

Guaranteed safe satellite guidance and navigation using reachability based switching controllers

Kristina Miller*, Sean Phillips+, Sayan Mitra*

*University of Illinois Urbana-Champaign

+Air Force Research Laboratory

Abstract—The safety of satellites is an increasingly difficult requirement as launches of new satellites increase the clutter of space environments. The deployment of new, experimental controllers is important to increase the autonomous capabilities of satellites but may be at odds with safety. In this work, we consolidate these two goals by synthesizing a formally safe controller and a *runtime assurance* logic that can switch between the safety and experimental controllers to guarantee the safe operation of a satellite. This switching logic leverages reachable and recoverable sets. We deploy the synthesized safety controller and switching logic in a close-proximity scenarios with both static and dynamic obstacles and show that the satellite remains safe.

I. INTRODUCTION

Capabilities of autonomous satellites are critical as space environments are ever-changing and hazardous and the ability for on-earth satellite operators to make decisions in real-time is hindered by communication lag. Furthermore, the safety and optimization of fuel usage for these satellites is essential as the longevity of the satellite’s ability to perform a mission can drastically reduce long-term costs of satellite missions [1]. Deployment of experimental controllers which can handle harsh environments while optimizing fuel usage is thus imperative. There is a large body of related research in fields such as learning-based control [2]–[4], model predictive control [5]–[9], or artificial intelligence [10], [11] to construct such controllers. However, these controllers do not formally guarantee safety.

Controller synthesis creates controllers with formal guarantees but may not be optimal. The relevant research question is *how can we leverage optimality from the experimental controller with the safety guarantees of the safe controller?* The field of *runtime assurance* (RTA) addresses this via a switching logic that switches between the experimental and safety controller to maintain safety while still using the experimental controller whenever possible, as seen in Figure 1.

There are two main designs of a RTA system. Explicit switching mechanisms choose between a proposed input from the experimental controller and the input from the safety controller. In such systems, trajectories of the satellite are simulated over some time. If the simulations are unsafe, the safe input is chosen. If not, then the experimental input is used. A variety of techniques for simulating trajectories exist. A non-exhaustive list includes single trajectory [12], forwards reachability [13]–[15], and backwards reachability [16].

The current state of the art is *active set invariance filtering* (ASIF). Here, control barrier functions (CBFs) are used to filter the proposed input to a guaranteed safe input. ASIF is shown to work in linear [17], [18], nonlinear [19], [20], and multi-agent [21] scenarios. In practice, CBFs can be difficult to compute and recompute in real time.

In this work, we propose an RTA algorithm called *Polaris* after the northern guide star that sailors used for safe navigation. *Polaris* uses an explicit switching mechanism to switch between

Approved for public release; distribution is unlimited. Public Affairs approval # AFRL-2023-5015

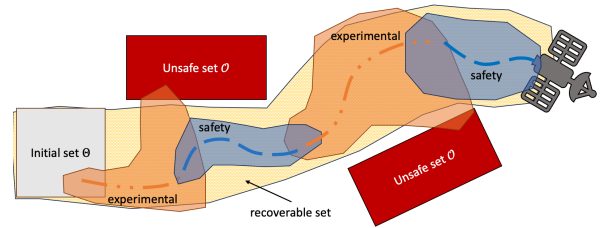


Fig. 1: Example satellite RTA. Safety is maintained by switching between experimental and safety controllers. *Polaris* checks if the reachable sets of the experimental controller (orange) exit a recoverable set (yellow).

a user-provided experimental controller and a synthesized safety controller. The safety controller consists of a reference trajectory and tracking controller. A safety envelope (*recoverable set*) is constructed about the reference trajectory using stability analysis from control theory. The reachable set of the satellite using the experimental controller is also computed. If this reachable set exits the safety envelope, the safety controller is used. *Polaris* guarantees the safety of the system (Theorem 2), as well as reduces the computational complexity from ASIF while being less conservative and safer than the naïve reachability approach. In Section III, we synthesize a safety controller consisting of a reference trajectory and a tracking controller. A *recoverable set* is then constructed about the reference trajectory using error-bound information from the tracking controller. In Section IV, we present *Polaris* and show that the synthesized safety controller and switching logic guarantees safety of the system.

Notation: The set of real numbers is denoted by \mathbb{R} , and the set of non-negative real numbers is denoted by $\mathbb{R}_{\geq 0}$. The empty set is denoted by \emptyset . Given an n -dimensional vector $x \in \mathbb{R}^n$, the 1-norm is denoted by $|x|$ and the 2-norm is denoted by $\|x\|$. Given a set $A \subset \mathbb{R}^n$, the over-approximation of A is given by $\bar{A} \supseteq A$ and the under approximation of A is given by $\underline{A} \subseteq A$. Consider a set $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^m$, $m \leq n$, the projection of some vector $x \in A$ onto B is given by $x \downarrow B$. A polytope is given by $\text{Poly}(H, b) = \{x \in \mathbb{R}^n | Hx \leq b\}$ where $H \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The number of rows in H is denoted by $\text{dP}(H) = m$. Given a scalar $x \in \mathbb{R}$, the ceiling of x is denoted by $\lceil x \rceil$.

II. PROBLEM STATEMENT

A *linear control system* $\mathcal{A} = \langle (A, B, D), \mathcal{X}, \mathcal{U}, \mathcal{W} \rangle$ is defined by its (i) state space $\mathcal{X} \subseteq \mathbb{R}^n$, (ii) input space $\mathcal{U} \subseteq \mathbb{R}^m$ which satisfies $|u_i| < u_{i, \max}$ for $i \in \{1, \dots, m\}$, $u_{i, \max} \geq 0$, (iii) disturbance space $\mathcal{W} \subseteq \mathbb{R}^p$ which satisfies $\|w\| \leq w_{\max}$ for some $w_{\max} > 0$ for every $w \in \mathcal{W}$, and (iv) system matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $D \in \mathbb{R}^{n \times p}$. The system state evolves according to $\dot{x} = Ax + Bu + Dw$ where $x \in \mathcal{X}$, $u \in \mathcal{U}$, and $w \in \mathcal{W}$. Given upper time bound $T \geq 0$, initial state $x_0 \in \mathcal{X}$, and controller $g : \mathcal{X} \times [0, T] \rightarrow \mathcal{U}$, a *trajectory* of \mathcal{A} is given by $\xi_{x_0, g} : [0, T] \rightarrow \mathcal{X}$ and satisfies $\xi_{x_0, g}(0) = x_0$, and for all $t \in [0, T]$, $\frac{d}{dt} \xi_{x_0, g}(t) = A\xi_{x_0, g}(t) + Bg(\xi_{x_0, g}(t), t) + Dw^t$ where $w^t \in \mathcal{W}$ for all $t \in [0, T]$.

Consider *safety controller* $S : \mathcal{X} \times [0, T] \rightarrow \mathcal{U}$ and *experimental controller* $U : \mathcal{X} \times [0, T] \rightarrow \mathcal{U}$. We introduce a *mode space* $\mathcal{M} = \{0, 1\}$, where 0 corresponds to the use of S and 1 to U . The system dynamics are augmented to

$$\dot{x} = f(x, t, m) = \begin{cases} Ax + BS(x, t) + Dw^t & m = 0 \\ Ax + BU(x, t) + Dw^t & m = 1 \end{cases}$$

for all $t \in [0, T]$, $x \in \mathcal{X}$, and $w^t \in \mathcal{W}$. Let $\mathcal{L} : [0, T] \rightarrow \mathcal{M}$ be a *switching logic*. Given initial state $x_0 \in \mathcal{X}$, the trajectory of \mathcal{A} using \mathcal{L} is $\xi_{x_0, \mathcal{L}}^{\{s, u\}} : [0, T] \rightarrow \mathcal{X}$ and satisfies $\xi_{x_0, \mathcal{L}}^{\{s, u\}}(0) = x_0$ and $\frac{d}{dt} \xi_{x_0, \mathcal{L}}^{\{s, u\}}(t) = f(\xi_{x_0, \mathcal{L}}^{\{s, u\}}(t), t, \mathcal{L}(t))$.

Systems in operation must typically avoid an *unsafe set* $\mathcal{O}^t \subseteq \mathcal{X}$ at every time $t \in [0, T]$. The collection of unsafe sets over $[0, T]$ is \mathcal{O} where $\mathcal{O}^t \in \mathcal{O}$ for every $t \in [0, T]$. A trajectory $\xi : [0, T] \rightarrow \mathcal{X}$ is considered *safe* if $\xi(t) \notin \mathcal{O}^t$ for every $t \in [0, T]$. Then our problem is the following. Given system \mathcal{A} , *initial set* Θ from which \mathcal{A} starts, unsafe sets \mathcal{O} , experimental controller U , and upper time bound $T > 0$, synthesize safety controller $S : \mathcal{X} \times [0, T] \rightarrow \mathcal{U}$ and switching logic $\mathcal{L} : [0, T] \rightarrow \mathcal{M}$ such that for every $x_0 \in \Theta$, the resulting $\xi_{x_0, \mathcal{L}}^{\{s, u\}}$ is safe.

Example 1 (Example: satellite control for ARPOD missions). *In satellite autonomous rendezvous, proximity operations, and docking (ARPOD) problems, a chaser satellite must perform some operations relative to a target satellite. In such scenarios, the state of the chaser relative to the target can be written in Hill's frame which is centered on the body of the target. The relative state of the chaser is given by the relative position and velocity such that $x = [r_x \ r_y \ r_z \ v_x \ v_y \ v_z]^T$; the input is given by the thrusts such that $u = [F_x \ F_y \ F_z]^T$; and the disturbances are given by $w = [w_x \ w_y \ w_z]$. The relative dynamics follow the Clohessy-Wiltshire equations $\dot{x} = Ax + Bu + Dw$ where*

$$A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \\ 3\eta^2 & 0 & 0 & 0 & 2\eta & 0 \\ 0 & 0 & 0 & -2\eta & 0 & 0 \\ 0 & 0 & -\eta^2 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix}, \quad \text{and } D = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{I}_3 \end{bmatrix}. \quad (1)$$

Here, $\mathbf{0}_{3 \times 3}$ is a 3 by 3 zero matrix and \mathbf{I}_3 is a 3 by 3 identity matrix; and η is the mean motion parameter, typically $\eta = 0.0012\text{m/s}^2$ for low earth orbits, and m is the mass of the satellite, which we will set to be $m = 10\text{kg}$ for a CubeSat.

As a running example, we consider the scenario where the chaser must perform a transfer between two natural motion trajectories (NMTs) while avoiding some obstacles. This characterizes various types of ARPOD missions such as docking or inspection. An NMT is a cyclic trajectory in Hill's frame that requires no input to the chaser. There are three types of NMTs: stationary, linear, and elliptical. The state of any chaser in an NMT satisfies $v_y = -2\eta r_x$, and (i) $r_x = r_z = v_x = v_z = 0$ for a stationary point, (ii) $r_y = v_x = 0$, $r_z = c \sin(\Psi)$, and $v_z = \eta \cos(\Psi)$ for some amplitude c and initial angle Ψ for a linear NMT, and (iii) $v_x = \frac{\eta}{2} r_y$ for an elliptical NMT.

III. SYNTHESIZING SAFETY CONTROLLERS AND APPROXIMATING THE RECOVERABLE SET

A. Finding tracking error bounds for arbitrary reference trajectories

We first synthesize a tracking controller for \mathcal{A} . Given an upper time bound $T > 0$, the system must track some reference trajectory

$\xi^{\text{ref}} : [0, T] \rightarrow \mathcal{X}$ with reference input signal $\mu^{\text{ref}} : [0, T] \rightarrow \mathcal{U}$ which satisfies $\frac{d}{dt} \xi^{\text{ref}}(t) = A\xi^{\text{ref}}(t) + B\mu^{\text{ref}}(t)$.

A *tracking controller* $h : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{U}$ takes in state x , reference state x^{ref} , and reference input u^{ref} and is of the form

$$h(x, x^{\text{ref}}, u^{\text{ref}}) = \Lambda(x^{\text{ref}} - x) + u^{\text{ref}} \quad (2)$$

where $\Lambda \in \mathbb{R}^{m \times n}$ is a gain matrix. The related synthesis problem is to find a value for Λ such that the system trajectory converges to ξ^{ref} , ie. the tracking error goes to 0. We use a linear-quadratic regulator (LQR) for Λ . First, we define the tracking error as a function of x^{ref} and x such that

$$e(x^{\text{ref}}, x) = x^{\text{ref}} - x. \quad (3)$$

Given h , ξ^{ref} , μ^{ref} , and initial state x_0 , the resulting trajectory of \mathcal{A} satisfies

$$\frac{d}{dt} \xi_{x_0, h}(t) = A\xi_{x_0, h}(t) + B(\Lambda(\xi^{\text{ref}}(t) - \xi_{x_0, h}(t)) + \mu^{\text{ref}}(t)) \quad (4)$$

for every $t \in [0, T]$, and we abuse the notation to rewrite tracking error as a function of time $e(t) = e(\xi^{\text{ref}}(t), \xi_{x_0, h}(t))$. The tracking error evolves according to

$$\dot{e} = (A - B\Lambda)e \quad (5)$$

Consider some quadratic cost

$$J = \int_0^T (x^\top Qx + u^\top Ru) dt \quad (6)$$

where $M \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{n \times n}$, and $R \in \mathbb{R}^{m \times m}$. Then, $\Lambda = R^{-1}B^\top P$ where $P \in \mathbb{R}^{n \times n}$ is positive definite and satisfies the algebraic Riccati equation (ARE)

$$A^\top P + PA - (PB)R^{-1}(B^\top P) + Q = 0. \quad (7)$$

The subroutine finding P satisfying (7) is `solveARE(A, B, Q)`.

Proposition 1. *Fix system \mathcal{A} , some $Q, R \succ 0$, and let $P = \text{solveARE}(A, B, Q)$. For $\Lambda = R^{-1}B^\top P$, $(A - B\Lambda)$ is stable.*

The proof comes from the derivation of the LQR. The full safety controller S is constructed using ξ^{ref} , μ^{ref} , and Λ computed as in Proposition 1. Then S is computed using subroutine `constructS(A, ξ^{ref} , μ^{ref})`, which returns $s(x, t) = \Lambda(\xi^{\text{ref}}(t) - x) + \mu^{\text{ref}}(t)$. Given some $x_0 \in \mathcal{X}$, we can use input to state stability to bound the distance between ξ^{ref} and $\xi_{x_0, s}$.

Lemma 1. *Given system \mathcal{A} , upper time bound $T > 0$, reference trajectory ξ^{ref} , reference input signal μ^{ref} , and initial state x_0 , compute Λ as in Proposition 1 and let $S = \text{constructS}(A, \xi^{\text{ref}}, \mu^{\text{ref}})$. Then $\|\xi^{\text{ref}}(t) - \xi_{x_0, s}(t)\| \leq e_0 \gamma \exp(-\beta t) + \frac{\gamma}{\beta} \|D\| w_{\max}$ for some $\gamma > 0$ and $0 < \beta < \lambda_{\max}$ where λ_{\max} is the maximum real value of the eigenvalues of $(A - B\Lambda)$ and $e_0 = \|\xi^{\text{ref}}(0) - x_0\|$.*

The proof comes from using input-to-stability to find an upper bound on the magnitude of the tracking error at any point in time.

B. Constructing approximations of the recoverable set

In *Polaris*, we use recoverable sets to determine when to switch between the experimental (U) and safety (S) controllers. If S is deployed while \mathcal{A} is within the recoverable set, the resulting trajectory should remain safe. We define a *time shifted trajectory* $\xi_{x_t, S}^{0 \rightarrow t} : [0, T - t] \rightarrow \mathcal{X}$, which satisfies $\xi_{x_t, S}^{0 \rightarrow t}(0) = x_t$ and $\frac{d}{d\tau} \xi_{x_t, S}^{0 \rightarrow t}(\tau) = A\xi_{x_t, S}^{0 \rightarrow t}(\tau) + BS(\xi_{x_t, S}^{0 \rightarrow t}(\tau), \tau + t)$ for $t \in [0, T]$, and recoverable sets are defined in Definition 1.

Definition 1 (Recoverable set). Given a system \mathcal{A} , upper time bound $T > 0$, reference trajectory ξ^{ref} , controller S , and unsafe set $\mathcal{O} \subseteq \mathcal{X} \times \mathbb{R}_{\geq 0}$. At time $t \in [0, T]$, the recoverable set \mathcal{C}_S^t is given by

$$\mathcal{C}_S^t = \{x_t \in \mathcal{X} \mid \xi_{x_t, S}^{0 \rightarrow t}(\tau) \notin \mathcal{O}^{t+\tau} \quad \forall \tau \in [0, T-t]\}. \quad (8)$$

The set of all recoverable sets over $[t_0, t_1]$, $0 \leq t_0 \leq t_1 \leq T$, is denoted by $\mathcal{C}_S^{[t_0, t_1]}$ where each $\mathcal{C}_S^t \in \mathcal{C}_S^{[t_0, t_1]}$ for $t \in [t_0, t_1]$.

We use Lemma 1 and time invariance to construct an under-approximation of the recoverable set about a reference trajectory. The constructed safety controller should thus result in a safe trajectory for every initial state within some initial set, leading to Assumption 1.

Assumption 1. Consider system \mathcal{A} , upper time bound $T > 0$, initial set $\Theta \subseteq \mathcal{X}$, and unsafe sets $\mathcal{O}^t \subseteq \mathcal{X}$ for each time $t \in [0, T]$. Then reference trajectory ξ^{ref} satisfies

$$\min_{y \in \mathcal{O}^t} \|\xi^{\text{ref}}(t) - y\| \geq e_0 \gamma \exp(-\beta t) + \frac{\gamma}{\beta} \|D\| w_{\max} \quad (9)$$

where $e_0 = \max_{x \in \Theta} \|\xi^{\text{ref}}(0) - x\|$ and γ and β are computed as in Lemma 1.

Fix system \mathcal{A} , upper time bound $T > 0$, initial set Θ , unsafe sets \mathcal{O} , and reference trajectory ξ^{ref} which satisfies Assumption 1 for some reference input signal μ^{ref} and gain matrix Λ . Fix arbitrary switching time $t \in [0, T]$. We must find a set of states which is contained within the set \mathcal{C}_S^t given by Definition 1. Using Lemma 1, we get tracking error

$$\begin{aligned} & \|\xi^{\text{ref}}(t + \tau) - \xi_{x_t, S}^{0 \rightarrow t}(\tau)\| \\ & \leq \|\xi^{\text{ref}}(t) - \xi_{x_t, S}^{0 \rightarrow t}(0)\| \gamma \exp(-\beta \tau) + \frac{\gamma}{\beta} \|D\| w_{\max} \quad (10) \end{aligned}$$

which is valid over $\tau \in [0, T-t]$.

Lemma 2. Consider a system \mathcal{A} , reference trajectory ξ^{ref} , reference input signal μ^{ref} , gain matrix Λ constructed as in Proposition 1, and $S = \text{constructS}(\Lambda, \xi^{\text{ref}}, \mu^{\text{ref}})$. For any $t \in [0, T]$, if $x_t \in \mathcal{B}_{\hat{r}(t)}(\xi^{\text{ref}}(t))$ then $\|\xi^{\text{ref}}(t + \tau) - \xi_{x_t, S}^{0 \rightarrow t}(\tau)\| \leq r_{\min}$ for all $\tau \in [0, T-t]$ where $\hat{r}(t) = \frac{r_{\min} - \frac{\gamma}{\beta} \|D\| w_{\max}}{\gamma \max_{\tau \in [t, T]} \exp(-\beta(\tau-t))}$.

The proof follows from Lemma 1. The subroutine for computing the under approximation is called `GetRecover`($\mathcal{A}, \Lambda, \mathcal{O}, \xi^{\text{ref}}, r_{\min}$) and it returns recoverable set approximation

$$\mathcal{C}_S^t = \mathcal{B}_{\hat{r}(t)}(\xi^{\text{ref}}(t)) \times \{t\}, \quad (11)$$

$$\hat{r}(t) = \frac{r_{\min} - \frac{\gamma}{\beta} \|D\| w_{\max}}{\gamma \max_{\tau \in [t, T]} \exp(-\beta(\tau-t))}. \quad (12)$$

To ensure safety, we get r_{\min} in the following way. At time $s \in [0, T]$, the minimum distance between $\xi^{\text{ref}}(s)$ and \mathcal{O}^s is given by $d_{\min}(s) = \min_{y \in \mathcal{O}^s} \|\xi^{\text{ref}}(s) - y\|$, and the absolute minimum distance from ξ^{ref} and \mathcal{O} over $[0, T]$ is $r_{\min} = \min_{s \in [0, T]} d_{\min}(s)$. Therefore, we must ensure that the quantity in (10) is less than or equal to r_{\min} over $\tau \in [0, T-t]$. Note that this is a very conservative under-approximation of the recoverable set with no disconnectivities and that an interesting future direction is to come up with a less conservative approximation.

C. Synthesizing reference trajectories

We use a mixed integer program (MIP) to synthesize the reference trajectories and reference input signals for our safety controller. Thus, we find a discrete sequence of states which can be constructed into a reference trajectory and reference input signal. In this section, we convert between discrete time and continuous time trajectories.

Consider system \mathcal{A} with state matrix A , input matrix B , and linear continuous time dynamics $\dot{x} = Ax + Bu$. The corresponding

discrete time trajectory is $x^{k+1} = A^\delta x^k + B^\delta u^k$ for $x^{k+1}, x^k \in \mathcal{X}$, $u^k \in \mathcal{U}$, and $k \in \mathbb{Z}_{\geq 0}$. Here, δ is the time step interval between each x^k and x^{k+1} , and the matrices $A^\delta \in \mathbb{R}^{n \times n}$ and $B^\delta \in \mathbb{R}^{n \times m}$ are given by

$$A^\delta = e^{A\delta} \text{ and } B^\delta = \left(\int_0^\delta e^{As} ds \right) B. \quad (13)$$

Consider discrete time trajectory $\chi : \{0, \dots, K\} \rightarrow \mathcal{X}$ and discrete time input signal $\nu : \{0, \dots, K\} \rightarrow \mathcal{U}$ of length $K > 0$ which satisfy $\chi^{k+1} = A^\delta \chi^k + B^\delta \nu^k$ for $k \in \{0, \dots, K-1\}$. We define the functions D2C_x and D2C_u to convert χ and μ to a continuous time trajectory and input signal respectively. That is, $\xi = \text{D2C}_x(\chi, \nu, \delta)$ satisfies $\xi(0) = \chi^0$, $\xi(k\delta) = \chi^k$ for $k \in \{0, \dots, K\}$, and $\frac{d}{dt}\xi(t) = A\xi(t) + B\nu^{\lfloor \frac{t}{\delta} \rfloor}$. Similarly, $\mu = \text{D2C}_u(\nu, \delta)$ satisfies $\mu(t) = \nu^k$ for $t \in [k\delta, (k+1)\delta)$, $k \in \{0, \dots, K\}$, and we can see that $\frac{d}{dt}\xi(t) = A\xi(t) + B\mu(t)$ for every $t \in [0, K\delta]$. Furthermore, if we consider Λ computed as in Proposition 1 and $S = \text{constructS}(\Lambda, \xi, \mu)$, then we get a corollary of Lemma 1.

Corollary 1. Consider system \mathcal{A} , time step δ , length $K > 0$, initial state x_0 and gain matrix Λ computed as in Proposition 1. Consider discrete time trajectory $\chi : \{0, \dots, K\} \rightarrow \mathcal{X}$ and discrete time input signal $\nu : \{0, \dots, K\} \rightarrow \mathcal{U}$ which satisfies $\chi^{k+1} = A^\delta \chi^k + B^\delta \nu^k$ for $k \in \{0, \dots, K-1\}$ and $S = \text{constructS}(\Lambda, \xi, \mu)$ for $\xi = \text{D2C}_x(\chi, \nu, \delta)$ and $\mu = \text{D2C}_u(\nu, \delta)$. Then $\|\chi^k - \xi_{x_0, S}(k\delta)\| \leq e_0 \gamma \exp(-\beta k\delta) + \frac{\gamma}{\beta} \|D\| w_{\max}$ for each $k \in \{0, \dots, K\}$.

We can use this result to construct safe trajectories. Consider an initial set $\Theta \subseteq \mathcal{X}$, some length $K > 0$, and time step interval $\delta > 0$. We define the subroutine `getErrBounds`($\mathcal{A}, \Lambda, \Theta, \delta, K$) which returns tracking error bounds $\{\varepsilon^k\}_{k=0}^K$ where $\varepsilon^0 = \max_{x \in \Theta} \|\text{Center}(\Theta) - x\|$ and $\varepsilon^k = \varepsilon^0 \gamma \exp(-\beta k\delta) + \frac{\gamma}{\beta} \|D\| w_{\max}$ for $k \in \{1, \dots, K\}$. We use a *perception oracle* [22] which returns an over approximation of the unsafe sets over $[0, K\delta]$ as timed polytopes.

Definition 2. Given state space $\mathcal{X} \subseteq \mathbb{R}^n$, upper time bound $T > 0$, and unsafe sets \mathcal{O} , a *perception oracle* $\text{FS}_T^\mathcal{O}$ returns a collection of timed polytopes $\{O_i = \text{Poly}(H_i, b_i)\}$ such that $\mathcal{O}^t \subseteq (\text{FS}_T^\mathcal{O} \cap \mathcal{X} \times \{t\}) \downarrow \mathcal{X}$ for every $t \in [0, T]$.

We now use a MIP to find discrete time trajectories and input signals. Let φ be an *atomic proposition*. If x satisfies φ , we denote this as $x \models \varphi$. The boolean operator *AND* is denoted by \wedge , such that given atomic propositions φ_1 and φ_2 , if x satisfies both we write this as $x \models \varphi_1 \wedge \varphi_2$. We now introduce `safeConstraints`($\mathcal{A}, \Theta, \text{FS}_T^\mathcal{O}, \{\varepsilon^k\}_{k=0}^K$)

$$\arg \max J(\chi, \mu) \text{ s.t. } \forall k \in \{0, \dots, K-1\} :$$

$$x^{k+1} \models A^\delta x^k + B^\delta u^k \quad (14)$$

$$x^k, x^{k+1} \models \bigwedge_{O_i \in \text{FS}_T^\mathcal{O}} \text{linSafety}(k, O_i, \max(\varepsilon^k, \varepsilon^{k+1})) \quad (15)$$

$$-u_{\max} \leq \mu^k \leq u_{\max} \quad (16)$$

$$x^0 \models \text{Center}(\Theta) \quad (17)$$

$$x^K \models \text{goalCond} \quad (18)$$

where $J(\chi, \mu)$ is a cost function (different from (6)) given by

$$J(\chi, \mu) = \min_{k \in \{0, \dots, K\}} \left(\min_{O_i \in \mathcal{O}^t} \left(\min_{y \in O_i \downarrow \mathcal{X}} \|\chi^k - y\| \right) \right). \quad (19)$$

This cost function is chosen such that the resulting trajectory is far away from the unsafe set in the hope that the resulting recoverable set is less conservative. The dynamic constraints are given by (14) and ensure that each χ^k follows the system dynamics. The safety

constraints are given in (15) and ensure that the resulting trajectory satisfies Assumption 1. Finally, the constraints in (16) ensure the reference input remains within \mathcal{U} ; the constraints in (17) ensure the the reference input starts in Θ ; and the constraints in (18) encode the goal conditions. For our ARPOD example, these are the conditions that result in the satellite ending in an NMT, which were introduced in [18], and implemented in [23], [24] to synthesize guaranteed safe controllers which drive satellites to NMTs. We do not re-derive the constraints here, but have re-stated them in Example 2.

Example 2 (NMT goal conditions). *Consider two polygons \mathcal{P} and \mathcal{Q} which are an approximation for some inner and outer NMTs which the satellite trajectory must end between. Then, the constraints for the end condition of the satellite are given by*

$$\begin{aligned} \bigwedge_{j=1, \dots, N_{\mathcal{P}}} \alpha_{x,j} r_x^k + \alpha_{y,j} r_y^k &\leq \gamma_j & (20) \\ \bigwedge_{j=1, \dots, N_{\mathcal{P}}} \hat{n}_{p,j}^\top \left(\begin{bmatrix} r_x \\ r_y \end{bmatrix} - \begin{bmatrix} r_{x,j} \\ r_{y,j} \end{bmatrix} \right) + M(1 - \zeta_j^k) &\geq 0 \\ \sum_{j=1, \dots, N_{\mathcal{P}}} \zeta_j^k &\geq 1 & (21) \end{aligned}$$

where $\mathbf{q}_j = \begin{bmatrix} q_{x,j} \\ q_{y,j} \end{bmatrix} = \begin{bmatrix} b_{\max} \cos(\frac{2\pi j}{N_{\mathcal{Q}}}) \\ 2b_{\max} \sin(\frac{2\pi j}{N_{\mathcal{Q}}}) \end{bmatrix}$, $\alpha_{x,j} = (q_{y,j} - q_{y,j+1})$, $\alpha_{y,j} = (q_{x,j} - q_{x,j+1})$, $\gamma_j = \alpha_{x,j} q_{x,j} + \alpha_{y,j} q_{y,j}$, the normal vector to the ellipse at point \mathbf{p}_j is given by $\hat{n}_{p,j}$ and $\mathbf{p}_j = \begin{bmatrix} r_{x,j} \\ r_{y,j} \end{bmatrix} = \begin{bmatrix} b_{\min} \cos(\frac{2\pi j}{N_{\mathcal{P}}}) \\ 2b_{\min} \sin(\frac{2\pi j}{N_{\mathcal{P}}}) \end{bmatrix}$.

The ellipses are defined by the semi-minor axes, which are given by b . The state constraint for the goal condition is

$$v_y^K = -2\eta r_x^K \wedge v_x^K = \frac{1}{2}\eta r_y^K \quad (22)$$

Then, *goalCond* comprises (20)–(22).

We now discuss the safety constraints. Consider timed polytope $O = \text{Poly}(H_O, b_O)$, states x, y , time step $k \in \{0, \dots, K\}$, and tracking error bound $\varepsilon > 0$. The safety constraints are

$$\begin{aligned} \text{linSafety}(k, O, \varepsilon) := & \\ \bigwedge_{r \in \text{dP}(H_O)} \left(-H_O^{(r)} \begin{bmatrix} x \\ k\delta \end{bmatrix} + (b_O^{(r)} + 2\varepsilon \|H_O^{(r)}\|) \leq M(1 - \alpha^r) \right. & \\ \wedge \left. -H_O^{(r)} \begin{bmatrix} y \\ (k+1)\delta \end{bmatrix} - (b_O^{(r)} + 2\varepsilon \|H_O^{(r)}\|) \leq M(1 - \alpha^r) \right) & \\ \sum_{r \in \text{dP}(H_O)} \alpha^r \geq 1 & \quad (23) \end{aligned}$$

where $M \gg 0$ and each $\alpha^r \in \{0, 1\}$ is a binary variable which indicates whether or not the first two constraints are met. If we choose appropriate δ , as stated in Assumption 2, we get the safety property in Lemma 3.

Assumption 2. *Consider some bound $\varepsilon > 0$, state matrix A , and input matrix B . We choose δ such that for discrete time trajectory $\chi = \{\chi^k\}_{k=0}^K$ and input signal $\nu = \{\nu^k\}_{k=0}^K$ that satisfy $\chi^{k+1} = A^\delta \chi^k + B^\delta \nu^k$ for $k \in \{0, \dots, K-1\}$ where A^δ and B^δ are constructed as in (13), $\xi = D2C_x(\chi, \nu, \delta)$ satisfies $\|\xi(t) - \chi^k\| \leq \varepsilon$.*

Lemma 3. *Consider system \mathcal{A} , error bound ε , fixed $\delta > 0$ that satisfies Assumption 2, and timed polytope $O = \text{Poly}(H_O, b_O)$. If $x, y \models \text{linSafety}(k, O, \varepsilon)$ for x, y at time step k and $y = A^\delta x + B^\delta u$ for some $u \in \mathcal{U}$ where A^δ and B^δ are computed as in (13), then $\xi = D2C_x(\{x, y\}, \{u\}, \delta)$ satisfies $\mathcal{B}_\varepsilon([\xi(t), k\delta + t]) \cap O = \emptyset$ for every $t \in [0, \delta]$.*

The proof comes from showing that segments connecting two waypoints lie outside the over-approximations of the unsafe sets, and then showing that the trajectory which connects these two waypoints also lies outside the unsafe sets.

The safety controller and recoverable set are synthesized in Algorithm 1 which we call *GetSafety*. The inputs are the system \mathcal{A} , length K , time step interval δ , initial set Θ , unsafe sets \mathcal{O} , and positive definite cost matrices $Q, R \succ 0$. First, we find P by solving the ARE, and the gain matrix Λ is computed. Next, we find the tracking error bounds at each $k\delta$, $k \in \{0, \dots, K\}$. Then we find the discrete time trajectory and input signal which results in a safe reference trajectory and input signal. Finally, we construct the safety controller and under approximation of the recoverable set. Any controller returned by *GetSafety* guarantees a safe trajectory when the safety controller is deployed within the recoverable set.

Algorithm 1: GetSafety

Input: system \mathcal{A} , length K , time step interval δ , initial set Θ , unsafe sets \mathcal{O} , cost matrices $Q, R \succ 0$
Output: safety controller S , recoverable set $\underline{\mathcal{C}}_S$

- 1 $P \leftarrow \text{solveARE}(A, B, Q)$
- 2 $\Lambda \leftarrow R^{-1} B^\top P$
- 3 $\{\varepsilon^k\}_{k=0}^K \leftarrow \text{getErrBounds}(\mathcal{A}, \Lambda, \Theta, \delta, K)$
- 4 $\chi, \nu \leftarrow \text{safeConstraints}(\mathcal{A}, \Theta, \mathbf{FS}_{K\delta}^{\mathcal{O}}, \{\varepsilon^k\}_{k=0}^K, \delta, K)$
- 5 $\xi^{\text{ref}} \leftarrow D2C_x(\chi, \nu, \delta)$
- 6 $\mu^{\text{ref}} \leftarrow D2C_u(\nu, \delta)$
- 7 $S \leftarrow \text{constructS}(\Lambda, \xi^{\text{ref}}, \mu^{\text{ref}})$
- 8 $r_{\min} \leftarrow \min_{t \in [0, K\delta]} \min_{y \in \mathcal{O}^t} \|\xi^{\text{ref}}(t) - y\|$
- 9 $\hat{\mathcal{C}}_S \leftarrow \text{GetRecover}(\mathcal{A}, \Lambda, \mathcal{O}, \xi^{\text{ref}}, r_{\min})$
- 10 **return** $S, \underline{\mathcal{C}}_S^{[0, K\delta]}$

Theorem 1. *Consider a system \mathcal{A} , length $K > 0$, time step δ , initial set $\Theta \subseteq \mathcal{X}$, and unsafe sets \mathcal{O}^t for $t \in [0, K\delta]$.*

- (i) *Any $S : [0, K\delta]$ returned by *GetSafety* guarantees $\xi_{x_0, s}(t) \notin \mathcal{O}^t$ for all $t \in [0, K\delta]$ for every $x_0 \in \Theta$, and*
- (ii) *$\xi_{x_t, s}^{0 \rightarrow t}(\tau) \notin \mathcal{O}^{t+\tau}$ for all $\tau \in [t, T-t]$ for any $x_t \in \underline{\mathcal{C}}_S^t$ for any $t \in [0, T]$ for $\underline{\mathcal{C}}_S$ returned by *GetSafety*.*

The proof uses results from Lemmas 1, 3, and 2, as well as corollary 1. We leave the proof to the full version of this paper. We note that *GetSafety* only returns S and $\underline{\mathcal{C}}_S^{[0, K\delta]}$ when \mathcal{O} allows for a ξ^{ref} and μ^{ref} satisfying Assumption 1 to exist. However, we may not find such a ξ^{ref} and μ^{ref} even if one exists if the maximum length of the discrete-time trajectory is too small. This can be mitigated by allowing $K \rightarrow \infty$, but drastically increases the computational time.

IV. SYNTHESIZING SWITCHING LOGICS

We now present *Polaris* in Algorithm 2. The input to *Polaris* is the system \mathcal{A} , experimental controller \mathcal{U} , initial set Θ , unsafe sets \mathcal{O} , upper time bound $T > 0$, and time step interval $\delta > 0$. It returns a safety controller $S : \mathcal{X} \times [0, T] \rightarrow \mathcal{U}$ and switching logic $\mathcal{L} : [0, T] \rightarrow \mathcal{M}$, which results in a safe trajectory $\xi_{x_0, \mathcal{L}}^{S, \mathcal{U}}$ for every $x_0 \in \Theta$.

First, the number of steps K is computed to be the ceiling of $\frac{T}{\delta}$. Then, the safety controller and recoverable set are computed using *GetSafety*. The over approximation of the reachable set, defined in Definition 3, of the system using the experimental controller over $[0, T]$ is then computed using *GetReach*. There are many off-the-shelf reachability tools which can be used to compute these over approximations, such as C2E2 [25], Hylaa [26], CORA [27], or DryVr [28].

Algorithm 2: Polaris

Input: system \mathcal{A} , experimental controller U , initial set Θ , unsafe sets \mathcal{O}^t , upper time bound T , time step interval δ , cost matrices $Q, R \succ 0$

Output: safety controller S , switching logic \mathcal{L}

- 1 $K \leftarrow \lceil \frac{T}{\delta} \rceil$
- 2 $S, \mathcal{C}_S^{[0, T]} \leftarrow \text{GetSafety}(\mathcal{A}, K, \delta, \Theta, \mathcal{O}, Q, R)$
- 3 $\overline{\mathcal{R}}_{\Theta, U}^{[0, T]} \leftarrow \text{GetReach}(\mathcal{A}, \Theta, U, 0, T)$
- 4 $\{m^k\}_{k=0}^K \leftarrow \{0\}_{k=0}^K$
- 5 $m^0 \leftarrow 1$
- 6 $k \leftarrow 0$
- 7 **while** $k < K$ **do**
- 8 $t_1 \leftarrow k\delta$
- 9 $t_2 \leftarrow (k+1)\delta$
- 10 **if** $m^k = 0$ **then**
- 11 $\Theta \leftarrow \overline{\mathcal{R}}_{\Theta, S}^{t_1}$
- 12 $\overline{\mathcal{R}}_{\Theta, U}^{[t_1, T]} \leftarrow \text{GetReach}(\mathcal{A}, \Theta, U, t_1, T)$
- 13 **if** $\overline{\mathcal{R}}_{\Theta, U}^{[t_1, t_2]} \subseteq \mathcal{C}_S^{[t_1, t_2]}$ **then**
- 14 $m^k \leftarrow 1$
- 15 **else**
- 16 **if** $m^k = 1$ **then**
- 17 $\Theta \leftarrow \overline{\mathcal{R}}_{\Theta, U}^{t_1}$
- 18 $\overline{\mathcal{R}}_{\Theta, S}^{[t_1, T]} \leftarrow \text{GetReach}(\mathcal{A}, \Theta, S, t_1, T)$
- 19 $m^k \leftarrow 0$
- 20 $m^{k+1} \leftarrow m^k$
- 21 $k \leftarrow k+1$
- 22 $\mathcal{L} \leftarrow \text{constructg}(\{m^k\}_{k=0}^K, \delta)$
- 23 **return** S, \mathcal{L}

Definition 3 (Reachable set). Consider a system \mathcal{A} , initial set $\Theta \subseteq \mathcal{X}$, upper time bound T , and controller $U: \mathcal{X} \times [0, T] \rightarrow \mathcal{U}$. Given $0 \leq t_0 \leq t_1 \leq T$, the reachable set $\mathcal{R}_{\Theta, U}^{t_0, t_1}$ is

$$\mathcal{R}_{\Theta, U}^{t_0, t_1} = \bigcup_{x_0 \in \Theta} \{x_{x_0, U}^{0 \rightarrow t_0}(t_1 - t_0)\}. \quad (24)$$

We denote all reachable sets over $[t_0, t_1]$ as $\mathcal{R}_{\Theta, U}^{[t_0, t_1]}$ where $\mathcal{R}_{\Theta, U}^{t_0, \tau} \in \mathcal{R}_{\Theta, U}^{[t_0, t_1]}$ for each $\tau \in [t_0, t_1]$.

Recall that if the mode $m = 0$, the experimental controller is used and if $m = 1$, the safety controller is used. In line 4, a length $K+1$ sequence of modes is set such that $m^k = 0$ for $k \in \{0, \dots, K\}$, and in line 5 $m^0 = 1$. In lines 7-21, the main loop of Polaris is performed. First, the time period over which the k^{th} mode is to be used is set to be $[t_1, t_2]$, $t_1 = k\delta$ and $t_2 = (k+1)\delta$. If $m^k = 0$, we recompute the reachable set of the system using U over $[t_1, T]$. In line 13, we check to see if the reachable set of the system using U is contained within the recoverable set over $[t_1, t_2]$. If so, we set $m^k = 1$. If not, we check to see if $m^k = 1$ currently. If it is, then we compute the reachable set of the system using S over $[t_1, T]$, and then we set $m^k = 0$ regardless of the current mode. Finally, we initialize $m^{k+1} = m^k$ and increment k .

After the main loop terminates, the mode controller is constructed using `constructg`. This subroutine takes in a length $K+1$ sequence of modes $\{m^k\}_{k=0}^K$ and a time step δ , and returns switching logic $\mathcal{L}: [0, T] \rightarrow \mathcal{M}$ that satisfies $\mathcal{L}(t) = m^k$ for $t \in [k\delta, (k+1)\delta)$ for every $k \in \{0, \dots, K-1\}$.

Theorem 2. Consider a system \mathcal{A} , upper time bound T , initial set Θ , unsafe sets \mathcal{O} and experimental controller U . Any safety controller S and switching logic \mathcal{L} returned by Polaris guarantees

that for every $x_0 \in \Theta$, $\xi_{x_0, \mathcal{L}}^{S, U}$ satisfies $\xi_{x_0, \mathcal{L}}^{S, U}(t) \notin \mathcal{O}^t$ for all $t \in [0, T]$.

We defer the full proof to the full version of this paper. There are three cases where to examine safety: (1) using U when the trajectories are contained within the recoverable set, (2) switching to S when the trajectories may exit the recoverable set, and (3) switching back to U when the trajectories return to the recoverable set.

V. NOMINAL RESULTS

We implement Polaris using Python and use it to solve our satellite example. The MIP used to synthesize the reference trajectory and input signal is solved using Gurobi [29] and the reachable sets are computed using Verse [30]. The resulting switching controller and safety controller are evaluated within the RTAeval framework [31]. Here, the experimental controller is also a tracking controller but with a different feedback matrix. We deploy Polaris in two different scenarios where the satellite must avoid: (i) an obstacle static in Hill's frame (Static), and (ii) another satellite operating in Hill's frame (Dynamic). Results are shown in Figure 2 and Table I.

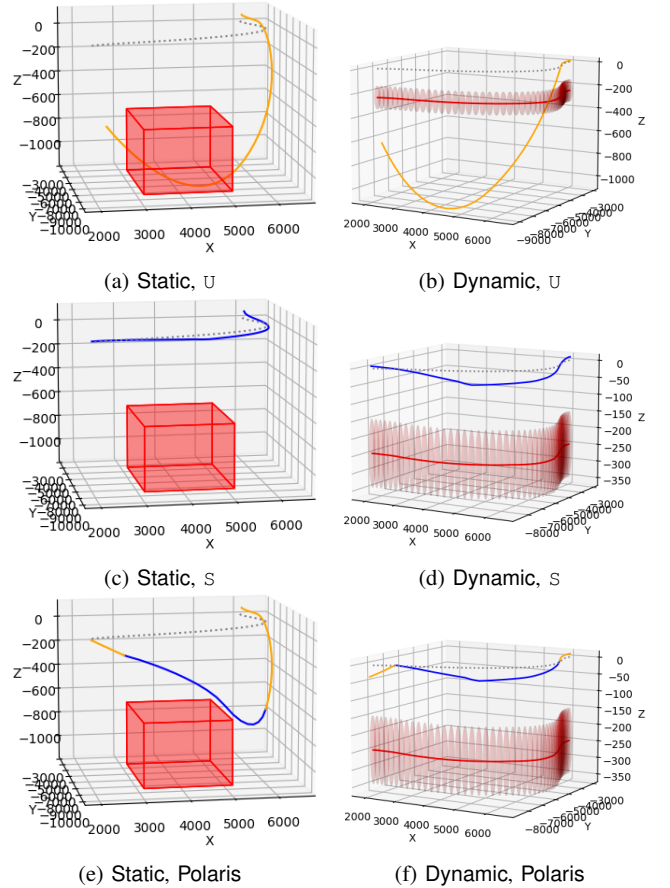


Fig. 2: The dotted gray trajectory is the reference trajectory, orange denotes U blue denotes S . The red sets show the unsafe sets.

In Figure 2, we show the resulting trajectories from using the untrusted (U) and safety (S) controllers, and the switching logic computed using Polaris in both the Static and Dynamic scenarios. Table I shows metrics regarding the minimum distance from the unsafe set, the time until a collision with the unsafe set if the satellite continues on its current trajectory, and the experimental controller usage. The satellite is safe if the distance and time to collision is greater than zero. Here, we can see that the experimental controller always causes the satellite to crash and the synthesized

TABLE I: Results from Static and Dynamic scenarios

Scenario	Dist from \mathcal{O}	TTC (s)	% \mathcal{U}
Static, \mathcal{S}	562.36	inf	0
Static, \mathcal{U}	0	0	100
Static, Polaris	98.01	81.16	63.26
Dynamic, \mathcal{S}	25.96	1538.86	0
Dynamic, \mathcal{U}	-31.40	0	100
Dynamic, Polaris	28.47	328.20	34.69

safety controller keeps the system safe. Furthermore, Polaris also keeps the system safe while increasing the experimental controller usage. We even note the time to collision is always reduced when Polaris is used as the \mathcal{U} controller always causes it to come closer to a collision with the unsafe set, however if distance is the metric that we care about, Polaris can keep the satellite even further away from the unsafe set. This can be seen in the Dynamic scenario. Thus, we can see that Polaris can keep the satellite system safe while increasing the experimental controller usage. Furthermore, since the safety controller and switching logic are synthesized prior to deployment, computation time during runtime is minimal.

VI. CONCLUSION

We solve the problem of guaranteeing safety of a linear system with an experimental controller by synthesizing a safety controller and RTA logic which switches between the synthesized safety and experimental controllers. The RTA logic is synthesized by constructing a recoverable set about a reference trajectory using stability analysis of a tracking controller and checking for the times when the reachable set of the system exits the recoverable set. We call this approach Polaris and it is deployed in a satellite ARPOD scenario and we show that the satellite remains safe even with dynamic obstacles.

Interesting future work is to extend Polaris to systems with nonlinear or hybrid dynamics. Less conservative recoverable sets can be constructed using different reachability methods. Polaris can be extended to create a switching logic for multiple experimental controllers, or optimize switching for different cost functions.

REFERENCES

- [1] C. Petersen, R. J. Caverly, S. Phillips, and A. Weiss, "Safe and constrained rendezvous, proximity operations, and docking," in *2023 American Control Conference (ACC)*, pp. 3645–3661, 2023.
- [2] D. Sun, S. Jha, and C. Fan, "Learning certified control using contraction metric," in *Conference on Robot Learning*, PMLR, 2021.
- [3] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.
- [4] D. C. Guastella and G. Muscato, "Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review," *Sensors*, vol. 21, no. 1, p. 73, 2020.
- [5] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.
- [6] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.
- [7] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *2012 IEEE International Conference on Robotics and Automation*, IEEE.
- [8] G. Behrendt, A. Soderlund, M. Hale, and S. Phillips, "Autonomous satellite rendezvous and proximity operations with time-constrained sub-optimal model predictive control," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9380–9385, 2023. 22nd IFAC World Congress.
- [9] C. Petersen, S. Phillips, D. Hustig-Schultz, and R. Sanfelice, "Towards hybrid model predictive control for computationally aware satellite applications," in *Proceedings of the Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems*, CAADCPS '21, (New York, NY, USA), p. 15–17, Association for Computing Machinery, 2021.
- [10] V. D. Sagar and T. Nanjundeswaraswamy, "Artificial intelligence in autonomous vehicles—a literature review," *i-Manager's Journal on Future Engineering and Technology*, vol. 14, no. 3, p. 56, 2019.
- [11] H. H. Lei, M. Shubert, N. Damron, K. Lang, and S. Phillips, "Deep reinforcement learning for multi-agent autonomous satellite inspection," in *44th annual AAS guidance, navigation and control (GN&C) conference*, Breckenridge, CO, 2022.
- [12] Y. Peng, G. Tan, and H. Si, "Rta-ir: A runtime assurance framework for behavior planning based on imitation learning and responsibility-sensitive safety model," *Expert Systems with Applications*, 2023.
- [13] U. Mehmood, S. Sheikhi, S. Bak, S. A. Smolka, and S. D. Stoller, "The black-box simplex architecture for runtime assurance of autonomous cps," in *NASA Formal Methods Symposium*, Springer, 2022.
- [14] A. Desai, S. Ghosh, S. A. Seshia, N. Shankar, and A. Tiwari, "Soter: a runtime assurance framework for programming safe robotics systems," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE.
- [15] B. Yalcinkaya, H. Torfah, A. Desai, and S. A. Seshia, "Ulgen: A runtime assurance framework for programming safe cyber-physical systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.
- [16] S. Bak, K. Manamcheri, S. Mitra, and M. Caccamo, "Sandboxing controllers for cyber-physical systems," in *2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*, IEEE.
- [17] K. Dunlap, M. Hibbard, M. Mote, and K. Hobbs, "Comparing run time assurance approaches for safe spacecraft docking," *IEEE Control Systems Letters*, vol. 6, pp. 1849–1854, 2021.
- [18] M. L. Mote, C. W. Hays, A. Collins, E. Feron, and K. L. Hobbs, "Natural motion-based trajectories for automatic spacecraft collision avoidance during proximity operations," in *2021 IEEE Aerospace Conference (50100)*, IEEE.
- [19] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICPPS)*, IEEE.
- [20] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, IEEE.
- [21] M. Hibbard, U. Topcu, and K. Hobbs, "Guaranteeing safety via active-set invariance filters for multi-agent space systems with coupled dynamics," in *2022 American Control Conference (ACC)*, IEEE.
- [22] K. Miller, C. Fan, and S. Mitra, "Planning in dynamic and partially unknown environments," *IFAC-PapersOnLine*, vol. 54, no. 5, 2021.
- [23] K. Miller, J. M. Brewer, A. A. Soderlund, and S. Phillips, "Sensor safety and multi-objective satellite control under nonlinear dynamics," in *2023 American Control Conference (ACC)*, IEEE, 2023.
- [24] K. Miller, S. Phillips, and A. A. Soderlund, "Multi-agent control of chaser satellites using games with lexicographic preferences," in *AIAA SCITECH 2023 Forum*.
- [25] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok, "C2e2: A verification tool for stateflow models," in *Tools and Algorithms for the Construction and Analysis of Systems: 21st International Conference, TACAS 2015*, Springer.
- [26] S. Bak and P. S. Duggirala, "Hylaa: A tool for computing simulation-equivalent reachability for linear systems," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, 2017.
- [27] M. Althoff, "An introduction to cora 2015.," *ARCH@ CPSWeek*, vol. 34, 2015.
- [28] C. Fan, B. Qi, S. Mitra, and M. Viswanathan, "Dryvr: Data-driven verification and compositional reasoning for automotive systems," in *International Conference on Computer Aided Verification*, Springer, 2017.
- [29] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023.
- [30] Y. Li, H. Zhu, K. Braught, K. Shen, and S. Mitra, "Verse: A python library for reasoning about multi-agent hybrid system scenarios," in *International Conference on Computer Aided Verification*, Springer, 2023.
- [31] K. Miller, C. K. Zeitler, W. Shen, M. Viswanathan, and S. Mitra, "Rtaeval: A framework for evaluating runtime assurance logic," *arXiv preprint arXiv:2306.04585*, 2023.