MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

A Robust Invariant Set Planner For Quadrotors

Greiff, Marcus; Weiss, Avishai; Berntorp, Karl; Di Cairano, Stefano

TR2024-081 June 26, 2024

Abstract

We propose a motion planner for quadrotors implemented as a search on a graph constructed from robust positively invariant (PI) sets. We model the position error dynamics of the quadrotor in closed-loop with an onboard controller as a second-order system with polytopic uncertainty in the gains. In addition, we consider bounded attitude tracking errors and additive input disturbances. Using linear matrix inequalities (LMIs), we compute small ellipsoidal robust PI sets and large ellipsoidal inflated safe PI sets around positional setpoints where collision avoidance and input constraints are satisfied. We use the sets to construct a graph where the nodes are associated to position setpoints, and the edges are included if it is possible to transition from one node setpoint to the next one while satisfying constraints and avoiding collisions. The motion plan is obtained by selecting the sequence of active setpoints based on set membership conditions. The construction of the graph can be performed offline, while the online computation of the motion plan is simple and fast, as demonstrated by a Monte- Carlo simulation study in a cluttered indoor environment.

European Control Conference (ECC) 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

A Robust Invariant Set Planner For Quadrotors

Marcus Greiff, Avishai Weiss, Karl Berntorp, Stefano Di Cairano

***Abstract*— We propose a motion planner for quadrotors implemented as a search on a graph constructed from robust positively invariant (PI) sets. We model the position error dynamics of the quadrotor in closed-loop with an onboard controller as a second-order system with polytopic uncertainty in the gains. In addition, we consider bounded attitude tracking errors and additive input disturbances. Using linear matrix inequalities (LMIs), we compute small ellipsoidal robust PI sets and large ellipsoidal inflated safe PI sets around positional setpoints where collision avoidance and input constraints are satisfied. We use the sets to construct a graph where the nodes are associated to position setpoints, and the edges are included if it is possible to transition from one node setpoint to the next one while satisfying constraints and avoiding collisions. The motion plan is obtained by selecting the sequence of active setpoints based on set membership conditions. The construction of the graph can be performed offline, while the online computation of the motion plan is simple and fast, as demonstrated by a Monte-Carlo simulation study in a cluttered indoor environment.**

I. Introduction

Unmanned aerial vehicles (UAVs) are increasingly utilized in industrial applications, such as indoor surveying and factory automation, where the environment is well known and largely static [1]. In such settings, autonomous UAV operations require motion-planning methods with rigorous safety guarantees that enable fast planning (and re-planning) of the UAV motion. In this work we are particularly interested in planners that:

(i) Provide theoretical guarantees of safety in the presence of disturbances and system modeling errors;
(ii) Ensure that relevant variables, such as the maximal thrust of the UAV, are kept within their bounds;
(iii) Report *a priori* if safe flight is impossible, and provide limitations for recovering of safe flight.

The dominating paradigm for UAV motion-planning leverages convex optimization, see, e.g., [2]–[5]. For example, the minimum-snap planners in [4], [5] were designed to facilitate aggressive maneuvering [3], [4]. Some extensions leverage simple disturbance models [6], but these planning methods generally neglect structured uncertainties and disturbances acting on the system. In practice, uncertainties are dealt with by inflating the obstacles, but determining the amount of inflation that ensures safety is a nontrivial task.

For planning methods based on convex optimization, point (i) is challenging to achieve without sacrificing computational efficiency. This motivates an exploration of graph-based planning, where instead of optimizing over trajectories in space, the UAV motions are encoded though a graph, enabling usage of highly efficient search algorithms [7]. Often, in graph-based planning motions are *explicitly* associated with graph edges [8]. Here, to comprehensively address (i)–(iii), we consider invariant-set motion planners (ISMPs) [9]–[11], in which the motions are encoded *implicitly*, as the closed-loop response to a sequence of setpoint commands (see Fig. 1). Leveraging invariant sets, such methods can provide guarantees on robust constraint satisfaction and on convergence to a target (positive invariant) PI set.

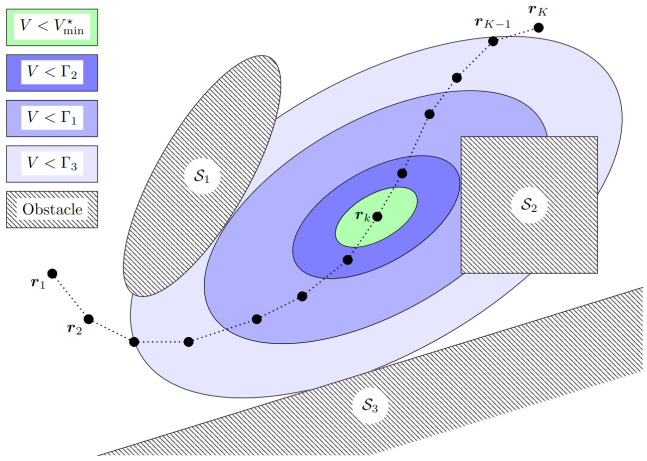

Fig. 1. Core idea of the RISMP method: A sequence of set-points $\{\boldsymbol{r}_k \in \mathbb{R}^3\}_{k=1}^K$ are computed in a world with $M = 3$ three obstacles $\{\mathcal{S}_i \subset \mathbb{R}^3\}_{i=1}^M$ (gray). Large safe sub-level sets $\{\Gamma_i\}_{i=1}^M$ (blue) of the a Lyapunov function V^k are computed for each single obstacle, centered at $\boldsymbol{r}_k$. Here, the sublevel set that is safe for all obstacles is $V^k \leq V_{\max} = \Gamma_2$. A small robust PI set for the node associated with the set-point $\boldsymbol{r}_k$ (green) is used to determine valid connections to this node.

The ISMP methods have been employed in various settings [9]–[17], most notably automotive applications [11], [14], [18], but they have also been considered for applications such as spacecraft orbit [9] and attitude control [16], [17]. Related invariance-based methods have been suggested for use in UAVs in [13], and there is a parallel track of developing invariance-based explicit reference governors (ERGs) [19] tailored for UAVs in [20].

In this paper, we extend the framework in [15] to construct a robust ISMP (RISMP) for UAVs that addresses (i)–(iii). To this end, we borrow preliminary results from the ERG literature [19] and extend the ideas in [15] to develop a new RISMP framework for UAVs. The method provides rigorous safety guarantees and enables long-horizon planning ($\sim$30s), with computation time on the order of milliseconds for large and cluttered environments.

Notation: Vectors are written $\boldsymbol{x} \in \mathbb{R}^n$ with $[\boldsymbol{x}]_i$ denoting

A. Weiss, K. Berntorp and S. Di Cairano are with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA (e-mail: weiss@merl.com, karl.o.berntorp@ieee.org, dicairano@ieee.org)

M. Greiff was with Mitsubishi Electric Research Laboratories at the time of this research (e-mail: marcus.greiff@gmail.com)

the i^{th} element, and $\boldsymbol{e}_i$ is a unit vector where $[\boldsymbol{e}_i]_i = 1$. The identity matrix is denoted by $\boldsymbol{I}_d \in \mathbb{R}^{d\times d}$. The $\star$ denotes a block in a symmetric matrix defined by an existing block and symmetry. We let $\|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^\top \boldsymbol{x}$, denote $\|\boldsymbol{x}\|_{\boldsymbol{P}}^2 = \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}$, and $(\mathbb{S}_+^n)\mathbb{S}_{++}^n$ is the cone of $n\times n$ positive (semi)definite matrices. For any $\boldsymbol{P} \in \mathbb{S}_{++}^n$, an associated ellipsoidal volume is denoted by $\Omega(\boldsymbol{P},\lambda) = \{\boldsymbol{x} \in \mathbb{R}^n | \|\boldsymbol{x}\|_{\boldsymbol{P}}^2 \leq \lambda\}$, and this set is PI if for all $\boldsymbol{x}(t_0) \in \Omega(\boldsymbol{P},\lambda) \Rightarrow \boldsymbol{x}(t) \in \Omega(\boldsymbol{P},\lambda) \forall t \geq t_0$. All invariant sets are denoted by $\mathcal{O}$. The convex combination of the elements of a set $\mathcal{S}$ is denoted by $\mathsf{Co}(\mathcal{S})$. The maximum and minimum eigenvalues of a real, symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ are $\bar{\lambda}(\boldsymbol{M}) \in \mathbb{R}$ and $\underline{\lambda}(\boldsymbol{M}) \in \mathbb{R}$, respectively. The set of rotations is denoted by $\mathsf{SO}(3) = \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} | \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\}$. A directed graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ is a set of vertices (or nodes) $\mathcal{V}$ together with ordered pairs $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ called edges. Vertices $\mathcal{V}_i, \mathcal{V}_j \in \mathcal{V}$ are adjacent if $(\mathcal{V}_i,\mathcal{V}_j) \in \mathcal{E}$ is an edge. A path is a sequence of adjacent vertices. We let $\boldsymbol{r}_k \in \mathbb{R}^3$ be a setpoint and $V^k = \|\boldsymbol{x}_e\|_{\boldsymbol{P}}^2$ be a Lyapunov function associated with $\mathcal{V}_k$, where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{pp}} & \boldsymbol{P}_{\boldsymbol{pv}} \\ \star & \boldsymbol{P}_{\boldsymbol{vv}} \end{bmatrix} \in \mathbb{S}_{++}^6, \quad \boldsymbol{P}_{\boldsymbol{pp}} \in \mathbb{S}_{++}^3, \quad \boldsymbol{P}_{\boldsymbol{vv}} \in \mathbb{S}_{++}^3,$$

with $\boldsymbol{x}_{e,k} = (\boldsymbol{p} - \boldsymbol{r}_k; \boldsymbol{v})$, and usually we omit the index k. Finally, $E \sim \mathcal{U}(\mathcal{M})$ is a uniform selection of an element $E \in \mathcal{M}$, with every element in $\mathcal{M}$ drawn with equal probability.

II. Assumptions and Problem Formulation

For small UAVs, there is a significant time-scale separation between the attitude dynamics and the positional dynamics. As such, it is natural to consider only the positional subsystem for motion-planning, but in a manner that accounts for bounded attitude tracking errors due to a nonideal attitude controller. Furthermore, bounded additive disturbances on the UAV accelerations due to imperfect actuation and sensing are also accounted for, as in the robust ERG in [20].

Assumption 1 *The system response of the UAV with a nonideal attitude controller is governed by*

$$\ddot{\boldsymbol{p}} = -\tilde{\boldsymbol{R}}^\top \boldsymbol{K}_p(\boldsymbol{p} - \boldsymbol{r}) - \tilde{\boldsymbol{R}}^\top \boldsymbol{K}_v \boldsymbol{v} + \boldsymbol{\Delta}, \tag{1a}$$

$$\boldsymbol{\Delta} = m^{-1}\boldsymbol{f} + g(\boldsymbol{I} - \tilde{\boldsymbol{R}})\boldsymbol{e}_3, \tag{1b}$$

where $\boldsymbol{f} \in \mathbb{R}^3$ is an external force, $\tilde{\boldsymbol{R}} \in \mathsf{SO}(3)$ denotes an attitude tracking error, with $\boldsymbol{K}_v \in \mathbb{S}_{++}^{3\times 3}$ and $\boldsymbol{K}_p \in \mathbb{S}_{++}^{3\times 3}$, and $m, g > 0$ denote mass and gravitational acceleration.

Equation (1) is the error dynamics that results from using a geometric proportional derivative (PD) controller [21], [22] for setpoint stabilization. Unlike [20], we do not assume perfect knowledge of such error dynamics.

Assumption 2 *The feedback is stabilizing when $\|\boldsymbol{I} - \tilde{\boldsymbol{R}}\|_2 \equiv 0$ and $\|\boldsymbol{f}\| \equiv 0$, and the gains are known in the sense that*

$$\boldsymbol{K} \triangleq (\boldsymbol{K}_p, \boldsymbol{K}_v) \in \mathsf{Co}(\{(\boldsymbol{K}_p^i, \boldsymbol{K}_v^i)\}_{i=1}^N) \triangleq \mathcal{K}. \tag{2}$$

While it would seem natural to assume that the UAV feedback gains are known, this is not necessarily true, and we use the flexibility offered by such an uncertainty to model different translational behaviors of the UAV about some (approximate) nominal tuning.

Assumption 3 *The attitude tracking error and the external forces acting on the UAV are bounded, $\sup_t |\alpha(t)| \leq \alpha_{\max}$, $\sup_t \|\boldsymbol{f}(t)\|_2 \leq F_{\max}$, respectively, where*

$$\alpha(t) \triangleq \max_{\|\boldsymbol{u}\|_2 = 1} \arccos\left(\boldsymbol{u}^\top \tilde{\boldsymbol{R}}(t)\boldsymbol{u}\right). \tag{3}$$

Given these assumptions, we state the problem:

Problem 1 *Given Assumptions **A1–A3**, compute a piecewise constant reference $\boldsymbol{r}(t) = \boldsymbol{r}_k$ for $t \in [t_{k-1}, t_k]$, where $\{\boldsymbol{r}_k \in \mathbb{R}^3\}_{k=1}^K$ such that $t_K - t_0 \geq 0$ is finite, the UAV avoids a set of convex obstacles $\{\mathcal{S}_i \subseteq \mathbb{R}^3\}_{i=1}^M$ at all times, and converges to a PI set centered at a desired point $\boldsymbol{r}_K$.*

In what follows, we design a motion planner that solves Problem 1. The proofs are omitted due to space limitations.

III. Smallest Ellipsoidal Robust PI Sets

In this section, we show how to compute inner approximations of PI sets to facilitate the design of an RISMP that addresses Problem 1. To this end, we first need to compute PI sets for the system defined by Assumptions **A1**–**A3**. The method in [20] can be applied when the gains in (2) are: known, i.e., $N = 1$; diagonal matrices; and there is no variation in the diagonal elements. Thus, [20] does not accommodate all modes satisfying Assumptions **A1**, **A2**. The LMI-based approach in [15], [23] is more easily generalized, but does not accommodate the structured uncertainty due to bounded attitude tracking error in Assumption **A3**, which is often necessary to consider for precise UAV planning and control.

A. Synthesis via LMIs

We formulate a LMI-based synthesis building on standard results for computing common Lyapunov functions for polytopic systems subject to norm-bounded inputs [23, Chapter 6], with additional constraints to handle the bounded attitude tracking errors in Assumption **A3**

$$\lambda^\star = \underset{\substack{\boldsymbol{P}\in\mathbb{S}_{++}^{6\times 6} \\ \bar{\boldsymbol{K}}\in\mathbb{S}_{+}^{6\times 6} \\ \lambda\geq 0}}{\operatorname{argmin}}(\lambda), \tag{4a}$$

subject to

$$\boldsymbol{P} \succeq \boldsymbol{I}, \tag{4b}$$

$$\begin{bmatrix} \boldsymbol{A}_i^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i + \boldsymbol{P} + \beta\bar{\boldsymbol{K}} & \star & \star \\ \boldsymbol{B}^\top \boldsymbol{P} & -\lambda\boldsymbol{I} & \star \\ \sqrt{\beta}\boldsymbol{B}^\top \boldsymbol{P} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{0}, \tag{4c}$$

$$\begin{bmatrix} \bar{\boldsymbol{K}} & \star \\ \boldsymbol{K}_i & \boldsymbol{I} \end{bmatrix} \succeq \boldsymbol{0}, \tag{4d}$$

for all $i \in 1, ..., N$, where

$$\beta = \sqrt{2(1-\cos(\alpha_{\max}))}, \tag{4e}$$

$$\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_p^i & -\boldsymbol{K}_v^i \end{bmatrix}, \tag{4f}$$

$$\boldsymbol{K}_i = \begin{bmatrix} \boldsymbol{K}_p^i & \boldsymbol{K}_v^i \end{bmatrix}, \tag{4g}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}. \tag{4h}$$

The characterization of an ellipsoidal PI set based on the solution of (4) is summarized in the following proposition.

Proposition 1 *Under Assumptions **A1**-**A3**, given a maximal attitude error $\alpha_{\max}$, if there exists a solution to* (4)*, there exists a constant $V_{\min} = \Delta_{\max}^2 \lambda^\star > 0$ such that $\mathcal{O}_{\min} = \{\boldsymbol{x}_e \in \mathbb{R}^6 : \mathcal{V}(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{r}) \leq V_{\min}\}$ is PI.*

In the context of UAV motion-planning, Proposition 1 permits a direct application of the ISMP under realistic assumptions, and with quantities (e.g., $\alpha_{\max}, F_{\max}, \mathcal{K}$) that can be learned from data for any closed-loop UAV controller that stabilizes about setpoints in space.

B. PI sets as function of attitude tracking error

To demonstrate this method of computing PI sets, and to get a sense of what the sets may look like for relevant parameters, consider a problem where $N = 5$ gains are randomized with magnitudes similar to those in [20]. Specifically, the gains are randomized such that $\lambda(\boldsymbol{K}_p^i) \in [\omega_n^2, \omega_n^2 + 1]^3$ and $\lambda(\boldsymbol{K}_v^i) \in [2\xi\omega_n, 2\xi\omega_n + 1]^3$ with $\omega_n = 5$ and $\xi = \sqrt{1/2}$. The mass of the micro UAV is $m = 0.1$kg, the external forces are bounded as $F_{\max} = 0.05$N, and the gravitational acceleration is $g = 9.81\text{m/s}^2$. The resulting additive disturbance in (1b) is bounded by $\Delta_{\max} = \sup_t \|\boldsymbol{\Delta}\|_2 = g(1 - \cos(\alpha_{\max})) + F_{\max}/m$, which is a function of $\alpha_{\max}$. Here, the largest feasible attitude tracking error in the sense of (3) is ≈ 0.45.

IV. Largest Ellipsoidal Safe PI Sets

Sec. III outlines a method for computing smallest ellipsoidal robust PI sets subject to Assumptions **A1**–**A3**. To formulate the RISMP, we also need to construct the largest ellipsoidal PI sets where constraints are satisfied and where the operation of the UAV is safe, i.e., free of collisions, denoted by $\mathcal{O}_{\max}$. In fact, we can only guarantee a transition from one node to the next if the set $\mathcal{O}_{\min}$ in Proposition 1 associated with one node is contained in the set $\mathcal{O}_{\max}$ of the next node. To compute $\mathcal{O}_{\max}$, we use the following Lemma.

Lemma 1 *Let $\boldsymbol{x}_e = (\boldsymbol{p} - \boldsymbol{r}; \boldsymbol{v})$ and let $V(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{r}) = \|\boldsymbol{x}_e\|_{\boldsymbol{P}}^2$ be a Lyapunov function. Let*

$$\Gamma(\boldsymbol{r}) = \frac{(\boldsymbol{c}_{\boldsymbol{p}}^\top \boldsymbol{r} - \boldsymbol{d}(\boldsymbol{r}))^2}{\begin{bmatrix} \boldsymbol{c}_{\boldsymbol{p}} \\ \boldsymbol{c}_{\boldsymbol{d}} \end{bmatrix}^\top \boldsymbol{P}^{-1} \begin{bmatrix} \boldsymbol{c}_{\boldsymbol{p}} \\ \boldsymbol{c}_{\boldsymbol{d}} \end{bmatrix}}, \tag{5}$$

then $V(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{r}) \leq \Gamma(\boldsymbol{r}) \Rightarrow \boldsymbol{c}_{\boldsymbol{p}}^\top \boldsymbol{p} + \boldsymbol{c}_{\boldsymbol{v}}^\top \boldsymbol{v} \leq \boldsymbol{d}(\boldsymbol{r})$.

Lemma 1 determines how much we can inflate the invariant set associated with a specific node of the graph while still retaining safety. We construct safe sublevel sets for the Lyapunov function in Lemma 1 from positional constraints for avoiding convex polyhedral obstacles, and from thrust constraints. Considering one input constraint and M obstacles, we compute $M + 1$ such sublevel sets, i.e., $\{\Gamma_i\}_{i=0}^M$, and the largest ellipsoidal safe sublevel set for $\boldsymbol{P}$ is $\mathcal{O}_{\max} \triangleq \{\boldsymbol{x}_e \in \mathbb{R}^6 : \mathcal{V}(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{r}) \leq V_{\max}\}$, where

$$V_{\max} = \min(\{\Gamma_i\}_{i=0}^M). \tag{6}$$

A. Polyhedral positional constraints

Consider a vertex at a point $\boldsymbol{r} \in \mathbb{R}^3$ and let $\boldsymbol{Q} = \boldsymbol{P}_{\boldsymbol{pp}} - \boldsymbol{P}_{\boldsymbol{pv}} \boldsymbol{P}_{\boldsymbol{vv}}^{-1} \boldsymbol{P}_{\boldsymbol{vp}}$ be a projection of its associated set $\mathcal{O}_{\min}$ on the positional dimensions via Schur complement. Introducing the coordinate transformation $\bar{\boldsymbol{p}} = \boldsymbol{Q}^{1/2}(\boldsymbol{p} - \boldsymbol{r})$, if $\mathcal{V}(\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{v}) \leq \Gamma$ is PI, then $\|\bar{\boldsymbol{p}}\|_2^2 \leq \Gamma$ is also PI. For all obstacles $\mathcal{S}_i = \{\boldsymbol{p} : \boldsymbol{A}_i^{\mathcal{S}} \boldsymbol{p} \leq \boldsymbol{b}_i^{\mathcal{S}}\}$, we apply the same linear transformation to obtain $\bar{\mathcal{S}}_i = \{\bar{\boldsymbol{p}} : \bar{\boldsymbol{A}}_i^{\mathcal{S}} \bar{\boldsymbol{p}} \leq \bar{\boldsymbol{b}}_i^{\mathcal{S}}\}$ with

$$\bar{\boldsymbol{A}}_i^{\mathcal{S}} = \boldsymbol{A}_i^{\mathcal{S}} \boldsymbol{Q}^{-1/2}, \quad \bar{\boldsymbol{b}}_i^{\mathcal{S}} = \boldsymbol{b}_i^{\mathcal{S}} - \boldsymbol{A}_i^{\mathcal{S}} \boldsymbol{r}. \tag{7}$$

In the transformed coordinates, we solve

$$\bar{\boldsymbol{p}}_i^\star = \underset{\bar{\boldsymbol{p}} \in \bar{\mathcal{S}}_i}{\operatorname{argmin}} \|\bar{\boldsymbol{p}}\|_2^2. \tag{8}$$

The hyperplane passing through $\bar{\boldsymbol{p}} = \bar{\boldsymbol{p}}^\star$ with normal $\bar{\boldsymbol{p}}^\star$ is

$$\boldsymbol{c}_{\boldsymbol{p},i}^\top \bar{\boldsymbol{p}} = (\bar{\boldsymbol{p}}_i^\star)^\top \bar{\boldsymbol{p}} \leq (\boldsymbol{p}_i^\star)^\top \boldsymbol{p}_i^\star \triangleq d_i. \tag{9}$$

so that for obstacle $\mathcal{S}_i$, $\Gamma_i = \|\bar{\boldsymbol{p}}_i^\star\|_2^2$. This is illustrated in a 2D projection in Fig. 1 where Γ_1 and Γ_3 are computed by the solution of (8).

B. Thrust constraints

Constraints induced by bounds on the thrust become more cumbersome to enforce in the context of Assumption **A2**, and depend on the UAV low-level controller. For the trajectory tracking control system in [21], the commanded thrust is $f = (mg\boldsymbol{e}_3 - \boldsymbol{K}_p(\boldsymbol{p} - \boldsymbol{r}) - \boldsymbol{K}_d \boldsymbol{v}) \cdot \boldsymbol{R}\boldsymbol{e}_3$, and the objective is to find a safe PI set such that $f \leq f_{\max}$. This can be done along the lines of [20], but is complicated slightly by the gains $\boldsymbol{K} \in \mathcal{K}$ being dense. To construct the level set, we first state a set of LMIs to bound the denominator in (5) for the linear constraint $f \leq f_{\max}$,

$$\lambda^\star = \underset{(\lambda_{11}, \lambda_{12}, \lambda_{22}) \in \mathbb{R}^3}{\operatorname{argmin}} (\lambda), \tag{10a}$$

subject to

$$\begin{bmatrix} \boldsymbol{P} & \bar{\boldsymbol{K}}_i \\ \bar{\boldsymbol{K}}_i & \boldsymbol{\Lambda} \end{bmatrix} \succeq \boldsymbol{0}, \ \ \forall i = 1, .., N, \tag{10b}$$

$$\begin{bmatrix} \lambda_{11}\boldsymbol{I} & \lambda_{12}\boldsymbol{I} \\ \star & \lambda_{22}\boldsymbol{I} \end{bmatrix} = \boldsymbol{\Lambda}, \tag{10c}$$

$$\lambda_{11} + 2\lambda_{12} + \lambda_{22} = \lambda. \tag{10d}$$

Lemma 2 *Given Assumption **A2**, let $\bar{\boldsymbol{K}} = \operatorname{diag}(\boldsymbol{K}_p, \boldsymbol{K}_d)$ and $\bar{\boldsymbol{K}}_i = \operatorname{diag}(\boldsymbol{K}_p^i, \boldsymbol{K}_d^i)$. There exists $\lambda > 0$ such that*

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{u} \end{bmatrix}^\top \bar{\boldsymbol{K}} \boldsymbol{P}^{-1} \bar{\boldsymbol{K}} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{u} \end{bmatrix} \leq \lambda \quad \forall \boldsymbol{u} \in \mathsf{S}^2. \tag{11}$$

The smallest λ satisfying (11) *is the solution $\lambda^\star$ of* (10)*.*

Lemma 2 is useful in defining safe robust PI sets with respect to a thrust constraint. Following [20] and using Lemma 1, we obtain a maximal level set $V \leq \Gamma_0$ such that $f(t) < f_{\max}$ for all $t \geq 0$, where $\Gamma_0 = (f_{\max} - mg)^2 m^{-2} (\lambda^\star)^{-1}$ and $\lambda^\star$ can be computed from (10).

V. Robust Invariant-Set Motion-Planning

Leveraging the smallest ellipsoidal robust PI sets (Sec. III) and inflated safe PI sets (Sec. IV) for specific setpoints $\boldsymbol{r}$, we construct a graph $\mathcal{G}$ with a vertex set $\mathcal{V}$. To simplify the exposition, let $\boldsymbol{r}_i \in \mathcal{R} \subseteq \mathbb{R}^3$ be a position setpoint. Furthermore, let $\mathcal{O}^i_{\min}$ be computed as in Sec. III for $\boldsymbol{r}_i$, and $\mathcal{O}^i_{\max}$ be the PI sets computed as in Sec. IV for $\boldsymbol{r}_i$. Thus, each vertex in $\mathcal{G}$ is associated with a triple $(\boldsymbol{r}_i, \mathcal{O}^i_{\min}, \mathcal{O}^i_{\max})$. We describe how to define the graph connectivity in an offline step in Sec. V-A, and how to solve Problem 1 by searching over the graph online in Sec. V-B. Finally, conditions for updating the reference are stated in Sec. V-C.

A. Building the graph (offline)

The connectivity of the graph can be determined directly from the triple associated with each vertex. The first step is to determine suitable reference positions, either by sampling or by a lattice-based construction as in [15]. Then, the connectivity can be defined by considering when it is safe to switch from one $\boldsymbol{r}_i$ to another $\boldsymbol{r}_j$ based on the invariant sets. If $\mathcal{O}^i_{\min} \subset \mathcal{O}^j_{\max}$, any trajectory starting in $\mathcal{O}^i_{\max}$ eventually converges to $\mathcal{O}^i_{\min}$, and it is safe to change the setpoint, as the UAV resides in $\mathcal{O}^j_{\max}$. Thus, as in standard ISMPs [15], [17], [18], we construct edges in the graph such that

$$(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E} \text{ iff } \mathcal{O}^i_{\min} \subset \mathcal{O}^j_{\max}. \tag{12}$$

In the coordinates $\bar{\boldsymbol{r}}_i = (\boldsymbol{P_{pp}} - \boldsymbol{P_{pv}} \boldsymbol{P_{vv}^{-1}} \boldsymbol{P_{vp}})^{1/2} \boldsymbol{r}_i$, the condition (12) is efficiently checked across $\mathcal{V} \times \mathcal{V}$ as

$$\|\bar{\boldsymbol{r}}_i - \bar{\boldsymbol{r}}_j\|_2 < \sqrt{V^j_{\max}} - \sqrt{V^i_{\min}}. \tag{13}$$

We can associate costs with the edges in several ways. For simplicity, define the weights $c_{ij} = \|\bar{\boldsymbol{r}}_i - \bar{\boldsymbol{r}}_j\|_2$. It is relatively common to find nodes that have a single edge leading into a vertex, but no edges leading back to any other vertex. To make the theoretical statements concise, these vertices are pruned from $\mathcal{G}$ along with any disconnected vertices.

B. Modifying and searching the graph (online)

Searching for a path from an initial node to a terminal node is fast. For this purpose, we can use a standard Dijkstra algorithm. We need to choose the initial and terminal vertices before initializing the graph search. The initial vertex is

$$\min_{\mathcal{V}_i \in \mathcal{V}} \|\boldsymbol{x}(t_0) - (\boldsymbol{r}_i; \boldsymbol{0})\|^2_{\boldsymbol{P}}, \tag{14}$$

and for a requested terminal position $\boldsymbol{p}_\infty = \lim_{t \to \infty} \boldsymbol{p}(t)$, we add a new node $\mathcal{V}_j$ to the graph with $\boldsymbol{r}_j = \boldsymbol{p}_\infty$. For this new node, an inflated safe PI set is determined as in Sec. IV, and edges are added to the neighboring nodes prior to executing the graph search algorithm. The output of the algorithm is a path consisting of a sequence of $K \geq 1$ vertices.

C. Switching between set-points (online)

Given the path in the graph as a sequence of K vertices, next we describe the conditions for switching between the sequence of piecewise constant setpoints associated to the vertices, see also [15], [17], [18]. With a slight abuse of notation, we re-number the vertices of the graph according to their sequence in the path, i.e., $\mathcal{V}_1$ is the starting vertex and $\mathcal{V}_K$ is the terminal vertex. If at any time $t^\star \geq t_{k-1}$, $\boldsymbol{x}(t^\star) \in \mathcal{O}^k_{\max}$, we let $t_k = t^\star$ and start tracking the next reference, $\boldsymbol{r}(t) = \boldsymbol{r}_k$. Here, (12) implemented by (13) ensures that $t_k - t_{k-1}$ is finite and that the RISMP solves Problem 1.

While the detailed analysis of the properties is beyond the scope of this paper, the resulting motion planner provides safety with respect to the obstacles, since for all $t \geq t_0$, $\boldsymbol{x}(t) \in \bigcup_{k=1}^K \mathcal{O}^k_{\max}$ and $(\bigcup_{k=1}^K \mathcal{O}^k_{\max}) \cap (\bigcup_{i=1}^M \mathcal{S}_i) = \emptyset$, and constraint satisfaction, since for all $t \geq t_0$, $f(t) \leq f_{\max}$. Furthermore, the stabilizing design of the gains in (4) ensures that any neighborhood of the goal is entered in finite time.

The success or failure in computing a feasible state trajectory between any initial condition $\boldsymbol{x}(t_0) \in \bigcup_i \mathcal{O}^i_{\max}$ and target node in the graph depends entirely on the connectivity of $\mathcal{G}$. For instance, if two rooms are connected by a passage that is too narrow to pass subject to the problem data, the graph will not be fully connected, and the RISMP will not be able to safely fly the UAV from one room to another. The graph connectivity can be studied offline, before running any UAV mission. Such prior statements about feasibility and operational limits makes the method well suited for tasks such as indoor factory automation and surveying (e.g., [1]).

VI. Simulation Examples

To demonstrate the RISMP, we consider a scenario in which the UAV is to navigate through a complex environment comprised of 13 polyhedral obstacles which overlap to form a corridor and a small room, as depicted in Fig. 2. The task is to compute and fly a trajectory for the UAV from an initial point in the corridor to a target node in the inner room.

A. Offline processing

We consider a Crazyflie 2.1 UAV with the nominal stock cascaded controller structure that results yields a system response well approximated by a second-order PD-controlled system in (1) with nominal gains

$$\boldsymbol{K}^\star_p = \mathrm{diag}(7.78, 7.38, 11.30), \tag{15a}$$
$$\boldsymbol{K}^\star_v = \mathrm{diag}(3.28, 3.27, 3.75), \tag{15b}$$

To constitute $\mathcal{K}$ for the minimal PI synthesis, we sample $N = 10$ gains

$$(\boldsymbol{K}^i_p, \boldsymbol{K}^i_v) = (\boldsymbol{K}^\star_p, \boldsymbol{K}^\star_v)\mathrm{diag}(\boldsymbol{\eta}), \;\; \boldsymbol{\eta} \sim \mathcal{U}([0.9, 1.1]^6). \tag{16}$$

We consider the model parameters $g = 9.81\text{m/s}^2$, $m = 0.03\text{kg}$, $\alpha_{\max} = 0.1\text{rad}$, $f_{\max} = 2mg = 0.2943\text{N}$, $F_{\max} = 0.02\text{N}$. The synthesis of a common Lyapunov function and robust PI set as per (10b) yields a level set $V_{\min} = 0.24$ In our implementation, this is computed by `CVX` with `SDPT3` and default numerical tolerances. The graph is constructed

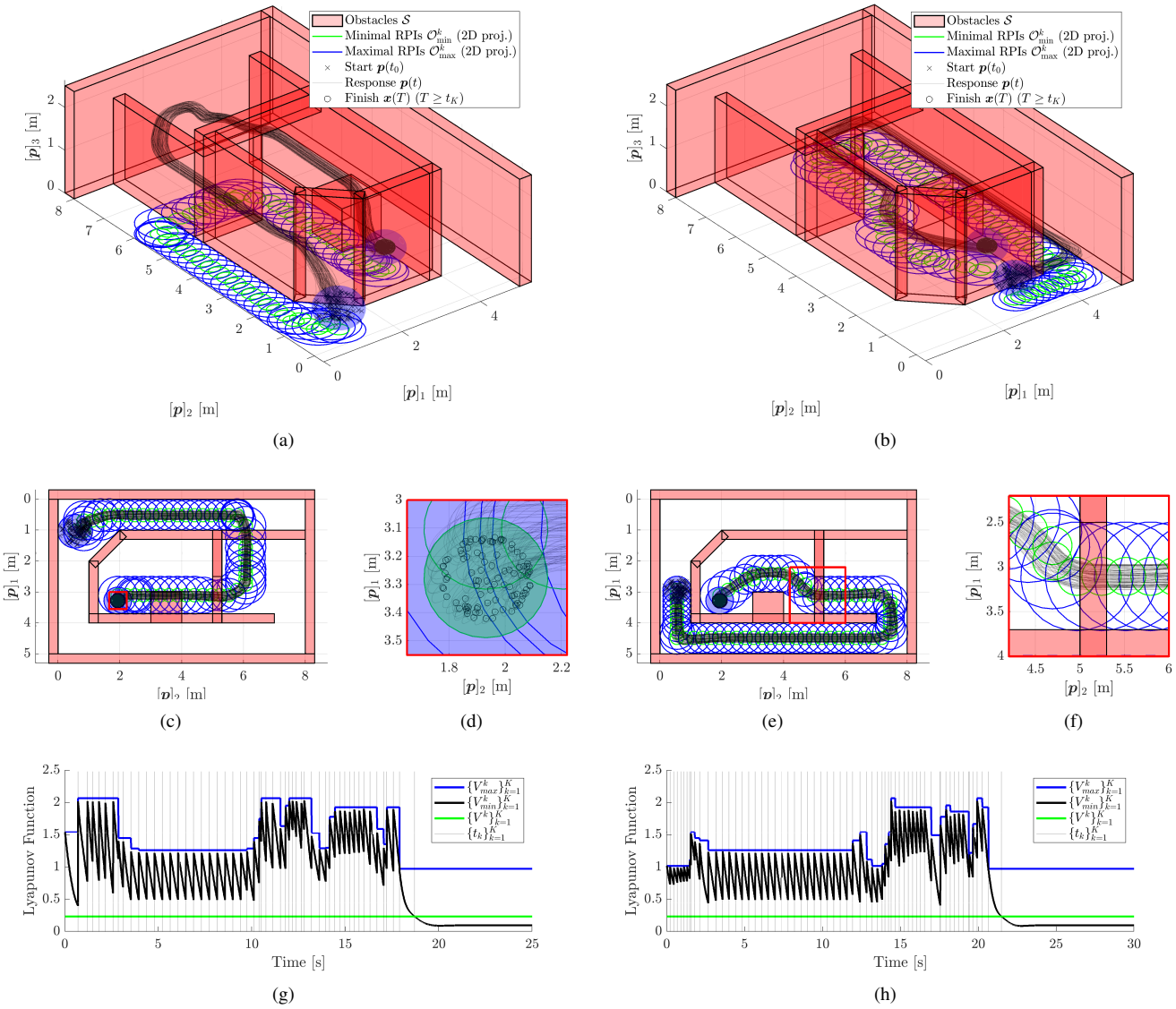

Fig. 2. Two scenarios (A,B) where the RISMP is used to compute a solution from a position close to an initial node in the corridor to a target node in the inner room. (a), (b) The geometry of the world (red) with the 2D-projections of the invariant sets in the $[\boldsymbol{p}]_1[\boldsymbol{p}]_2$-plane and 100 realizations each of the UAV state trajectory subject to different initial conditions $\boldsymbol{x}(t_0) \in \text{bnd}(\mathcal{O}^1_{\max})$ at the time $t_0 = 0$ and different realizations of the disturbance (black). (c), (e) Top view of the same plots. (d) Zoom-in on the position of the UAV at the terminal time $T > t_K$ in scenario A. (f) Zoom in on the position of the UAV in scenario B, showing how the UAV navigates the doorway. (g),(h) Time-series plots of the Lyapunov function in scenarios A and B, respectively. The plots illustrate the inflated safe set $\mathcal{O}^k_{\max}$ by $V^k_{\max}$ (blue), the smaller ultimate set $\mathcal{O}^k_{\min}$ by $V^k_{\min}$ (green), and the exponential decay of the Lyapunov function $V^k(t)$ associated with the node $\mathcal{V}_k$ in the path computed by the RISMP. The switching instants $\{t_k\}_{k=1}^K$ are shown in gray.

as in Sec. V-A starting from an initial set of 9000 nodes (30×30×10) equidistantly sampled in space (5×8×2.5m). After the collision checks and removal of infeasible nodes, a total of 4019 nodes remain with an average of 6.7 directed edges leaving each node. This computation takes approximately 90s in our non-optimized Matlab implementation, using `quadprog` for solving the QPs in (8). Using an X-bit float representation, the graph of the size used in the simulations can be stored in as little as

$$(2 \times 16 + X) \times |\mathcal{E}| + 4 \times X \times |\mathcal{V}| + 37 \times X \text{ bits}, \quad (17)$$

which for $X = 32$bit results in 294.13kB. While too large to store on the Crazyflie 2.1 due to its limited memory, it is well within the total usable memory when operating the UAV with an onboard SD-card expansion deck.

B. Online Processing and Simulation Setup

In the simulations, we consider two different scenarios that primarily differ in the initial conditions of the UAV, referred to as Scenario A and B, respectively. A total of 100 simulations are run in each case, and for each such run, the following simulation parameters are varied:

- the controller gains $\boldsymbol{K} \sim \mathcal{U}(\mathcal{K})$;
- the initial conditions $\boldsymbol{x}(t_0) \sim \mathcal{U}(\text{bnd}(\mathcal{O}^1_{\max}))$;
- the input disturbance realization $\boldsymbol{\Delta}(t)$.

To compute an input disturbance, we let $\tilde{\boldsymbol{R}} = e^{\boldsymbol{S}(\alpha_{\max}\boldsymbol{u})}$ be a constant maximal perturbation with a rotation vector $\boldsymbol{u} \sim \mathcal{U}(\mathrm{S}^2)$ that is permitted by Assumption **A3**, take $\boldsymbol{f}$ to point in the direction $(\boldsymbol{I} - \tilde{\boldsymbol{R}})\boldsymbol{e}_3$, and define $\boldsymbol{\Delta}$ as a constant disturbance that maximizes $\|\boldsymbol{\Delta}\|_2$ subject to the sampled $\boldsymbol{u}$. In the simulations, the graph search is done using a standard Dijkstra algorithm in the `Boost Graph Library` [24]. In this setting, it takes less than 2ms to find an optimal solution in both scenarios, with $K = 54$ in scenario A and $K = 64$ in scenario B. The resulting state trajectories from 100 simulations are shown in Fig. 2.

C. Discussion

As shown in Fig.2-(a),(b) in all 200 simulations of Scenario A, B, the trajectories are collision free and reach the target invariant set $\mathcal{O}^K_{\min}$ in less than 30s, and the terminal state $\boldsymbol{x}(T) \in \mathcal{O}^K_{\min}$ for all $T \geq t_K$. See also the top view of the trajectories in Fig.2-(c),(d). When the initial conditions are shifted in scenario A and B, we get very different paths. In scenario A, the UAV navigates over a wall with small clearance, and in taking this route, it maintains its height and moves above the small box in the inner room. Instead, in scenario B, the UAV maintains its low height, see Fig.2-(f) , chooses a different corridor, and goes around the small box in the inner room.

In all simulations, it is verified that $\boldsymbol{x}(t) \in \bigcup_{k=1}^{K} \mathcal{O}^k_{\max}$ at all times, as shown by the time-series plots of two of the 200 Monte-Carlo runs Fig. 2-(g),(h), where on each interval $t \in [t_k, t_k + 1]$, the Lyapunov function $V^k(t)$ (black) is bounded by the associated $V^k_{\max}$ (blue) rendering the trajectory safe. Furthermore, $V^k(t)$ decays exponentially on each such interval, and enters the terminal minimal PI set at t_K, which differs in scenario A and B and also among the realizations in each scenario due to the uncertainty.

VII. Conclusions

We proposed a robust ISMP for quad-rotor UAVs using an LMI-based computation of small ellipsoidal robust PI sets subject to: (i) polytopic uncertainty in the controller gains, (ii) additive input disturbances, and (iii) bounded attitude tracking errors. We build upon the previous work in [15], [20], and additionally provide an efficient method of computing the inflated safe robust PI sets subject to polyhedral obstacles and thrust constraints. The latter are defined as sub-level sets to the Lyapunov functions used in constructing the robust PI sets.

The method performs a majority of its computations offline, and we demonstrate that a safe path can be computed online in as little as 2ms for a complex indoor navigation scenario with approximately 100m^3 of free-space. Furthermore, the resulting graph is small enough that it may be put in memory in micro UAVs such as the Crazyflie 2.1.

Future works will validate these theoretical results experimentally using a Crazyflie 2.1, and possibly consider more complex costs for the graph edges.

References

[1] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, "Unmanned aerial vehicle path planning and control assisted by augmented reality: the case of indoor drones," *Int. J. Production Research*, 2023.

[2] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *Int. Conf. Intelligent Robots and Systems*, 2019, pp. 1934–1940.

[3] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *Int. Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[4] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics research*. Springer, 2016, pp. 649–666.

[5] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," in *Int. Conf. Robotics and Automation*, 2015, pp. 42–49.

[6] M. Greiff, A. Vinod, S. Nabi, and S. Di Cairano, "Quadrotor motion planning in stochastic wind fields," in *American Control Conference*, 2023, pp. 4619–4625.

[7] S. M. Lavalle, *Planning Algorithms*. Cambridge Univ. Press, 2006.

[8] L. Quan, L. Han, B. Zhou, S. Shen, and F. Gao, "Survey of UAV motion planning," *Cyber-sys. and Rob.*, vol. 2, no. 1, pp. 14–21, 2020.

[9] A. Weiss, C. Petersen, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Safe positively invariant sets for spacecraft obstacle avoidance," *J. of Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 720–732, 2015.

[10] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *55th IEEE Conf. Decision and Control*, 2016, pp. 5986–5991.

[11] C. Danielson, K. Berntorp, S. Di Cairano, and A. Weiss, "Motion-planning for unicycles using the invariant-set motion-planner," in *2020 American Control Conference*, 2020, pp. 1235–1240.

[12] B. Convens, K. Merckaert, B. Vanderborght, and M. M. Nicotra, "Invariant set distributed explicit reference governors for provably safe on-board control of nano-quadrotor swarms," *Frontiers in Robotics and AI*, vol. 8, p. 129, 2021.

[13] N. Michel, "Invariant set design for the constrained control of a quadrotor," Ph.D. dissertation, Université Paris-Saclay, 2020.

[14] A. Weiss, C. Danielson, K. Berntorp, I. Kolmanovsky, and S. Di Cairano, "Motion planning with invariant set trees," in *Conf. on Cont. Tech. and App. (CCTA)*, 2017, pp. 1625–1630.

[15] C. Danielson, K. Berntorp, A. Weiss, and S. Di Cairano, "Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4456–4463, 2020.

[16] A. Weiss, F. Leve, M. Baldwin, J. R. Forbes, and I. Kolmanovsky, "Spacecraft constrained attitude control using positively invariant constraint admissible sets on SO(3)xR3," in *American Control Conference*, 2014, pp. 4955–4960.

[17] C. Danielson, J. Kloeppel, and C. Petersen, "Spacecraft attitude control using the invariant-set motion-planner," *IEEE Control Systems Letters*, vol. 6, pp. 1700–1705, 2021.

[18] K. Berntorp, R. Bai, K. F. Erliksson, C. Danielson, A. Weiss, and S. Di Cairano, "Positive invariant sets for safe integrated vehicle motion planning and control," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 112–126, 2019.

[19] M. M. Nicotra and E. Garone, "The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems," *IEEE Control Sys. Mag.*, vol. 38, no. 4, pp. 89–107, 2018.

[20] M. M. Nicotra, R. Naldi, and E. Garone, "A robust explicit reference governor for constrained control of unmanned aerial vehicles," in *American Control Conference*, 2016, pp. 6284–6289.

[21] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE Conf. Decision and Control*, 2010, pp. 5420–5425.

[22] M. Greiff, "Nonlinear control of unmanned aerial vehicles: Systems with an attitude," Ph.D. dissertation, Lund University, 2021.

[23] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[24] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual*. Pearson Education, 2001.