable set overapproximations are computed along the $t_\mathrm{p} = 0.6$ sec application of the performance controller and $k^* = 150$ reachable set overapproximations are computed along the $t_\mathrm{b} = 15$ sec application of the backup controller. On average, computing all 156 reachable set overapproximations of the system takes about 1 ms. Therefore, the RTA filter logic computation is about two orders of magnitude faster than the controller update rate. When necessary to avoid collisions, the RTA mechanism applies the backup controller to the system in order to guarantee system safety.

Results from the experiment are provided in Figures 7 and 8. Observe that, while the human operator commands unsafe control actions via the joystick, the RTA mechanism applies the backup control input when necessary in order to ensure the satisfaction of the safety constraint for all time. Furthermore, the angle $\theta(x) \leq 175$ deg, as required, for all states $x(t)$ along the system trajectories. A video of the experiment is available at `https://youtu.be/g1-zMepDm1I`.

VI. Conclusion

We developed and demonstrated an RTA filter for a torque-controlled spacecraft in free rotational motion subject to disturbances and a line-of-sight safety constraint. To design the RTA filter, we first compute a terminal set that is contained within the safety constraint set along with a feedback control policy that renders this set robustly forward invariant. We then propose an RTA mechanism that, given a desired control policy, determines if the desired control can be safely applied to the system. Because the a priori computed terminal set is generally conservative, the RTA mechanism allows the system to evolve beyond this set provided a safe return is possible. In turn, this is ensured by computing an overapproximation of the reachable set of states obtained by applying the desired controller followed by a backup control strategy. Reachable sets are efficiently overapproximated as hyperrectangules using the theory of mixed monotone systems and include system disturbances in the computation. The key innovation of our approach is to couple efficient online reachable set computations with a safe-but-conservative terminal set that is computed offline. Applying this methodology to an underactuated, five-dimensional spacecraft hardware platform required innovations in mixed monotone systems theory to allow for efficient, provably correct reachable set approximations that are computable online.

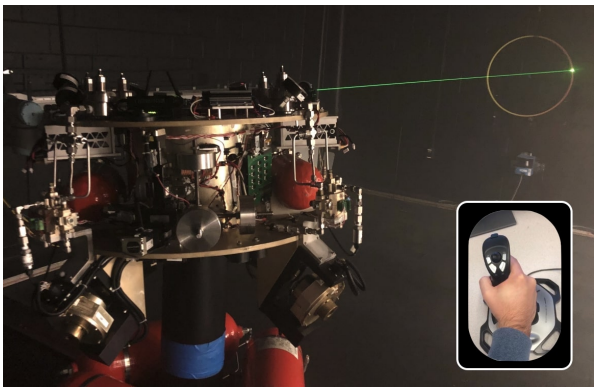

Fig. 6: Experimental Setup: The inertially-fixed $\widehat{K}$ vector is chosen to point toward a ring, with the border of the ring describing the extent of the unsafe set. A laser pointer is fixed to the body-fixed $\widehat{k}$ direction, so that the laser touches the ring when $\theta(x) = 175°$. Nominal, possibly unsafe, control actions are suggested by a human operator via a joystick.

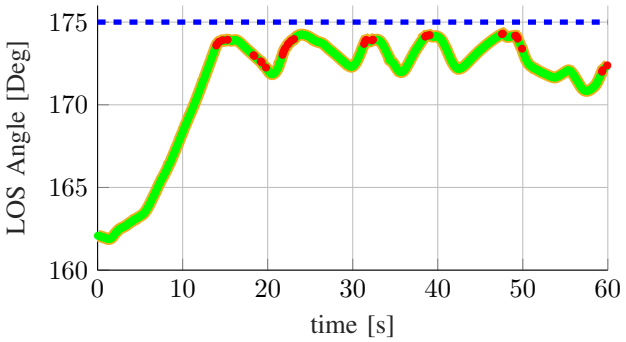

Fig. 7: Line-of-sight angle versus time. Green color indicates time iterations that are unfiltered and red color indicates times when the RTA filter is active. The extent of the admissible set, $\theta_{\max} = 175°$, is shown dashed in blue color.

References

[1] K. Hobbs, M. Mote, M. Abate, S. Coogan, and E. Feron, "Run Time Assurance for Safety-Critical Systems: An Introduction to Safety

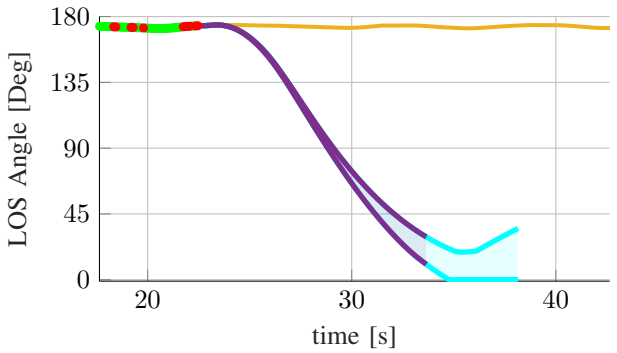

Fig. 8: In-the-loop reachability analysis: At time $t = 22.5$ sec, the backup controller is applied due to a violation of the safety constraint. The $t_{\mathrm{p}} + t_{\mathrm{b}} = 15.6$ sec reachable set prediction is shown in light blue color. If the backup controller is applied, the system is guaranteed to reenter the backup region $\mathcal{C}_{\mathrm{b}}$ in 11 sec, as shown in purple color.

Filtering Approaches for Complex Control Systems," *arXiv preprint arXiv:2110.03506*, 2021.

[2] M. L. Mote, *Optimization-based Approaches to Safety-Critical Control with Applications to Space Systems*. PhD thesis, Georgia Institute of Technology, 2021.

[3] K. L. Hobbs, *Elicitation and Formal Specification of Run Time Assurance Requirements for Aerospace Collision Avoidance Systems*. PhD thesis, Georgia Institute of Technology, 2020.

[4] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Natural Motion-based Trajectories for Automatic Spacecraft Proximity Operation Collision Avoidance ," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[5] J. G. Rivera, A. A. Danylyszyn, C. B. Weinstock, L. R. Sha, and M. J. Gagliardi, "An architectural description of the simplex architecture.," tech. rep., Carnegie-Mellon University Software Engineering Institute, Pittsburgh, PA, 1996.

[6] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The Simplex Architecture for Safe Online Control System Upgrades," in *Proceedings of the American Control Conference*, vol. 6, pp. 3504–3508, 1998.

[7] U. Mehmood, S. Bak, S. A. Smolka, and S. D. Stoller, "Safe CPS from Unsafe Controllers," *arXiv preprint arXiv:2102.12981*, 2021.

[8] O. Sanni, M. Mote, D. Delahaye, M. Gariel, T. Khamvilai, E. Feron, and S. Saber, "Ariadne: A Common-sense Thread for Enabling Provable Safety in Air Mobility Systems with Unreliable Components," 2021.

[9] S. Prajna, "Barrier Certificates for Nonlinear Model Validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[10] 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, pp. 3861–3876, Aug 2017.

[11] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European Control Conference*, pp. 3420–3431, 2019.

[12] C. Llanes, M. Abate, and S. Coogan, "Safety from in-the-loop reachability for cyber-physical systems," in *Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems (CAADCPS)*, pp. 9–10, 2021.

[13] C. Llanes, M. Abate, and S. Coogan, "Safety from fast, in-the-loop reachability with application to UAVs," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 127–136, IEEE, 2022.

[14] M. Abate and S. Coogan, "Enforcing Safety at Runtime for Systems with Disturbances," in *59th IEEE Conference on Decision and Control*, pp. 2038–2043, 2020.

[15] M. Abate, M. Mote, E. Feron, and S. Coogan, "Verification and runtime assurance for dynamical systems with uncertainty," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pp. 1–10, 2021.

[16] S. Coogan, "Mixed Monotonicity for Reachability and Safety in Dynamical Systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 5074–5085, 2020.

[17] J. Gouzé and K. Hadeler, "Monotone Flows and Order Intervals," *Nonlinear World*, vol. 1, pp. 23–34, 1994.

[18] D. Angeli, G. A. Enciso, and E. D. Sontag, "A Small-gain Result for Orthant-monotone Systems Under Mixed Feedback," *Systems & Control Letters*, vol. 68, pp. 9 – 19, 2014.

[19] G. Enciso, H. Smith, and E. Sontag, "Nonmonotone systems decomposable into monotone systems with negative feedback," *Journal of Differential Equations*, vol. 224, no. 1, pp. 205 – 227, 2006.

[20] L. Yang, O. Mickelin, and N. Ozay, "On sufficient conditions for mixed monotonicity," *IEEE Transactions on Automatic Control*, vol. 64, pp. 5080–5085, Dec 2019.

[21] S. Coogan and M. Arcak, "Stability of Traffic Flow Networks with a Polytree Topology," *Automatica*, vol. 66, pp. 246–253, Apr. 2016.

[22] H. L. Smith, "The Discrete Dynamics of Monotonically Decomposable Maps," *Journal of Mathematical Biology*, vol. 53, no. 4, p. 747, 2006.

[23] S. Coogan and M. Arcak, "Efficient Finite Abstraction of Mixed Monotone Systems," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pp. 58–67, 2015.

[24] P.-J. Meyer, A. Devonport, and M. Arcak, "TIRA: Toolbox for Interval Reachability Analysis," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pp. 224–229, 2019.

[25] P. Meyer and D. V. Dimarogonas, "Hierarchical Decomposition of LTL Synthesis Problem for Nonlinear Control Systems," *IEEE Transactions on Automatic Control*, vol. 64, pp. 4676–4683, Nov 2019.

[26] M. D. Shuster *et al.*, "A Survey of Attitude Representations," *Navigation*, vol. 8, no. 9, pp. 439–517, 1993.

[27] P. Tsiotras and J. M. Longuski, "A New Parameterization of the Attitude Kinematics," *Journal of the Astronautical Sciences*, vol. 43, no. 3, pp. 243–262, 1995.

[28] P. Tsiotras, M. Corless, and J. Longuski, "A Novel Approach to the Attitude Control of Axisymmetric Spacecraft," *Automatica*, vol. 31, no. 8, pp. 1099–1112, 1995.

[29] J. M. Brewer and P. Tsiotras, "Partial Attitude Synchronization for Networks of Underactuated Spacecraft," *Automatica*, vol. 97, pp. 27–37, 2018.

[30] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, vol. 78. Springer, 2008.

[31] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

[32] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard, "Control Applications of Sum of Squares Programming," in *Positive Polynomials in Control*, pp. 3–22, Springer, 2005.

[33] D. Henrion, M. Ganet-Schoeller, and S. Bennani, "Measures and LMI for Space Launcher Robust Control Validation," *IFAC Proceedings Volumes*, vol. 45, no. 13, pp. 236–241, 2012.

[34] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of *Studies in Applied Mathematics*. Philadelphia, PA: SIAM, June 1994.

[35] P. Tsiotras, "Further passivity results for the attitude control problem," *IEEE Transactions on Automatic Control*, vol. 43, no. 11, pp. 1597–1600, 1998.

[36] W. Tan and A. Packard, "Stability Region Analysis using Sum of Squares Programming," in *American Control Conference*, pp. 6–pp, 2006.

[37] M. Abate, M. Dutreix, and S. Coogan, "Tight Decomposition Functions for Continuous-Time Mixed-Monotone Systems With Disturbances," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 139–144, 2021.

[38] M. Khajenejad and S. Z. Yong, "Tight Remainder-Form Decomposition Functions with Applications to Constrained Reachability and Interval Observer Design," *arXiv preprint arXiv:2103.08638*, 2021.

[39] M. Abate and S. Coogan, "Computing Robustly Forward Invariant Sets for Mixed-Monotone Systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 4553–4559, 2020.

[40] M. Abate and S. Coogan, "Improving the fidelity of mixed-monotone reachable set approximations via state transformations," in *2021 American Control Conference (ACC)*, pp. 4674–4679, IEEE, 2021.

[41] M. Abate and S. Coogan, "Decomposition Functions for Interconnected Mixed Monotone Systems," *IEEE Control Systems Letters*, vol. 6, pp. 2120–2125, 2022.

[42] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, "Interval Analysis," in *Applied Interval Analysis*, pp. 11–43, Springer, 2001.

[43] D.-M. Cho, D. Jung, and P. Tsiotras, "A 5-DOF experimental platform for spacecraft rendezvous and docking," in *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference*, (Seattle, WA), p. 1869, AIAA, April 2009.

[44] N. R. Salgueiro Filipe, *Nonlinear Pose Control and Estimation for Space Proximity Operations: An Approach based on Dual Quaternions*. PhD thesis, Georgia Institute of Technology, 2014.