

# Invariant Set Planning for Quadrotors: Design, Analysis, Experiments

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

TR2025-010 January 08, 2025

## Abstract

We propose a motion planner for quadrotor unmanned aerial vehicles (UAVs) implemented as a graph search over robust positively invariant (PI) sets. We model the positional error dynamics of the quadrotor in closed-loop with an onboard controller as a second-order system with polytopic uncertainty in the gains. We also consider bounded attitude tracking errors and additive input disturbances. We propose a method for computing ellipsoidal robust PI sets using linear matrix inequalities that are expanded such that all trajectories therein remain safe, i.e., do not intersect obstacles and ensure satisfaction of UAV constraints. We construct a graph where the vertices are equilibrium positions and the edges are transitions between equilibria occurring within the PI sets. Hence, a graph search returns a sequence of setpoints steering the UAV from an initial position to a target, while remaining within the safe invariant sets. We show that, subject to the properties of the graph, from any initial position within an invariant set, any robust PI set in the graph is reachable in finite time. The graph construction is offline, and the online graph search and plan execution are simple and fast, thus allowing for real-time planning. We demonstrate the method in extensive simulations and in experiments with a Crazyflie 2.1 quadrotor.

*IEEE Transactions on Control Systems Technology 2025*



# Invariant Set Planning for Quadrotors: Design, Analysis, Experiments

Marcus Greiff, Himani Sinhar, Avishai Weiss, Karl Berntorp and Stefano Di Cairano

**Abstract**—We propose a motion planner for quadrotor unmanned aerial vehicles (UAVs) implemented as a graph search over robust positively invariant (PI) sets. We model the positional error dynamics of the quadrotor in closed-loop with an onboard controller as a second-order system with polytopic uncertainty in the gains. We also consider bounded attitude tracking errors and additive input disturbances. We propose a method for computing ellipsoidal robust PI sets using linear matrix inequalities that are expanded such that all trajectories therein remain safe, i.e., do not intersect obstacles and ensure satisfaction of UAV constraints. We construct a graph where the vertices are equilibrium positions and the edges are transitions between equilibria occurring within the PI sets. Hence, a graph search returns a sequence of setpoints steering the UAV from an initial position to a target, while remaining within the safe invariant sets. We show that, subject to the properties of the graph, from any initial position within an invariant set, any robust PI set in the graph is reachable in finite time. The graph construction is offline, and the online graph search and plan execution are simple and fast, thus allowing for real-time planning. We demonstrate the method in extensive simulations and in experiments with a Crazyflie 2.1 quadrotor.

**Index Terms**—Motion Planning, Invariant Sets, Constrained Control, Uncertain Systems, Quadrotors, LMIs.

## I. INTRODUCTION

UNMANNED aerial vehicles (UAVs) are increasingly investigated in industrial applications, such as indoor surveying and factory automation, where the environment is well known and largely static [1], [2]. Fully autonomous UAV operations require motion-planning (MP) with rigorous safety guarantees that enable fast planning, and replanning. Several methods can be considered for this purpose, see, e.g., [3]–[6] and the references therein. For factory automation [2] or construction inspection [1], we seek planners that: provide theoretical guarantees of safety in the presence of disturbances and system modeling errors; ensure that relevant variables, such as the maximal thrust of the UAV, are kept within their bounds; and report *a priori* if safe flight in an environment is possible. The last point is particularly relevant, as it can be used to study the feasibility of system deployment, and to determine appropriate operational limitations.

The dominating paradigm for UAV motion-planning leverages convex optimization [5], [7]–[13]. The minimum-snap planners in [8], [11]–[13] were designed to facilitate aggressive

maneuvering [9], [12], and impressive efforts have been made to ensure favorable scaling with the size of the workspace and the number of obstacles [10]. These methods generally do not account for structured uncertainties and disturbances acting on the system, or assume that these take very particular forms [8]. To ensure safety, such methods generally inflate the obstacles, but determining the amount of inflation that guarantees safety is a nontrivial task.

When considering disturbances and structured uncertainty, it is challenging to ensure rigorous safety guarantees with the planning methods that are based purely on convex optimization without significantly increasing the computational burden. This motivates our exploration of graph-based planning, where instead of optimizing over trajectories in space, the UAV motions are encoded through a graph search by highly efficient algorithms [3]. Often, graph-based planning associates motions of the UAV to graph edges, and parameterizes them *explicitly* [4], e.g., as motion primitives. Instead, we consider the Invariant-Set Motion Planner (ISMP) [14]–[16], in which the motions are encoded *implicitly* as the closed-loop response to a sequence of setpoint commands leveraging positive invariant (PI) sets, see Fig. 1. Such methods can provide guarantees on robust constraint satisfaction and ensure convergence to a target invariant set [17]. In this paper, we seek to extend the framework in [18] and construct a Robust ISMP (RISMP) tailored for UAVs.

ISMP approaches have been proposed for various applications [14]–[16], [18]–[21], most notably automated driving [16], [19], [22], and spacecraft orbit [14] and attitude control [20], [21], [23]. Related invariance-based methods have been suggested for UAVs [24], such as invariance-based explicit reference governors (ERGs) [25], [26], experimentally validated in [27], [28]. Due to the large state-space of the quadrotor UAV and computational challenges in computing its associated robust positive invariant sets, RISMP for UAVs has not been developed yet, despite its appealing properties. Here, based on some preliminary results from ERG [26] we extend the ideas in [18] to develop an RISMP framework for UAVs that addresses these computational challenges.

Although extensions to account for moving or initially unknown obstacles are possible, see, e.g., [14], [22], the RISMP method is particularly well suited for tasks in confined spaces of moderate size containing several obstacles that are largely static. This includes transport or monitoring in a factory setting along the lines of [2]. Here, the extreme computational scaling of the methods in [10], [13] is not required, and similarly, there is no need for the aggressive maneuvers in [12]. Instead, safety guarantees are critical, in particular when considering

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

M. Greiff was with MERL at the time of this research, and is now with Toyota Research Institute (e-mail: marcus.greiff@tri.global).

H. Sinhar is with the Sibley School of Mechanical and Aerospace Eng. at Cornell University, Ithaca, NY (e-mail: hs962@cornell.edu). She was an intern at MERL at the time of this research.

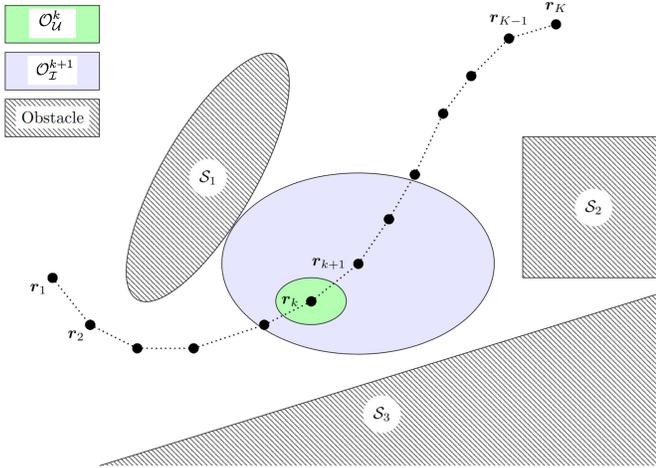


Fig. 1. Core idea of the RISMP: A sequence of reference points  $\{r_k\}_{k=1}^K$  are computed using a graph search, where the transition  $r_k \rightarrow r_{k+1}$  occurs if and only if the set  $\mathcal{O}_U^k$  associated with  $r_k$  (green) is contained in the interior of the larger set  $\mathcal{O}_I^{k+1}$  associated with  $r_{k+1}$  (blue). In addition to the ellipsoidal ( $S_1$ ), polytopic ( $S_2$ ), and wall ( $S_3$ ) obstacles, the size of the safe set (blue) is also constrained by bounds on the control inputs.

the realistic closed-loop disturbances and uncertainties of low-cost UAVs. ERG approaches may also be suitable for these purposes, but they are reliant on design choices such as the navigation field, which limit objectives that can be encoded in the planning problem [28].

### A. Contributions

The main contribution of this paper is the design of an RISMP for quadrotor UAVs, and its validation. We provide:

- A method for computing ellipsoidal robust PI (RPI) sets for quadrotor UAVs using linear matrix inequalities (LMIs), that accounts for bounded additive input disturbances, bounded attitude tracking errors, and polytopic uncertainties in the positional error dynamics.
- A specialized RISMP that constructs a graph from a finite collection of RPI sets offline, and performs online planning using simple search algorithms with a priori guarantees of safety and finite-time convergence to a terminal set, subject to known properties of the graph.
- A validation of the designed RISMP in simulation and experiments using the Crazyflie UAV, demonstrating its theoretical properties in practice.

Initial studies on RISMP appeared in [29], which focuses on describing the motion-planning algorithm. Compared to [29], this paper refines the algorithm and details all the design steps, especially the invariant set construction that handles uncertainty, describes and proves the properties of the algorithm, and validates the algorithm in simulations and experiments.

In the rest of this paper, after some preliminaries in Sec. II, we introduce the UAV models in Sec. III, and the motion planning problem and the general RISMP algorithm we use to solve it in Sec. IV. Secs. V–VII describe the design of the different components of the RISMP algorithm, Sec. VIII reports our simulation results, Sec. IX reports the experimental

results with the Crazyflie quadrotor, and in Sec. X we provide the conclusions.

## II. PRELIMINARIES

$\mathbb{R}$ ,  $\mathbb{R}_+$ ,  $\mathbb{Z}$ ,  $\mathbb{Z}_+$  are the sets of real, positive real, integer, positive integer numbers, and we denote intervals with notation such as  $\mathbb{Z}_{[1,N]} = \{1, \dots, N\}$ . Vectors are denoted as  $\mathbf{x} \in \mathbb{R}^n$ ,  $[\mathbf{x}]_i$  is the  $i^{\text{th}}$  element,  $\mathbf{e}_i$  is a unit vector where  $[e_i]_i = 1$ , and for  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{y} \in \mathbb{R}^m$ ,  $(\mathbf{x}; \mathbf{y}) = (\mathbf{x}^\top, \mathbf{y}^\top)^\top \in \mathbb{R}^{m+n}$ . The identity matrix is  $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ , and  $\star$  is a block defined by the context, and  $\text{diag}(\mathbf{A}, \mathbf{B})$  is a block diagonal matrix of its arguments. We let  $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top \mathbf{x}$ ,  $\|\mathbf{x}\|_P^2 = \mathbf{x}^\top \mathbf{P} \mathbf{x}$ , and  $(\mathbb{S}_+^n)$   $\mathbb{S}_+^n$  be the cone of  $n \times n$  positive (semi)definite matrices. The maximum and minimum eigenvalues of a real, symmetric matrix  $\mathbf{M} \in \mathbb{R}^{m \times m}$  are  $\bar{\lambda}(\mathbf{M}), \lambda(\mathbf{M}) \in \mathbb{R}$ , respectively. The set of rotations is  $\text{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} | \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}$ ,  $\mathcal{S}(\boldsymbol{\nu})$  is the skew symmetric matrix from vector  $\boldsymbol{\nu}$ , where  $\mathcal{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$ . A directed graph  $\mathcal{G} = (\mathbb{V}, \mathbb{E})$  is a set of vertices (or nodes)  $\mathbb{V} = \{v_1, \dots, v_{N_v}\}$  with ordered pairs  $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$  called edges. Vertices  $v_i, v_j \in \mathbb{V}$  are adjacent if  $(v_i, v_j) = \epsilon_{ij} \in \mathbb{E}$  is an edge, and a path  $\mathcal{P}$  is a sequence of adjacent vertices,  $\mathcal{P} = (v_{\sigma(k)})_{k=1}^K$  where  $\sigma: \mathbb{Z}_+ \rightarrow \mathbb{Z}_+$  defines the vertex sequence. We let  $E \sim \mathcal{U}(\mathcal{M})$  be a uniform selection of an element  $E \in \mathcal{M}$ , with uniform probability.

**Definition 1 (Ellipsoidal Set)** An ellipsoidal set in  $\mathbb{R}^n$  with center  $\mathbf{c} \in \mathbb{R}^n$ , inverse shape matrix  $\mathbf{P} \in \mathbb{S}_+^n$  and level  $\rho$  is

$$\mathcal{E}(\mathbf{c}, \mathbf{P}, \rho) = \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x} - \mathbf{c}\|_{\mathbf{P}}^2 \leq \rho\}$$

This is an over-parametrization, as the parameters  $\mathbf{P}$  and  $\rho$  can be combined. However, keeping these separate will be useful in the construction of the graph.

**Definition 2 (Polytopic Set)** Given a set of points  $\{\boldsymbol{\theta}_h\}_{h=1}^N$ , a polytopic set is the convex hull

$$\text{Co}(\{\boldsymbol{\theta}_h\}_{h=1}^N) = \left\{ \sum_{h=1}^N \zeta_h \boldsymbol{\theta}_h \mid \sum_{h=1}^N \zeta_h = 1, \zeta_h \geq 0, \forall h \in \mathbb{Z}_{[1,N]} \right\}.$$

**Definition 3 (Robust positive invariant)** Let  $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\Delta})$ , where  $\mathbf{x} \in \mathbb{R}^n$  is the state and  $\boldsymbol{\Delta} \in \mathbb{D} \subset \mathbb{R}^\Delta$  is the disturbance, a robust positive invariant (RPI) set  $\mathcal{O}$  is such that  $\mathbf{x}(t_0) \in \mathcal{O} \Rightarrow \mathbf{x}(t) \in \mathcal{O}$ , for all  $\boldsymbol{\Delta}(t) \in \mathbb{D}$  for all  $t \geq t_0$ .

**Definition 4 (Ultimate and Inflated Sets)** For the system  $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\Delta}(t))$ , two ellipsoidal RPI sets  $\mathcal{O}_U = \mathcal{E}(\mathbf{c}, \mathbf{P}, \rho_U)$ ,  $\mathcal{O}_I = \mathcal{E}(\mathbf{c}, \mathbf{P}, \rho_I)$  such that  $\rho_I > \rho_U$ , are called ultimate set and inflated set, respectively, if for all  $\mathbf{x}(t_0) \in \mathcal{O}_I$ ,  $\lim_{t \rightarrow \infty} \mathbf{x}(t) \in \mathcal{O}_U$ .

The conditions in Definition 4 imply that  $\mathcal{O}_U \subset \mathcal{O}_I$  and that any ellipsoid  $\mathcal{E}(\mathbf{c}, \mathbf{P}, (1 + \varepsilon)\rho_U) \subset \mathcal{O}_I$  where  $\varepsilon > 0$  is finite, is entered in finite time from any  $\mathbf{x} \in \mathcal{O}_I$ . In what follows, ultimate and inflated sets are ellipsoidal sublevel sets of a quadratic function  $V = \|\mathbf{x} - \mathbf{c}\|_{\mathbf{P}}^2$ .

**Definition 5 (Projected Inflated Set)** For an inflated set  $\mathcal{O}_I = \mathcal{E}(\mathbf{c}, \mathbf{P}, \rho_I)$ , with inverse shape matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \star & \mathbf{P}_{22} \end{bmatrix} \in \mathbb{S}_+^n,$$

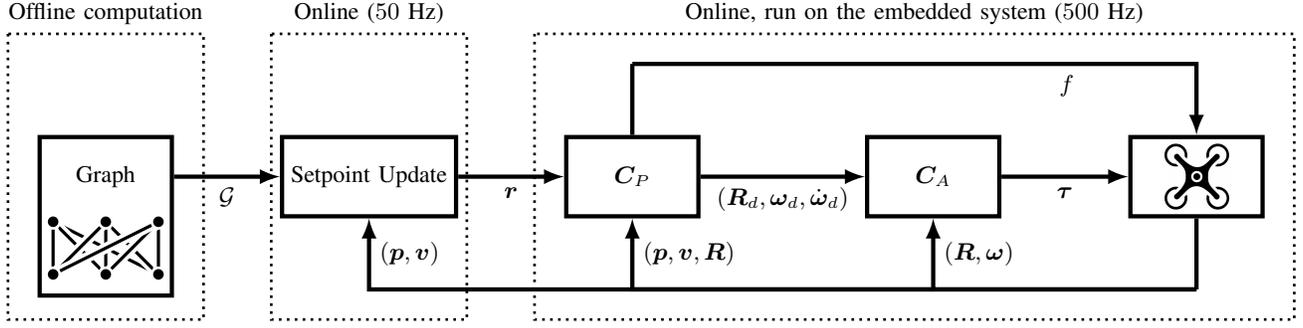


Fig. 2. Sketch of the UAV control system considered in this paper. The RISMP is defined by the construction of the graph and conditions for updating the setpoint  $r_k$ . The figure depicts the controller in [30] implemented as a stabilizing controller, here partitioned into a positional controller  $C_P$  whose gains are not perfectly known, and an attitude controller  $C_A$ , which is not ideal and results in a bounded attitude tracking error  $\tilde{\mathbf{R}} = \mathbf{R}_d^\top \mathbf{R} \in \text{SO}(3)$ .

with  $\mathbf{P}_{11} \in \mathbb{S}_{++}^3$ , we define a projection on  $\mathcal{R}$  as

$$\overline{\mathcal{O}}_{\mathcal{I}} = \mathcal{E}(\mathbf{r}, \mathbf{P}_{11} - \mathbf{P}_{12} \mathbf{P}_{22}^{-1} \mathbf{P}_{21}, \rho_{\mathcal{I}}) \subset \mathcal{R}, \quad (1)$$

where  $\mathbf{r} \in \mathcal{R}$  is the position sub-vector within  $\mathbf{c} \in \mathcal{R} \times \mathbb{R}^{n-3}$ .

### III. MODELING AND PROBLEM FORMULATION

We consider the full nonlinear UAV dynamics from the Euler-Lagrange equation [31] with a configuration manifold  $(\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times \text{SO}(3)$ ,

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (2a)$$

$$m\dot{\mathbf{v}} = T\mathbf{R}\mathbf{e}_3 - m\mathbf{g}\mathbf{e}_3, \quad (2b)$$

$$\dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\boldsymbol{\omega}), \quad (2c)$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \mathbf{S}(\mathbf{J}\boldsymbol{\omega})\boldsymbol{\omega} + \boldsymbol{\tau}, \quad (2d)$$

where  $\{G\}$  is a global frame with basis  $\{e_i\}_{i=1}^3$ , and  $\{B\}$  is the UAV body frame with basis  $\{e_i^B\}_{i=1}^3$ , where  $[e_1, e_2, e_3] = \mathbf{R}^\top [e_1^B, e_2^B, e_3^B]$ ,  $\mathbf{p} \in \mathcal{R}$  is the position in the global frame  $\{G\}$ ,  $\mathbf{v} \in \mathbb{R}^3$  is the velocity in  $\{G\}$ ,  $\mathbf{R} \in \text{SO}(3)$  is the attitude,  $\boldsymbol{\omega} \in \mathbb{R}^3$  is the attitude rate in  $\{B\}$ ,  $T > 0$  is the rotor thrust,  $\boldsymbol{\tau}$  is the torque in  $\{B\}$ . The thrust is upper bounded by  $T \leq T_{\max}$ , and the inertia matrix  $\mathbf{J} \in \mathbb{S}_{++}^3$ , mass  $m > 0$ , and gravitational acceleration  $g > 0$  are parameters in (2).

Usually, the UAV control system is structured in a fast attitude control loop, which operates on references computed in a positional control loop [30]. PD/PID-like controllers are often applied to a tracking error on special Euclidean groups [30], [32]–[34], for both stabilization and tracking. If there are discrete changes in setpoints, this will generate bounded attitude tracking errors that act as disturbances on the UAV motion. The overall control system architecture considered in this paper is shown in Fig. 2, with  $C_P$  denoting a positional controller and  $C_A$  denoting an attitude controller.

Therefore, it is natural to consider the positional subsystem of (2) for motion-planning to account for realistic attitude tracking errors due to non-ideal attitude control, see, e.g., [26], and its validation [27]. Specifically, for a position reference  $\mathbf{r} \in \mathbb{R}^3$  with corresponding equilibrium  $\bar{\mathbf{x}}(\mathbf{r})$ , we consider the error dynamics of (2) in closed-loop with a geometric

proportional derivative (PD) controller [30], [34] for setpoint stabilization (see, e.g., [26]).

$$\ddot{\mathbf{p}} = -\tilde{\mathbf{R}}^\top \mathbf{K}_p(\mathbf{p} - \mathbf{r}) - \tilde{\mathbf{R}}^\top \mathbf{K}_v \mathbf{v} + \boldsymbol{\Delta}, \quad (3a)$$

$$\boldsymbol{\Delta} = m^{-1} \mathbf{f} + g(\mathbf{I} - \tilde{\mathbf{R}})\mathbf{e}_3, \quad (3b)$$

where  $\mathbf{f} \in \mathbb{R}^3$  is the external disturbance force,  $\tilde{\mathbf{R}} \in \text{SO}(3)$  is the attitude tracking error,  $\mathbf{K}_v \in \mathbb{S}_{++}^{3 \times 3}$ ,  $\mathbf{K}_p \in \mathbb{S}_{++}^{3 \times 3}$ . Here,  $\tilde{\mathbf{R}} = \mathbf{R}_d^\top \mathbf{R}$ , where  $\mathbf{R}_d$  is a commanded rotation and  $\mathbf{R}$  is the response, see Fig. 2, so that perfect attitude tracking is achieved when  $\tilde{\mathbf{R}} = \mathbf{I}$ , i.e.,  $\mathbf{R}_d = \mathbf{R}$ , see e.g., [30]. For our purposes, it is relevant to consider the Euler axis angle of the attitude error,  $\alpha \triangleq \max_{\|\boldsymbol{\nu}\|_2=1} \arccos(\boldsymbol{\nu}^\top \tilde{\mathbf{R}}\boldsymbol{\nu})$ .

**Assumption 1** Attitude error  $\tilde{\mathbf{R}}$  and disturbance force  $\mathbf{f}$  have known bounds  $\sup_{t \geq t_0} |\alpha(t)| \leq \alpha_{\max}$ ,  $\sup_{t \geq t_0} \|\mathbf{f}(t)\|_2 \leq F_{\max}$ .

The bounds on attitude errors and external forces result in a bound on the additive disturbance in (3b),  $\|\boldsymbol{\Delta}\| \leq \Delta_{\max}$ . Unlike [26], we do not assume perfect knowledge of the feedback gains  $\mathbf{K}_v$ ,  $\mathbf{K}_p$ . This may seem counterintuitive, but in practice the full control law of the UAV may be a black/gray box, or it may be hard to model compactly due to switching modes, nonlinearities, and high frequency components. Adding uncertainty enables representing different UAV responses about some approximate, nominal control law.

**Assumption 2** The gains of (3) are not known, but satisfy

$$\mathbf{K} \triangleq (\mathbf{K}_p, \mathbf{K}_v) \in \text{Co}(\{(\mathbf{K}_p^h, \mathbf{K}_v^h)\}_{h=1}^N) \triangleq \mathcal{K}, \quad (4)$$

for known elements  $\{(\mathbf{K}_p^h, \mathbf{K}_v^h)\}_{h=1}^N$ .

**Assumption 3** For all, possibly time-varying, values  $\mathbf{K} \in \mathcal{K}$ , (3) is uniformly asymptotically stable (UAS) when  $\boldsymbol{\Delta} = \mathbf{0}$ .

Assumption 3 is sufficient to ensure that the equilibrium point is uniformly exponentially stable [35, Theorem 5.4], see [36] for the definition of UAS. By converse Lyapunov theorems, the trajectories of (3) are uniformly ultimately bounded when  $\boldsymbol{\Delta}$  is non-vanishing [36, Lemma 9.3]. Thus, there exist an ellipsoidal ultimate set to the interior of which all trajectories of (3) converge.

The UAV may be subject to additional constraints, and we impose a constraint on the maximum thrust,  $T_{\max}$ . Since (3) is

a closed-loop model with the controller in [30], the maximum thrust constraint<sup>1</sup> is

$$T = (mge_3 - \mathbf{K}_p(\mathbf{p} - \mathbf{r}) - \mathbf{K}_v\mathbf{v})^\top (\mathbf{R}e_3) \leq T_{\max} \quad (5)$$

which involves the UAV state and setpoint. Next, we formulate the motion planning problem and present the structure of the algorithm that we use for solving it.

#### IV. MOTION PLANNING PROBLEM AND ALGORITHM

We consider a three-dimensional environment  $\mathcal{R} = \mathbb{R}^3$ , where the UAV must move from an initial position to a final position in the environment while avoiding obstacles. Let  $\mathbf{x} \in \mathbb{R}^n = \mathcal{R} \times \mathbb{R}^{n-3}$  be the state of the UAV (3), for simplicity we let the first three elements be the UAV position vector,  $\mathbf{p}$ .

**Definition 6 (Obstacle)** *The environment includes  $M$  known obstacles with occupancy region  $\{\mathcal{S}_\ell \subset \mathcal{R}\}_{\ell=1}^M$ , where each  $\mathcal{S}_\ell$  is either a polyhedral or an ellipsoidal set.*

The restriction to polyhedral or ellipsoidal obstacles stems from computational aspects, to make the problem tractable. The occupancy region in Definition 6 includes all positions at which the UAV may be in collision with an obstacle, i.e., it accounts also for the geometry of the UAV. Now, we can formally state the problem.

**Problem 1** *Compute a setpoint trajectory  $\mathbf{r}(t) = \mathbf{r}_k$  for  $t \in [t_{k-1}, t_k]$ , defined by a finite set of points  $\{\mathbf{r}_k \in \mathcal{R} | k \in \mathbb{Z}_{[1, K]}\}$ , such that the resulting closed-loop trajectory of (3) avoids the obstacles  $\mathbf{p}(t) \notin \bigcup_i \mathcal{S}_i$  for all  $t \geq t_o$ , and  $\mathbf{x}(t)$  converges to an ultimate set associated with  $\mathbf{r}_K \in \mathcal{R}$  in finite time  $t_K - t_o \geq 0$ .*

In this paper, we solve Problem 1 by an invariant-set motion planner. ISMP abstracts the continuous trajectory planning problem for a dynamical system into a search over a graph of discrete points by leveraging invariant sets. Specifically, the algorithm first determines a set of grid points to be nodes of a graph. For each point, it constructs a controller rendering it a stable equilibrium, and an invariant set not intersecting with obstacles such that the closed-loop trajectories starting anywhere in the invariant set converge to the grid point. Then, based on the membership of a grid point to the invariant sets of other grid points, the algorithm constructs the graph edges.

Given initial and final points, ISMP constructs a path in the graph, which is an abstract motion plan. The dynamical system trajectory is obtained from the closed-loop system where the grid point is used as a setpoint within the corresponding invariant set, and the setpoint value changes to the next path grid point when the corresponding invariant set is entered.

The general invariant planning algorithm needs to be modified if the system is subject to uncertainty as in (3). The set construction needs to account for both, model uncertainty and disturbances. The edge construction must account for convergence to a set, rather than a point, see Fig. 1. As for the execution, the batch, i.e., offline, computation of the entire trajectory is no longer possible, since entering the invariant

set of the next waypoint depends also on the effects of the uncertainties, which are not predictable. Rather, the setpoints are updated in real-time, based on the actual UAV state entering the next invariant set.

Next, we develop a robust ISMP (RISMP) for UAV application, summarized in Algorithm 1.

---

#### Algorithm 1 Robust invariant-set motion planning for UAV

**Parameters:**  $\mathbf{P}$  (ellipsoid shape matrix),  $N_v$  (number of nodes),  $\rho_s > 1$  (scaling constant)

**Data:**  $\mathbf{p}(0)$  (initial position),  $\mathbf{p}_\infty$  (target position)

#### Graph Construction:

- 1: Sample grid points  $\{\mathbf{r}_i\}_{i=1}^{N_v-1}$ ,  $\mathbf{r}_i \in \mathcal{R}$ ,  $i \in \mathbb{Z}_{[1, N_v-1]}$
- 2: **for**  $i = 1, \dots, N_v - 1$  **do**
- 3:     Construct  $\mathcal{O}_U^i = \mathcal{E}((\mathbf{r}_i; \mathbf{0}), \mathbf{P}, \rho_U^i)$  for (3)
- 4:     Construct  $\mathcal{O}_I^i = \mathcal{E}((\mathbf{r}_i; \mathbf{0}), \mathbf{P}, \rho_I^i)$  for (3) to be obstacle free,  $\overline{\mathcal{O}_I^i} \cap \mathcal{S}_\ell = \emptyset$  for all  $\ell = \mathbb{Z}_{[1, M]}$ , and constraints admissible, if  $\mathbf{x}(0) \in \mathcal{O}_I^i$ , then  $\mathbf{x}(t) \in \mathcal{X}(\mathbf{r})$ , for all  $t \geq 0$
- 5: **end for**
- 6: Construct  $\mathcal{G} = (\mathbb{V}, \mathbb{E})$  such that  $v_i = \mathbf{r}_i$ ,  $(v_i, v_j) \in \mathbb{E}$  iff  $\rho_s \mathcal{O}_U^i \subseteq \mathcal{O}_I^j$ ,  $\rho_s > 1$

#### Motion Plan:

- 7: Add  $v_{N_v} = \mathbf{r}_{N_v} = \mathbf{p}_\infty$  with  $\mathcal{O}_U^{N_v}$ ,  $\mathcal{O}_I^{N_v}$  to  $\mathcal{G}$  as in Step 6
  - 8: Compute a path  $\mathcal{P} = (v_{\sigma(k)})_{k=1}^K$ , such that  $(\mathbf{p}(0); \mathbf{0}) \in \mathcal{O}_I^{\sigma(1)}$ ,  $v_{\sigma(K)} = v_{N_v}$ ,  $(v_{\sigma(k)}, v_{\sigma(j+1)}) \in \mathbb{E}$ ; set  $k \leftarrow 0$
  - 9: **while**  $k < K$  **do**
  - 10:     **repeat**
  - 11:         Apply  $\mathbf{r} = \mathbf{r}_{\sigma(k)}$  to the closed-loop system (3)
  - 12:         **until**  $\mathbf{x}(t) \in \mathcal{O}_I^{\sigma(k+1)}$
  - 13:          $k \leftarrow k + 1$
  - 14:     **end while**
  - 15: Apply  $\mathbf{r} = \mathbf{r}_{\sigma(N_v)}$  to the closed-loop system (3)
- 

The steps of Algorithm 1 are detailed in the following sections. Specifically, Sec. V describes the computation of the set at Step 3, by considering first a nominal setting similar to [18], and then extending it to the non-ideal attitude tracking setting with polytopic feedback uncertainty. Sec. VI describes the computation of the sets at Step 4. The construction of the graph at Step 6, the computation of the path at Step 8, and the trajectory generation at Steps 11, 15 are described in Sec. VII.

#### V. CONSTRUCTION OF THE ULTIMATE SETS

Next, we describe the computation of the ultimate sets  $\mathcal{O}_U^i$  for each node  $v_i \in \mathcal{G}$  for (3) to design the RISMP algorithm that addresses Problem 1, where for simplicity we omit the index  $i$ . The method presented in [26] can be applied when the gains (4) are: (i) known, i.e.  $N = 1$ , (ii) diagonal, and (iii) there is no variation in the diagonal elements. Thus, [26] does not apply to (3) under Assumption **A2**, **A3**. The LMI-based approach in [18], [37] is more easily generalized, but does not accommodate the structured uncertainty due to the bounded attitude tracking error in Assumption **A1**, which may be necessary to consider in precise UAV planning and control [26], [27].

<sup>1</sup>We do not enforce a lower bound  $T \geq 0$  because for  $T = 0$  the UAV falls at high acceleration due to gravity, and the controller already avoids that.

### A. Perfect Attitude Tracking (Ideal Case)

First, we consider the computation of ellipsoidal ultimate sets for (3a) in the ideal case of  $\tilde{\mathbf{R}} \equiv \mathbf{I}$ . In this case we have a stable linear system with polytopic uncertainty in the dynamics by **A2** and bounded input disturbances by **A1**, with the tracking errors  $\mathbf{x}_e = (\mathbf{p} - \mathbf{r}; \mathbf{v})$  evolving by

$$\dot{\mathbf{x}}_e = \mathbf{A}\mathbf{x}_e + \mathbf{B}\Delta, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_p & -\mathbf{K}_v \end{bmatrix}, \quad \mathbf{K} \in \mathcal{K}. \quad (6)$$

**Remark 1** For system (6), when  $\mathbf{c} = (\mathbf{r}; \mathbf{0})$  the sets in Definition 4 are translation invariant with respect to  $\mathbf{r} \in \mathcal{R}$ , i.e., if  $\mathcal{O}_U = \mathcal{E}((\mathbf{r}; \mathbf{0}), \mathbf{P}, \rho_U)$  is an ultimate set, then, for any  $\mathbf{r}' \in \mathcal{R}$ ,  $\mathcal{O}_U = \mathcal{E}((\mathbf{r}'; \mathbf{0}), \mathbf{P}, \rho_U)$  is also an ultimate set.

In the following, let  $V = \|\mathbf{x}_e\|_{\mathbf{P}}^2$  be a quadratic Lyapunov function candidate, with  $\mathbf{P}$  partitioned as

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

There are multiple ways to compute RPI sets for (6), see, e.g., [37]. We formulate the convex optimization problem

$$\gamma^* = \min_{\substack{\mathbf{P} \in \mathbb{S}_{++}^{6 \times 6} \\ \gamma > 0}} \gamma \quad (7a)$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{A}_h^\top \mathbf{P} + \mathbf{P} \mathbf{A}_h + \mathbf{P} & \mathbf{P} \mathbf{B} \\ \mathbf{B}^\top \mathbf{P} & -\gamma \mathbf{I} \end{bmatrix} \preceq \mathbf{0}, \quad h \in \mathbb{Z}_{[1, N]}, \quad (7b)$$

$$\mathbf{P} \succeq \mathbf{I}, \quad (7c)$$

where

$$\mathbf{A}_h = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_p^h & -\mathbf{K}_v^h \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}.$$

Solving (7) implies the existence of a common Lyapunov function  $V(\mathbf{p}, \mathbf{v}, \mathbf{r}) = \|\mathbf{x}_e\|_{\mathbf{P}}^2$  for all closed-loop systems characterized by the gains  $\mathbf{K} \in \mathcal{K}$ . By a straightforward application of the comparison lemma [36, Lemma 3.4], the level set  $\|\mathbf{x}_e\|_{\mathbf{P}}^2 \leq \gamma^* \Delta_{\max}^2$  is an ultimate bound for (6) for the given setpoint  $\mathbf{r}$ . Thus,  $\mathcal{E}((\mathbf{r}; \mathbf{0}), \mathbf{P}, \rho_U)$  is an ultimate set with  $\rho_U = \gamma^* \Delta_{\max}^2$  for all  $\|\Delta(t)\| \leq \Delta_{\max}^2$ .

### B. Discussion on Conservativeness

Computing ultimate sets by (7) is conservative, thus we compare it to the method in [26, Sec. VI-A]. Consider a simple case with known scalar gains  $k_p, k_v$ ,

$$\frac{d}{dt} \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_v \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta, \quad p(t_o) = v(t_o) = 0, \quad (8)$$

where  $\sup_t |\Delta(t)| \leq 1$ . We compute  $\delta = \sup_t |p(t)|$  using:

- the system 1-norm, from the impulse response  $H$  from input  $\Delta$  to output  $y = p$ ,  $\delta_A = \|H\|_1$ ;
- the ellipsoidal set computed as in [26],  $\delta_B$ ;
- the ellipsoidal set computed by (7), where

$$\mathbf{P} = \begin{bmatrix} P_{pp} & P_{pv} \\ \star & P_{vv} \end{bmatrix} \in \mathbb{S}_{++}^2,$$

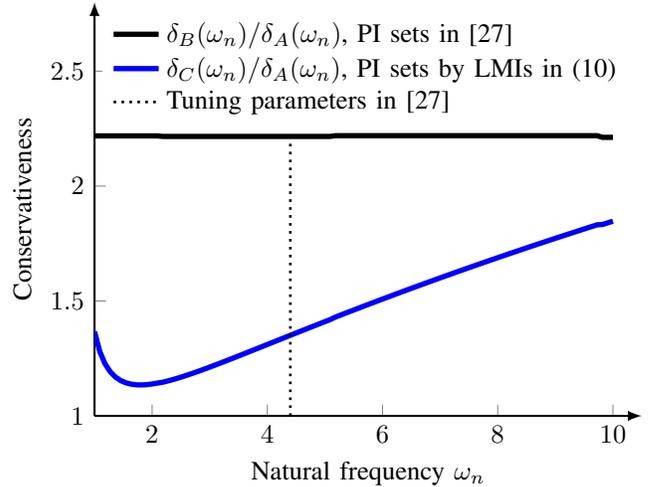


Fig. 3. Conservativeness of the PI sets against the system 1-norm for (8) with  $k_k = \omega_n^2$  and  $k_v = 2\xi\omega_n$  for variable  $\omega_n$  and fixed  $\xi = \sqrt{1/2}$ .

and the resulting margin using (1) as

$$\delta_C = \sqrt{\frac{\rho_U}{P_{pp} - P_{pv} P_{vv}^{-1} P_{vp}}}.$$

for the same parameters as in [26],  $k_p = 19.34, k_d = 6.22$ , making a closed-loop system with a natural frequency  $\omega_n = 4.4$  and damping ratio  $\xi = \sqrt{1/2}$ , the results are  $\delta_A = 0.056, \delta_B = 0.125, \delta_C = 0.076$ . The system 1-norm  $\delta_A$  is a lower bound on the maximal positional perturbation due to a bounded input. The margin  $\delta_C$  computed by (7) are closer to  $\delta_A$  than the margin  $\delta_B$  from [26]. Studying the conservativeness over a wider range of tuning parameters, see Fig. 3 for the results for varying  $\omega_n \in [1, 10]$  while keeping the relative damping at  $\xi = \sqrt{1/2}$ , such relation appears to hold more generally. Next, we generalize the method to incorporate the bounded attitude tracking errors in the Assumption **A1**.

### C. Imperfect Attitude Tracking (Non-ideal Case)

For the more realistic setting where the attitude tracking error is non-zero but bounded, as in Assumption **A1**, we start by giving two useful lemmas.

**Lemma 1** Any  $\tilde{\mathbf{R}} = \exp(\mathbf{S}(\alpha\boldsymbol{\nu})) \in \text{SO}(3)$  corresponding to a rotation of  $\alpha \in [0, \pi/2)$  about a unit vector  $\boldsymbol{\nu} \in \mathbb{R}^3$ , and any matrices  $\Phi, \Psi \in \mathbb{R}^{3 \times m}$ , satisfy the bound

$$\begin{aligned} \mathbf{C} &= \Phi^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top) \Psi + (\Phi^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top) \Psi)^\top \\ &\preceq (\Phi^\top \Phi + \Psi^\top \Psi) \sqrt{2(1 - \cos(\alpha))}. \end{aligned}$$

The full proof is reported in Appendix A, and follows by writing  $\mathbf{C}$  as a quadratic form in  $(\Phi, \Psi)$ , considering the characteristic equation of the resulting block-structured matrix using Schur complements, and geometric reasoning about the singular values of  $(\mathbf{I} - \tilde{\mathbf{R}}^\top)$ .

**Lemma 2** For any  $\mathbf{K} \in \text{Co}(\{\mathbf{K}_i\}_{i=1}^N)$  and  $\bar{\mathbf{K}}$  such that

$$\begin{bmatrix} \bar{\mathbf{K}} & \mathbf{K}_h^\top \\ \mathbf{K}_h & \mathbf{I} \end{bmatrix} \succeq \mathbf{0} \quad \forall h \in \mathbb{Z}_{[1, N]}, \quad \text{it holds } \bar{\mathbf{K}} \succeq \mathbf{K}^\top \mathbf{K}.$$

The proof is given in Appendix B.

With non-ideal attitude tracking, instead of (7) we solve

$$\gamma^* = \min_{\substack{P \in \mathbb{S}_{++}^{6 \times 6} \\ K \in \mathbb{S}_{++}^{6 \times 6} \\ \gamma \geq 0}} (\gamma), \quad (9a)$$

$$\text{s.t. } P \succeq I, \quad (9b)$$

$$\begin{bmatrix} \bar{K} & \star \\ K_h & I \end{bmatrix} \succeq 0 \quad (9c)$$

$$\begin{bmatrix} A_h^\top P + P A_h + P + \beta \bar{K} & \star & \star \\ B^\top P & -\gamma I & \star \\ \sqrt{\beta} B^\top P & 0 & -I \end{bmatrix} \preceq 0 \quad (9d)$$

$h \in \mathbb{Z}_{[1, N]}$ ,

where

$$A_h = \begin{bmatrix} 0 & I \\ -K_p^h & -K_v^h \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

$$K_h = \begin{bmatrix} K_p^h & K_v^h \end{bmatrix}, \quad \beta = \sqrt{2(1 - \cos(\alpha_{\max}))},$$

The characterization of an ellipsoidal RPI set based on the solution of (9) is summarized in the following proposition.

**Proposition 1** *If Assumptions A1–A3 hold,  $\alpha_{\max}$  is the maximal attitude error; and there exists a solution to (9), there exists a constant  $\rho_U = \Delta_{\max}^2 \gamma^* > 0$  such that  $\mathcal{O}_U = \mathcal{E}(c, P, \rho_U)$  is RPI for any  $c = (r; \mathbf{0})$ .*

The proof is in Appendix C, and follows the perfect attitude case, using Lemma 1 and Lemma 2 to bound the term due to the errors in attitude control. Each node  $v_i \in \mathcal{G}$  has an associated ultimate set  $\mathcal{O}_U^i$ . However, since  $\mathcal{O}_U^i$  depends only on the closed-loop dynamics, uncertainty, and chosen Lyapunov function, and since in this paper these are equal for all the nodes,  $\mathcal{O}_U^i = \mathcal{O}_U$ ,  $\rho_U^i = \rho_U$  for all  $i \in \mathbb{Z}_{[1, N_v]}$ . Hence, in the rest of this paper we drop the superscript. Proposition 1 allows for applying RISMP under realistic assumptions and with quantities, e.g.,  $\alpha_{\max}$ ,  $F_{\max}$ ,  $\mathcal{K}$ , that can be estimated from data for any UAV with a control system that stabilizes its position about a setpoint.

## VI. CONSTRUCTION OF THE INFLATED SETS

Next, we need to construct forward invariant inflated sets  $\mathcal{O}_I^i$  associated to nodes  $v_i \in \mathcal{G}$  according to Definition 4 such that the trajectories of (3) within them are safe, i.e., satisfy the input constraints and do not enter the obstacles sets  $\{\mathcal{S}_\ell\}_{\ell=1}^M$ . Since a transition  $(v_i, v_j) \in \mathbb{E}$  can be guaranteed to be safely taken only if the set  $\mathcal{O}_U^i$  associated with  $v_i$  is contained in the set  $\mathcal{O}_I^j$  associated to the vertex  $v_j$ , we want  $\mathcal{O}_I^j$  to be as large as possible. In what follows, we omit the indices to  $\mathcal{O}_I$  for the ease of notation, and use the following lemma.

**Lemma 3** ([38, Proposition 1]) *Let  $x_e = (p - r; v)$ ,  $V(p, v, r) = \|x_e\|_P^2$  with  $P$  computed in (9) and*

$$\Gamma(r) = \frac{(c_p^\top r - d(r))^2}{\begin{bmatrix} c_p \\ c_d \end{bmatrix}^\top P^{-1} \begin{bmatrix} c_p \\ c_d \end{bmatrix}}. \quad (10)$$

*If  $V(p, v, r) \leq \Gamma(r)$  then  $c_p^\top p + c_v^\top v \leq d(r)$ .*

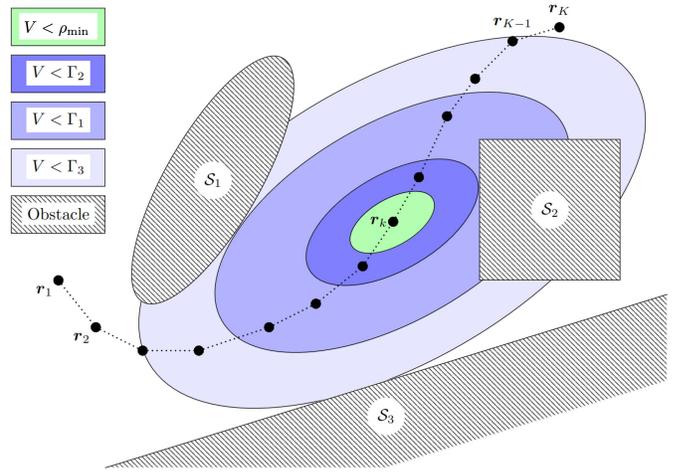


Fig. 4. Setpoints  $\{r_k \in \mathcal{R}\}_{k=1}^K$  in a world with  $M = 3$  three obstacles  $\{\mathcal{S}_i \subset \mathcal{R}\}_{i=1}^M$  (gray). The ultimate set  $\mathcal{O}_U = \mathcal{E}(c, P, \rho_U)$  associated with the equilibrium point  $c = (r_k; \mathbf{0})$  is shown in green. Candidate inflated sets  $\{\mathcal{E}(c, P, \Gamma_i)\}_{i=1}^M$  (blue) are computed, and the safe inflated set is  $\mathcal{O}_I = \mathcal{E}(c, P, \rho_I)$  with  $\rho_I = \Gamma_2$ . These sets are used to compute the graph in the RISMP method, as shown in Fig. 1 and discussed in Sec. VII.

Lemma 3 determines how much the ellipsoidal ultimate set  $\mathcal{O}_U$  associated to a vertex of the graph can be enlarged to construct  $\mathcal{O}_I$  where obstacle avoidance and actuator limitations are satisfied. Given  $M_c$  input constraints,  $M$  obstacles and  $\mathcal{O}_U = \mathcal{E}(c, P, \rho_U)$ , we compute  $M + M_c$  candidate inflated sets  $\{\mathcal{E}(c, P, \Gamma_\ell)\}_{\ell=1}^{M+M_c}$  using Lemma 3, and select

$$\rho_I = \min(\{\Gamma_\ell\}_{\ell=0}^{M+M_c}), \quad (11)$$

resulting in  $\mathcal{O}_I = \mathcal{E}(c, P, \rho_I)$ . The computation of  $\Gamma_\ell$  for polyhedral obstacles, ellipsoidal obstacles, and thrust constraints is discussed in Sec. VI-A, Sec. VI-B, and Sec. VI-C, respectively.

### A. Polyhedral Positional Constraints

Consider a setpoint  $r \in \mathcal{R}$  and let  $Q = P_{pp} - P_{pv} P_{vv}^{-1} P_{vp}$  be the inverse shape matrix of  $\mathcal{O}_I$  in Definition 5. Introducing a coordinate transformation  $\bar{p} = Q^{1/2}(p - r)$ , if  $V(p, v, r) \leq \Gamma$  is RPI, then  $\|\bar{p}\|_2^2 \leq \Gamma$  is also RPI. We apply the linear transformation to the obstacles  $\mathcal{S}_\ell = \{p : A_\ell^S p \leq b_\ell^S\}$ , resulting in  $\bar{\mathcal{S}}_\ell = \{\bar{p} : \bar{A}_\ell^S \bar{p} \leq \bar{b}_\ell^S\}$  with  $\bar{A}_\ell^S = A_\ell^S Q^{-1/2}$ ,  $\bar{b}_\ell^S = b_\ell^S - A_\ell^S r$ . In the transformed coordinates, we solve

$$\bar{p}_\ell^* = \underset{\bar{p} \in \bar{\mathcal{S}}_\ell}{\operatorname{argmin}} \|\bar{p}\|_2^2. \quad (12)$$

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

$$c_{p,\ell}^\top \bar{p} = (\bar{p}_\ell^*)^\top \bar{p} \leq (p_\ell^*)^\top p_\ell^*, \quad (13)$$

and hence for obstacle  $\mathcal{S}_\ell$ ,  $\Gamma_\ell = \|\bar{p}_\ell^*\|_2^2$ . See  $\Gamma_1$  and  $\Gamma_3$  in Fig. 4 for an example of two ellipsoidal obstacles.

### B. Ellipsoidal Positional Constraints

As in Sec. VI-A, let  $Q = P_{pp} - P_{pv} P_{vv}^{-1} P_{vp}$ , and now consider ellipsoidal obstacle sets,  $\mathcal{S}_\ell = \{p \in \mathbb{R}^3 : \|p - b_\ell^S\|_{A_\ell^S} \leq 1\}$ . In this case, the computation of candidate inflated sets (12) is a quadratically constrained QP (QCQP).

Using the change of coordinates from Sec. VI-A,  $\bar{\mathcal{S}}_\ell = \{\bar{\mathbf{p}} \in \mathbb{R}^3 : \|\bar{\mathbf{p}} - \bar{\mathbf{b}}_\ell^S\|_{\bar{\mathbf{A}}_\ell^S} \leq 1\}$ , with  $\bar{\mathbf{A}}_\ell^S = \mathbf{Q}^{-1/2} \mathbf{A}_\ell^S \mathbf{Q}^{-1/2}$ ,  $\bar{\mathbf{b}}_\ell^S = -\mathbf{Q}^{-1/2}(\mathbf{r} - \mathbf{b}_\ell^S)$ . By solving (12), we can construct the linear constraint as in (13), and apply Lemma 3 to yield  $\Gamma_\ell = \|\bar{\mathbf{p}}_\ell^*\|_2^2$ . This is a more computationally demanding but less conservative than the approach in [25], that uses a hyperplane as a constraint in Lemma 3.

**Remark 2** *The tightness of the approximation in [25] depends on the spectrum of  $\bar{\mathbf{A}}_\ell^S$ . If the inflated sets and obstacles are spherical, the difference between the method in [25] and the solution of the QCQP in (12) is negligible. Thus, under certain conditions of obstacle shapes and computational limitations, the method in [25] can be used in the RISMP algorithm proposed here.*

**Remark 3** *If the obstacle sets do not account for the physical shape of the UAV, one can find a covering ellipsoidal set of the UAV shape  $\mathcal{E}(\mathbf{p}, \mathbf{Q}, \rho_{\text{bnd}})$  for some  $\rho_{\text{bnd}} > 0$ , where  $\mathbf{Q}$  is the projection of  $P$  in positions coordinates, and then set  $\Gamma_\ell = (\|\bar{\mathbf{p}}_\ell^*\|_2 - \sqrt{\rho_{\text{bnd}}})^2$ .*

### C. Thrust Constraints

Bounds on the thrust of the UAV are more cumbersome to enforce in the context of Assumption **A2**, and depend on the UAVs low-level controllers (see Fig. 2). For the trajectory tracking controller considered here [30] the commanded thrust is  $T = (mg\mathbf{e}_3 - \mathbf{K}_p(\mathbf{p} - \mathbf{r}) - \mathbf{K}_d\mathbf{v})^\top (\mathbf{R}\mathbf{e}_3)$ , and for all the states in  $\mathcal{O}_T$  it must hold that  $T \leq T_{\text{max}}$ . Standard approaches such as [26] are difficult to apply if the gains  $\mathbf{K} \in \mathcal{K}$  are dense. In this case, we bound the denominator in (10) as in the following lemma

**Lemma 4** *Given Assumption **A2**, let  $\bar{\mathbf{K}} = \text{diag}(\mathbf{K}_p, \mathbf{K}_v)$  and  $\bar{\mathbf{K}}_h = \text{diag}(\mathbf{K}_p^h, \mathbf{K}_v^h)$ . There exists a  $\gamma > 0$  such that*

$$\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\nu} \end{bmatrix}^\top \bar{\mathbf{K}} \mathbf{P}^{-1} \bar{\mathbf{K}} \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\nu} \end{bmatrix} \leq \gamma \quad \forall \boldsymbol{\nu} \in \mathbb{R}^3 \text{ s.t. } \|\boldsymbol{\nu}\|_2 = 1. \quad (14)$$

The smallest such  $\gamma$  is

$$\gamma^* = \min_{\gamma_{11}, \gamma_{12}, \gamma_{22}} (\gamma_{11} + 2\gamma_{12} + \gamma_{22}), \quad (15a)$$

$$\text{s.t. } \begin{bmatrix} \mathbf{P} & \bar{\mathbf{K}}_h \\ \bar{\mathbf{K}}_h & \boldsymbol{\Lambda} \end{bmatrix} \succeq \mathbf{0}, \quad \forall h \in \mathbb{Z}_{[1, N]} \quad (15b)$$

$$\begin{bmatrix} \gamma_{11} \mathbf{I} & \gamma_{12} \mathbf{I} \\ * & \gamma_{22} \mathbf{I} \end{bmatrix} = \boldsymbol{\Lambda}. \quad (15c)$$

The proof is straightforward and follows by the definition of convexity and the application of Schur complements. Existence of a bound  $\gamma$  follows as  $\bar{\lambda}(\bar{\mathbf{K}} \mathbf{P}^{-1} \bar{\mathbf{K}})$  is bounded for any  $\mathbf{P} \succ \mathbf{0}$ , which is ensured by the constraint in (9b).

Using Lemma 3,  $\Gamma = (f_{\text{max}} - mg)^2 m^{-2} (\gamma^*)^{-1}$  ensures that for the control law in [30],  $T < T_{\text{max}}$  for all states in the ellipsoid  $\mathcal{O}_T = \mathcal{E}(\mathbf{c}, \mathbf{P}, \Gamma)$ . As we have one such constraint, we let  $M_c = 1$  and associate  $\Gamma_0$  with this thrust constraint.

## VII. RISMP PLANNING AND EXECUTION

Using the ultimate sets from Sec. V and inflated safe sets from Sec. VI, we proceed to construct a graph  $\mathcal{G}$  with a vertex

set  $\mathbb{V}$ . Let  $\mathbf{r}_i \in \mathcal{R}$  be a position setpoint,  $\mathcal{O}_U^i = \mathcal{E}(\mathbf{r}^i, \mathbf{P}, \rho_U)$  the associated ultimate set, and  $\mathcal{O}_T^i = \mathcal{E}(\mathbf{r}^i, \mathbf{P}, \rho_T^i)$  the associated inflated safe set. Thus, each vertex in  $\mathcal{G}$  is associated with a triple  $(\mathbf{r}_i, \mathcal{O}_U^i, \mathcal{O}_T^i)$ , and each edge  $(v_i, v_j) \in \mathbb{E} \subset \mathbb{V} \times \mathbb{V}$  is associated with a weight  $c_{ij} \geq 0$ . This weight is stored in a sparse matrix  $\mathbf{C}$ , such that  $[\mathbf{C}]_{ij} = c_{ij}$  if  $(v_i, v_j) \in \mathbb{E}$ , and  $[\mathbf{C}]_{ij} = 0$  otherwise.

Next, we describe how to construct the graph edges in Sec. VII-A. We show how to compute a path by searching the graph in Sec. VII-B, and how to generate the UAV motion that solves Problem 1 in Sec. VII-C.

### A. Graph Building

As shown in Algorithm 1, the first step of RISMP is to determine suitable reference positions, which become the graph nodes, this is done by a lattice construction, random sampling, or optimization methods [18]. Then, we compute the ultimate sets and inflated sets. A node  $v_i$  is pruned from the graph if  $\mathcal{O}_U^i \cap \bigcup_{\ell=1}^M \mathcal{S}_\ell \neq \emptyset$ , as its equilibrium is not safe.

The edges of the directed graph are determined from the sets associated with each vertex. Specifically, if  $\mathcal{O}_U^i \subset \mathcal{O}_T^j$ , then any trajectory starting in  $\mathcal{O}_U^i$  eventually converges to  $\mathcal{O}_U^j$ , and from this it will be safe to change move towards  $\mathbf{r}_j$ . Thus,

$$(v_i, v_j) \in \mathbb{E} \text{ if } \rho_s \mathcal{O}_U^i \subseteq \mathcal{O}_T^j. \quad (16)$$

where  $\rho_s = 1 + \varepsilon_s$  is a scaling constant with  $\varepsilon_s > 0$ , arbitrarily small, which ensures that the ultimate set  $\mathcal{O}_U^i$  is strictly in the interior of  $\mathcal{O}_T^j$  with margin  $\varepsilon_s$ , and provides an upper bound on the time to enable the transitions. A larger  $\varepsilon_s$  reduces the number of edges in the graph and hence the number of feasible paths, but provides a smaller upper bound on the time to enable the transitions. In the coordinates  $\bar{\mathbf{r}}_i = (\mathbf{P}_{pp} - \mathbf{P}_{pv} \mathbf{P}_{vv}^{-1} \mathbf{P}_{vp})^{1/2} \mathbf{r}_i$ , condition (16) is efficiently checked across  $\mathbb{V} \times \mathbb{V}$  as

$$\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\|_2 < \sqrt{\rho_T^j} - \sqrt{\rho_s \rho_U}. \quad (17)$$

Vertices that are disconnected or that have only inbound edges may be pruned from the graph to reduce computations.

**Remark 4** *By geometry, two vertices for which  $\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\|_2 > \sqrt{\Gamma_\ell} - \sqrt{\rho_s \rho_U}$  cannot be connected. This imposes an upper bound on how coarsely the vertices of the graph are sampled, where, for instance,  $\ell$  is the index of the inflated set limited by the thrust constraint (5), and can be used to optimize the graph construction by only checking (16) for vertices that are sufficiently close.*

**Remark 5** *The edge weights can be defined in several ways. For simplicity, we set the weight associated with  $(v_i, v_j) \in \mathbb{E}$  to  $c_{ij} = \|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\|_2$ . Different definitions are possible, which may cause an increase of computational load. A more extensive study of alternative computationally efficient weight definitions is left to future work.*

### B. Graph Searching

A path from an initial vertex to a terminal vertex can be determined efficiently using the Dijkstra algorithm [39],

although any graph search method can be used within RISMP. Before initializing the graph search, we choose the initial and terminal vertices. The initial vertex is chosen as

$$\sigma(1) = \arg \min_{i: v_i \in \mathbb{V}} \|\mathbf{x}(t_o) - (\mathbf{r}_i; \mathbf{0})\|_{\mathcal{P}}^2, \quad (18a)$$

$$\text{s.t., } \mathbf{x}(t_o) \in \mathcal{O}_{\mathcal{I}} \quad (18b)$$

such that  $\|\mathbf{x}(t_o) - (\mathbf{r}_i; \mathbf{0})\|_{\mathcal{P}}^2 \leq \rho_{\mathcal{I}}^i$ . For a target position  $\mathbf{p}_{\infty} \in \mathcal{R}$ , we add to the graph vertex  $v_{N_v}$  with  $\mathbf{r}_{N_v} = \mathbf{p}_{\infty}$ . For this new vertex, an inflated safe set is determined as in Sec. VI, and the edges are added to the neighboring vertices as in Sec. VII-A. The path  $\mathcal{P} = (v_{\sigma(k)})_{k=1}^K$  resulting from the graph search is a sequence of  $K \geq 1$  connected vertices,  $(v_{\sigma(k)}, v_{\sigma(k+1)}) \in \mathbb{E}$  for all  $k \in \mathbb{Z}_{[1, K-1]}$  leading from the initial node to the terminal node,  $v_{\sigma(K)} = v_{N_v}$ .

### C. RISMP Execution

Online, the RISMP algorithm determines the setpoint for the UAV in closed-loop with the controller (3) using the inflated sets  $\mathcal{O}_{\mathcal{I}}^i$  associated to the nodes  $v_i$ . Initially, we set  $t_1 = t_o$ , and  $\mathbf{r}(t_1) = \mathbf{r}_{\sigma(1)}$ . Given  $t > t_{k-1}$ , if  $\mathbf{x}(t) \in \mathcal{O}_{\mathcal{I}}^{\sigma(k)}$ , we let  $t_k = t$  and  $\mathbf{r}(t) = \mathbf{r}_{\sigma(k)}$ , otherwise  $\mathbf{r}(t) = \mathbf{r}_{\sigma(k-1)}$ .

**Remark 6** We consider motion planning in a known map with known obstacles. However, RISMP can be used in partially known maps with moving or previously unknown obstacles by adding and removing graph nodes during execution based on online information on the map and obstacles. These have been explored for an invariant-based planner in [22], and can be easily integrated with the RISMP method described here.

The construction (16) ensures that  $t_k - t_{k-1}$  is finite and that the RISMP solves Problem 1. Specifically, we give the following result.

**Result 1 (Finite-time Guarantees and Safety)** Given the quadrotor model (3) and Assumptions A1–A3, if:

- (i) there exist a triple  $(\gamma, \bar{\mathbf{K}}, \mathbf{P})$  solving (9);
- (ii)  $\mathcal{G}$  is computed as in Sec. VII-A and is fully connected;
- (iii) the initial condition satisfies  $\mathbf{x}(t_o) \in \bigcup_{i=1}^{N_v} \mathcal{O}_{\mathcal{I}}^i$ ;
- (iv) the setpoint  $\mathbf{r}_k$  is updated as in Sec. VII-C;

then for every  $v_{\sigma(K)} \in \mathcal{G}$  there exists  $\delta > 0$  and a time

$$t_K \leq \sum_{k=1}^K \log(\delta^{-1}(\rho_{\mathcal{I}}^{\sigma(k)} - \rho_s \rho_U)) \triangleq t_{\max}, \quad (19)$$

such that  $\mathbf{x}(t) \in \mathcal{O}_{\mathcal{U}}^{\sigma(K)}$  for all  $t > t_K$ . In addition, the state trajectory  $(\mathbf{x}(t))_{t \geq t_o}$  satisfies:

- (P1) finite-time convergence:  $\mathbf{x}(t) \in \mathcal{O}_{\mathcal{U}}^{\sigma(K)}$  for all  $t > t_{\max}$ ;
- (P2) collision avoidance:  $\mathbf{p}(t) \notin (\bigcup_{\ell=1}^M \mathcal{S}_{\ell})$  for all  $t \geq t_o$ ;
- (P3) constraint satisfaction:  $T(t) \leq T_{\max}$  for all  $t \geq t_o$ .

The constant  $\delta$  comes from the closed-loop motion model being UAS, from the strict inequality in (17) and from  $\mathcal{O}_{\mathcal{U}}^i$  being strictly in the interior of  $\mathcal{O}_{\mathcal{I}}^j$ , and the existence of a finite  $K$  follows from (ii). The bound on  $t_K$  and (P1) follows directly from repeated application of the comparison lemma [36, Lemma 3.4] given the setpoint update in (iv). Since the sets  $\mathcal{O}_{\mathcal{I}}^i$  are forward invariant as per (i), they yield

collision avoidance (P2) as  $\mathbf{p}(t) \in \bigcup_i \overline{\mathcal{O}_{\mathcal{I}}^i}$  for  $t \geq t_o$ , and  $\overline{\mathcal{O}_{\mathcal{I}}^i} \cap (\bigcup_{\ell=1}^M \mathcal{S}_{\ell}) = \emptyset$  by construction. Similarly, constraint satisfaction (P3) follows the state trajectory  $\mathbf{x}(t) \in \bigcup_i \mathcal{O}_{\mathcal{I}}^i$  for all  $t \geq t_o$  and within  $\mathcal{O}_{\mathcal{I}}^i$  the constraints, here (5), are satisfied by construction.

The graph  $\mathcal{G}$  may not be fully connected even if the physical space is, e.g., if there is a passage too narrow to be navigated safely when accounting for the uncertainty. In this case the method may still be successfully applied within each connected subgraph.

**Remark 7** In Result 1, the success or failure in computing a feasible state trajectory between any initial condition  $\mathbf{x}(t_o) \in \bigcup_i \mathcal{O}_{\mathcal{I}}^i$  and target vertex in the graph depends entirely on the connectivity of  $\mathcal{G}$ , which is known offline. The connectivity requirement of  $\mathcal{G}$  can be removed, and substituted with the existence of a path in the graph  $\mathcal{G}$  for the specific target, i.e., on the graph search in Section VII-B returning a feasible path. However, in such case the success or failure of the algorithm is only known after executing the graph search, and multiple instance of search must be executed with different initial nodes, specifically for all  $v_{\sigma(1)} = v_i$ , such that  $\mathbf{x}(t_o) \in \mathcal{O}_{\mathcal{I}}^i$ .

## VIII. VALIDATION IN SIMULATIONS

First, we validate the RISMP method in simulations, where we can inject specific disturbances and uncertainty and run Monte-Carlo analysis of the closed-loop behavior. We consider a scenario where the space is cluttered by polyhedral obstacles in the shape of buildings, see Fig. 8.b, which will be also used for experiments. In the scenario, the RISMP receives the current UAV location and a target grid point, computes the trajectory from the current location to the target, and controls the UAV to reach the ultimate set associated to the target.

### A. Model Identification and Offline Processing

In order to relate the simulation results to the subsequent experimental results, we identify the parameters in (3) from quadrotor data obtained by flying a Crazyfly 2.1 UAV with the controller in [30] to follow prescribed setpoints. By gray-box identification [40] we find that the motion model is well approximated by second-order system (3) with gains

$$\mathbf{K}_p^* = \text{diag}(7.78, 7.38, 11.30), \quad (20a)$$

$$\mathbf{K}_v^* = \text{diag}(3.28, 3.27, 3.75). \quad (20b)$$

The sequence of setpoints, real response, and simulated response from (3) with the gains (20) are shown in Fig. 5. To assess robustness with respect to imperfectly known gains, we construct  $\mathcal{K}$  by sampling  $N = 3$  gains around the nominal one found by identification (20)

$$(\mathbf{K}_p^i, \mathbf{K}_v^i) = (\mathbf{K}_p^*, \mathbf{K}_v^*) \text{diag}(\boldsymbol{\eta}), \quad \boldsymbol{\eta} \sim \mathcal{U}([0.9, 1.1]^6). \quad (21)$$

For the subsequent simulation results the gains are

$$\mathbf{K}_p^1 = \text{diag}(7.77, 7.38, 11.30), \quad \mathbf{K}_v^1 = \text{diag}(3.28, 3.27, 3.75),$$

$$\mathbf{K}_p^2 = \text{diag}(7.66, 7.45, 10.79), \quad \mathbf{K}_v^2 = \text{diag}(3.14, 3.12, 3.71),$$

$$\mathbf{K}_p^3 = \text{diag}(7.90, 7.16, 11.73), \quad \mathbf{K}_v^3 = \text{diag}(3.26, 3.31, 3.67).$$

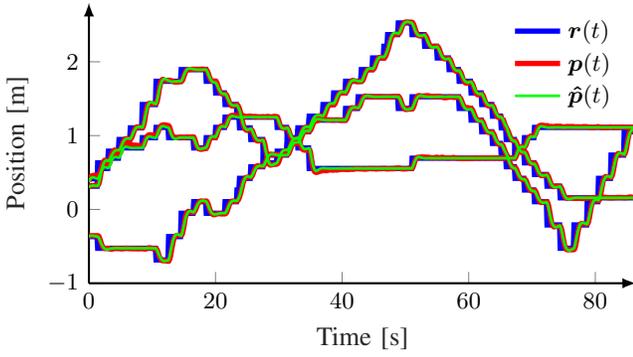


Fig. 5. Reference trajectory  $r(t)$  (blue), real system response  $p(t)$  (red), and simulated system response with the identified gains  $\hat{p}(t)$  (green).

From first principles, UAV specifications, and data we obtain the parameters  $g = 9.81\text{m/s}^2$ ,  $m = 0.03\text{kg}$ ,  $\alpha_{\max} = 0.1\text{rad}$ ,  $T_{\max} = 2mg = 0.2943\text{N}$  (see Sec. III), and  $F_{\max} = 0.02\text{N}$ . The synthesis of a common Lyapunov function by (9) yields  $\rho_U = 0.233$  with

$$P = \begin{bmatrix} 6.052 & 0 & 0 & 0.956 & 0 & 0 \\ 0 & 5.798 & 0 & 0 & 0.935 & 0 \\ 0 & 0 & 9.798 & 0 & 0 & 1.343 \\ 0.956 & 0 & 0 & 1.202 & 0 & 0 \\ 0 & 0.935 & 0 & 0 & 1.182 & 0 \\ 0 & 0 & 1.343 & 0 & 0 & 1.301 \end{bmatrix}, \quad (22)$$

resulting in the worst-case position margins of 0.21m, 0.21m, 0.17m for  $x$ ,  $y$ ,  $z$ , respectively, when projecting  $\mathcal{O}_U$  on each dimension. We compute  $\mathcal{O}_U$  by solving the LMIs (9) using CVX with SDPT3 and default numerical tolerances [41], [42]. The graph is then constructed as in Sec. VII-A starting from an initial set of 4000 vertices ( $20 \times 20 \times 10$ ), equidistantly sampled in a  $3\text{m} \times 3\text{m} \times 1\text{m}$  rectangular region. The graph construction takes approximately 24.38s in a Lenovo Carbon X1 laptop with 2.8GHz Intel i7-1165G7 4-cores CPU and 32MB RAM for a non-optimized Matlab implementation using `quadprog` to solve the QPs in (12) for polyhedral obstacles. After pruning infeasible, disconnected, and sink vertices, a total of 1939 vertices remain with an average of 52.72 directed edges each, resulting in a memory occupancy on the order of 1MB. This can be significantly reduced, as hinted in Remark 4.

### B. Online Processing and Simulation Setup

In simulations, we consider two different scenarios, referred to as Scenario A and B, respectively, that differ only in size of the obstacles. In scenario A, the obstacles are 0.8m tall, which allows for flights above them, whereas in Scenario B the obstacles are 1.5m tall, that forces the quadrotor to navigate between them due to the maximum altitude in the graph being 1m. When solving the graph-search problem, this results in different paths as shown in Fig. 6.

We run a total of 100 simulations for each scenario, where we vary: (i) the controller gains  $K \sim \mathcal{U}(K)$ ; (ii) the initial conditions in the boundary of the initial inflated set,  $x(t_0) \sim \mathcal{U}(\text{bnd}(\mathcal{O}_U^1))$ ; (iii) the input disturbance realization  $\Delta(t)$ , which is constructed as follows. First, we randomly sample a unitary vector  $\nu \in \mathbb{R}^3$  and apply the maximal rotation around

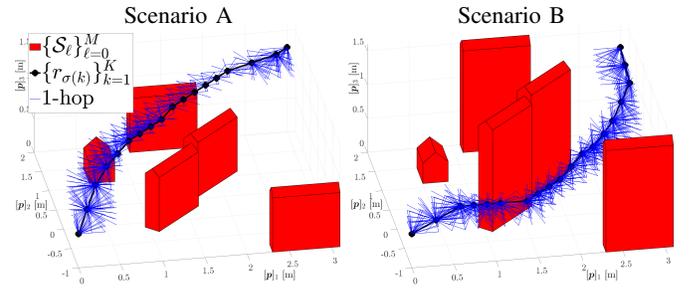


Fig. 6. Sequence of setpoints  $\{r_k\}_{k=1}^K$  and associated 1-hop neighborhood of the vertices  $\{v_k\}_{k=1}^K$  visualized for scenario A and B, respectively.

it, obtaining the rotation matrix  $\tilde{R} = e^{S(\alpha_{\max}\nu)}$ . Then, the disturbance signal  $\Delta(t)$  is applied in a direction such that  $f$  is aligned with  $(I - \tilde{R})e_3$ , see (3), as a sinusoid with the amplitude obtained from  $F_{\max}$ , and the frequency that maximizes the  $\mathcal{H}_\infty$  gain from disturbance to position error for the closed-loop UAV model, i.e., the worst case frequency. For the parameters used in this paper, the worst case disturbance is low frequency, so that it may be applied as a constant at maximum amplitude.

In the simulations and subsequent experiments, the graph is searched using the Boost Graph Library [43] with a standard Dijkstra algorithm [39]. In this setting, it takes less than 2ms to find an optimal solution in both scenarios, with  $K = 18$  in scenario A and  $K = 19$  in scenario B. The resulting state trajectories from 100 simulations are shown in Fig. 7.

### C. Discussion

The quadrotor reaches the target invariant set  $\mathcal{O}_U^{\sigma(K)}$  in less than 10s in all the 200 simulations and remains there afterwards, as expected by invariance. When changing the obstacle height changes, we obtain very different paths. In scenario A, the UAV navigates over the obstacles, while in scenario B the UAV navigates between the obstacles. We verified that  $x(t) \in \bigcup_{k=1}^K \mathcal{O}_U^{\sigma(k)}$ , in all simulations, and at all times. This is shown for two of the 200 Monte-Carlo runs, one for each scenario, in Fig 7.g–h, where for  $t \in [t_k, t_{k+1}]$ , the Lyapunov function  $V^{\sigma(k)}(t)$  (blue) is bounded by the associated  $\rho_U^{\sigma(k)}$  (red) demonstrating that the trajectory remains safe at all times. Furthermore,  $V^{\sigma(k)}(t)$  decays monotonically on each such time interval, and enters the ultimate set at  $t_K$ . The time instant  $t_K$  is different between scenario A and B, and also differs in the different trajectory realizations for each scenario, since it is affected by disturbances and model errors.

## IX. EXPERIMENTS WITH A CRAZYFLIE 2.1

To further validate the approach, we implement a real-time experiment using a Crazyflie 2.1 quadrotor UAV. For this purpose, the RISMP was implemented in C++ and interfaced with the `CrazySwarm` driver in ROS [44], for a scenario similar to that in Sec. VIII. The physical obstacles are shown in Fig. 8.b. A random selector determines randomly a target vertex (black) in  $\mathcal{G}$ , RISMP computes the motion plan, and then control the UAV along the path using the on-board

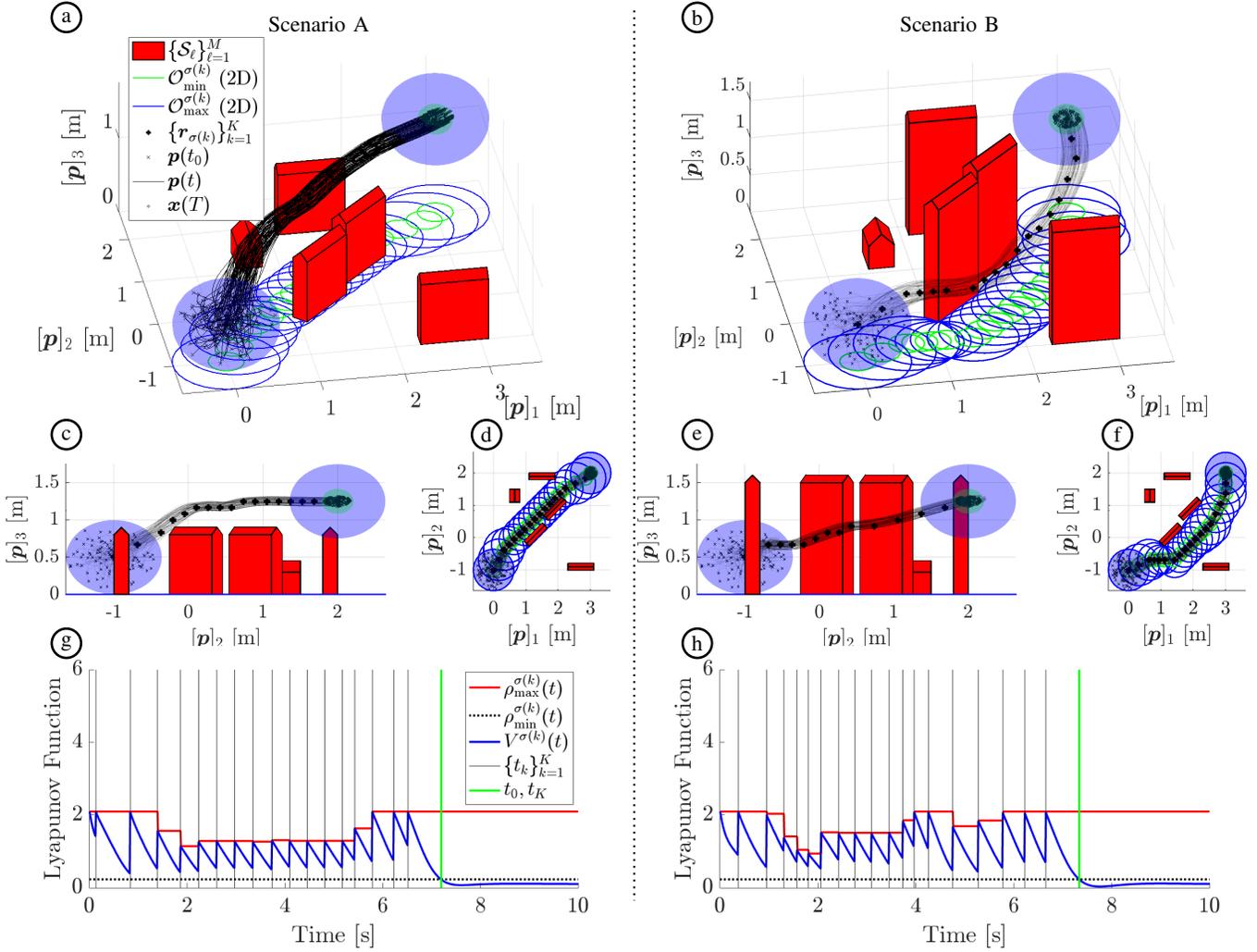


Fig. 7. Two scenarios where the RISMP computes a solution from a position close to the origin to a target position on the other end of the scene. (a)–(b) The geometry of the world (red) with the 2D-projections of the invariant sets in the  $xy$ -plane and 100 realizations of the UAV state trajectory for different initial conditions  $\mathbf{x}(t_0) \in \text{bnd}(\mathcal{O}_I^\perp)$  at the time  $t_0 = 0$  and different realizations of the disturbance (black, thin). (c)–(f) Side and top views of the same plots. (g)–(h) Time-series of the Lyapunov function in scenarios A and B, respectively. The plots illustrate  $\rho_I^{\sigma(k)}(t)$  (red) and  $\rho_U$  (black dotted) and the exponential decay of the Lyapunov function  $V^{\sigma(k)}(t)$  (blue) associated with the vertex  $v_\sigma(k)$  in the path computed by the RISMP. The sequence of setpoint update times  $\{t_k\}_{k=1}^K$  are shown in gray, and the final time  $t_K$  when the UAV enters the target PI set is shown in green.

controller, which is of the form of [30], with the update logic implemented in ROS, see Fig. 2 and executed at 50Hz. The position of the UAV is obtained from an OptiTrack motion capture system, while the remaining variables in (3) are estimated by an extended Kalman filter (EKF) that uses the position from the OptiTrack and inertial measurements. When the UAV reaches the target ultimate set (see the green lines in Fig. 8.j), a new random target is selected, which corresponds to running several missions akin to those in Sec. VIII in series. The computation of the optimal path never exceeds 2ms in a desktop computer with a 3.4GHz AMD Ryzen 9 5950X 16-cores CPU, and 128GB RAM.

The path in  $\mathcal{G}$  (gray) and the UAV motion (blue) are shown in Fig. 8.a. During the flight comprising 17 consecutive missions, the quadrotor remains airborne and avoids the obstacles at all times. A video of the experiment is available at <https://youtu.be/ewIcOwDwZC8>. As shown by the positional response, this entails significant variation in elevation

when avoiding the buildings (see Fig. 8.f). Throughout the experiment, the tracking performance is consistent with the performance in Fig. 5. The jumps in the position tracking error in Fig. 8.g-h are induced by setpoint changes, and they indicate that even in cluttered environments, the RISMP can modify the setpoint by sizable amounts while ensuring safety. In this environment, the geometric constraints are the main limitation on how large the inflated sets can be. For the Crazyflie, the thrust limit  $T_{\max} = 2mg$  is never violated, see Fig. 8.i.

Analyzing the Lyapunov function  $V^{\sigma(k)}(t)$  in Fig. 8.g-h in relation to the level sets  $\mathcal{O}_I^{\sigma(k)}$  for  $t \in [t_{k-1}, t_k]$ , we confirm that the UAV remains safe since  $V^{\sigma(k)}(t) \leq \rho_I^{\sigma(k)} < \Gamma_0$  for  $t \in [t_{k-1}, t_k]$ , for all  $k \in \mathbb{Z}_{[1, K]}$ . Zooming in, we see that the Lyapunov function is monotonically decreasing for constant setpoint, but not quite exponentially, as the disturbances, primarily imperfect model of the controller, thrust misalignment due to non-ideal attitude tracking, and airflow in the room, are time-varying. These signals estimated from EKF indicates

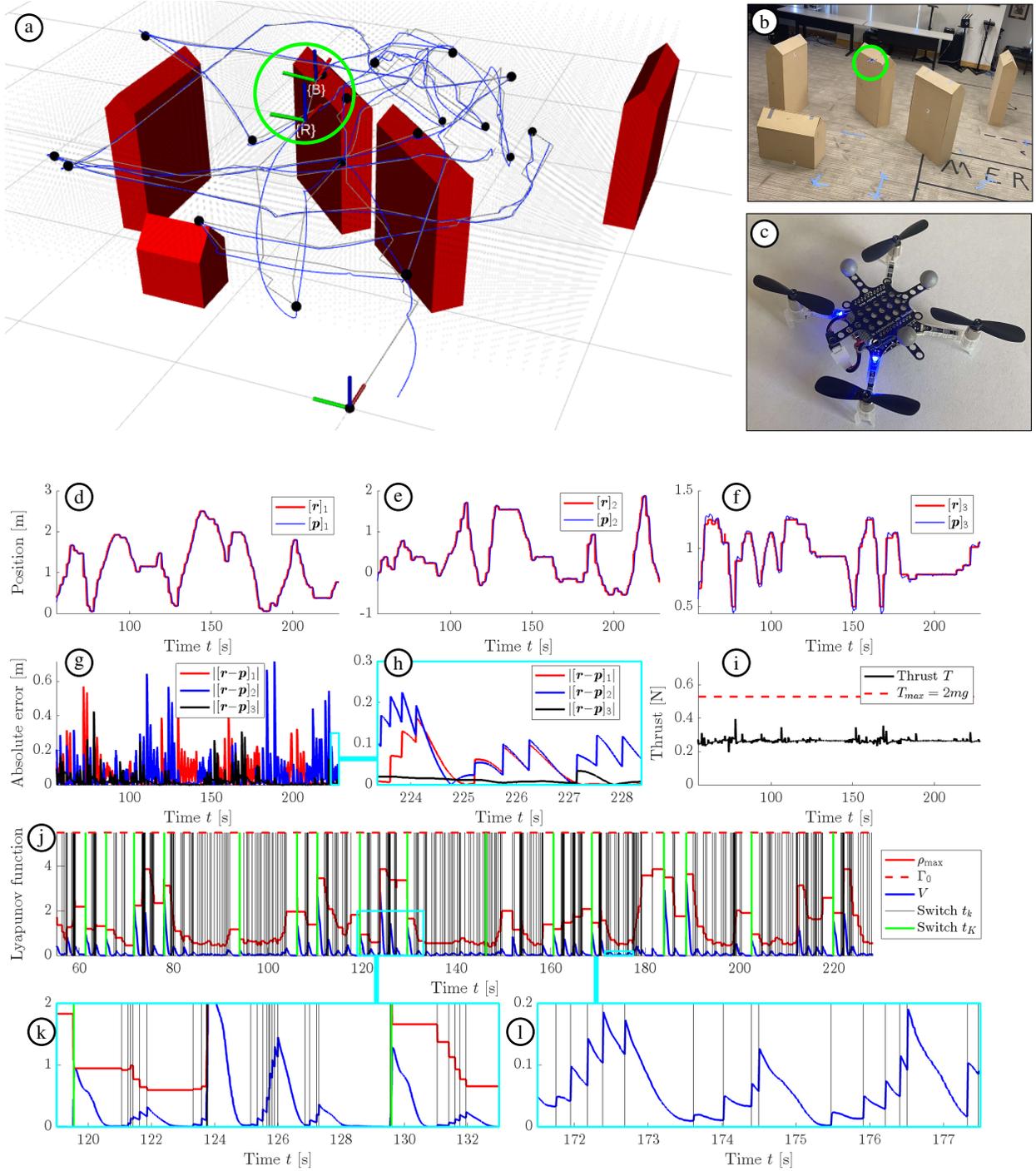


Fig. 8. System response from an experiment with the Crazyflie 2.1 UAV using the RISMP. (a) The world consists of polyhedral obstacles (red) and the Crazyflie UAV is navigating along a path between an initial vertex  $v_{\sigma(1)}$  and a terminal vertex  $v_{\sigma(K)}$  (both black) with each corresponding solution  $\{\mathbf{r}_{\sigma(k)}\}_{k=1}^K$  highlighted in gray. (b) Picture of the experiment setup. (c) Picture of the Crazyflie 2.1 used in the experiments. (d) – (f) Positional setpoints (red) and the system response (blue) in the three positional dimensions. (g) Positional setpoint tracking errors. (h) Zoom on the positional tracking errors at the end of the last maneuver. (i) Thrust used by the UAV. (j) Lyapunov function  $V^{\sigma(k)}$  (blue) and level sets  $\rho_T^{\sigma(k)}$  (red) on  $t \in [t_{k-1}, t_k]$  as a function of time. Green lines indicate the times at which a new target vertex is selected when a path is computed in the RISMP, and gray lines indicate every time instant where the constant setpoint is updated. (k) – (l) Zoom on the Lyapunov function, showing the transient behavior when the setpoint is updated and the decay of the errors.

constraint satisfaction at all times.

Thus, the results demonstrate the validity of the modeling assumptions **A1–A3** and the effectiveness of RISMP. The

control system implemented as in Fig. 2 may improve further by implementing the setpoint update logic on the quadrotor embedded system, as this would avoid communication delays.

However, since the control system is implemented on the embedded system and RISMP uses positive invariant sets, the only impact of such delays is to add unnecessary time to the setpoint update, increasing the total time for completing a mission, but it does not compromise the stability or safety of the flight.

## X. CONCLUSIONS

We developed an invariant set-based motion for quadrotors that is robust to modeling errors, imperfect attitude tracking, and polytopic uncertainty in the closed-loop gains, which safe planning and control when the UAVs closed-loop control system can only be modeled approximately.

The proposed robust invariant set-based motion planner achieves finite time convergence from the initial position to a neighborhood of the target while avoiding obstacles. The existence of solutions is related to the properties of a graph constructed from the possible target locations and can be analyzed offline. We evaluated the method both in Monte-Carlo simulations and in experiments for a real-time implementation on a Crazyflie 2.1 in ROS.

The RISMP has been primarily developed for motion planning in static environments where both online safety and offline certification of feasibility are important. However, it can be extended to handle initially unknown map and time-varying/initially unknown obstacles based on our previous results, by sampling the grid points and modifying the graph connectivity in real-time [14], [22]. Additionally, the edge weights can be made to encode perception or control performance objectives, which may be the subject of future work.

## REFERENCES

- [1] H. Freimuth and M. König, "Planning and executing construction inspections with unmanned aerial vehicles," *Automation in Construction*, vol. 96, pp. 540–553, 2018.
- [2] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, "Unmanned aerial vehicle (UAV) path planning and control assisted by augmented reality (AR): the case of indoor drones," *Int. J. Production Research*, pp. 1–22, 2023.
- [3] S. M. Lavelle, *Planning Algorithms*. Cambridge University Press, 2006.
- [4] 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.
- [5] A. Ait Saadi, A. Soukane, Y. Meraihi, A. Benmessaoud Gabis, S. Mirjalili, and A. Ramdane-Cherif, "UAV path planning using optimization approaches: A survey," *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 4233–4284, 2022.
- [6] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *J. Intell. Robotic Systems*, vol. 57, pp. 65–100, 2010.
- [7] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *Int. conf. on intelligent robots and systems*, 2019, pp. 1934–1940.
- [8] 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.
- [9] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *Int. J. Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.
- [10] T. Marcucci, P. Nobel, R. Tedrake, and S. Boyd, "Fast path planning through large collections of safe boxes," *Transactions on Robotics*, 2024.
- [11] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science robotics*, vol. 8, no. 84, 2023.
- [12] 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.
- [13] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," in *Int. conf. on robotics and automation*, 2015, pp. 42–49.
- [14] 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.
- [15] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *55th IEEE Conference on Dec. and Control*, 2016, pp. 5986–5991.
- [16] C. Danielson, K. Berntorp, S. Di Cairano, and A. Weiss, "Motion-planning for unicycles using the invariant-set motion-planner," in *American Control Conf.*, 2020, pp. 1235–1240.
- [17] C. Danielson, L. J. Bridgeman, and S. Di Cairano, "Necessary and sufficient conditions for constraint satisfaction in switched systems using switch-robust control invariant sets," *Int. J. Robust and Nonlinear Control*, vol. 29, no. 9, pp. 2589–2602, 2019.
- [18] 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 Trans. Automatic Control*, vol. 65, no. 10, pp. 4456–4463, 2020.
- [19] A. Weiss, C. Danielson, K. Berntorp, I. Kolmanovsky, and S. Di Cairano, "Motion planning with invariant set trees," in *IEEE Conf. Control Tech. and Applications*, 2017, pp. 1625–1630.
- [20] A. Weiss, F. Leve, M. Baldwin, J. R. Forbes, and I. Kolmanovsky, "Spacecraft constrained attitude control using positively invariant constraint admissible sets on  $SO(3) \times R^3$ ," in *American Control Conf.*, 2014, pp. 4955–4960.
- [21] C. Danielson, J. Kloeppe, and C. Petersen, "Spacecraft attitude control using the invariant-set motion-planner," *IEEE Control Systems Letters*, vol. 6, pp. 1700–1705, 2021.
- [22] 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 Trans. Intelligent Vehicles*, vol. 5, no. 1, pp. 112–126, 2019.
- [23] C. Danielson and T. Brandt, "Constraint admissible positive invariant sets for vehicles in  $SE(3)$ ," *IEEE Control Systems Letters*, 2023.
- [24] N. Michel, "Invariant set design for the constrained control of a quadrotor," Ph.D. dissertation, Université Paris-Saclay, 2020.
- [25] M. M. Nicotra and E. Garone, "The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems," *Control Systems Magazine*, vol. 38, no. 4, pp. 89–107, 2018.
- [26] M. M. Nicotra, R. Naldi, and E. Garone, "A robust explicit reference governor for constrained control of unmanned aerial vehicles," in *American Control Conf.*, 2016, pp. 6284–6289.
- [27] 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.
- [28] H. Sinhmar, M. Greiff, and S. Di Cairano, "Practical and safe navigation function based motion planning of UAVs," in *2024 International Conference on Robotics and Automation (ICRA)*, 2024, pp. 1008–1013.
- [29] M. Greiff, A. Weiss, K. Berntorp, and S. Di Cairano, "A robust invariant set planner for quadrotors," in *European Control Conf.*, 2024, pp. 1455–1460.
- [30] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on  $SE(3)$ ," in *49th IEEE Conference on Dec. and Control*, 2010, pp. 5420–5425.
- [31] T. Lee, "Computational geometric mechanics and control of rigid bodies," Ph.D. dissertation, University of Michigan, 2008.
- [32] F. Goodarzi, D. Lee, and T. Lee, "Geometric nonlinear PID control of a quadrotor UAV on  $SE(3)$ ," in *European Control Conf.*, 2013, pp. 3845–3850.
- [33] M. Greiff, Z. Sun, and A. Robertsson, "Attitude control on  $SU(2)$ : Stability, robustness, and similarities," in *Control System Letters*, 2022, pp. 73–78.
- [34] M. Greiff, "Nonlinear control of unmanned aerial vehicles: Systems with an attitude," Ph.D. dissertation, Lund University, 2021.
- [35] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer, 1997, vol. 8.
- [36] H. K. Khalil, "Nonlinear systems third edition," *Patience Hall*, vol. 115, 2002.
- [37] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [38] M. M. Nicotra and E. Garone, "Explicit reference governor for continuous time nonlinear systems subject to convex constraints," in *American Control Conf.*, 2015, pp. 4561–4566.
- [39] J. Evans, *Optimization algorithms for networks and graphs*. CRC Press, 2017.

- [40] L. Ljung and T. Glad, *Modeling of dynamic systems*. Prentice-Hall, Inc., 1994.
- [41] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014.
- [42] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 - a MATLAB software package for semidefinite programming, version 1.3," *Optimization methods and software*, vol. 11, no. 1-4, pp. 545-581, 1999.
- [43] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual*. Pearson Education, 2001.
- [44] J. A. Preiss, W. Hönig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Int. Conf. Robotics and Automation*, 2017, pp. 3299-3304.

## APPENDIX

## A. Proof of Lemma 1

**Proof 1** As  $\mathbf{C}$  is symmetric, it can be written as

$$\mathbf{C} = \begin{bmatrix} \Phi \\ \Psi \end{bmatrix}^\top \underbrace{\begin{bmatrix} \mathbf{0} & (\mathbf{I} - \tilde{\mathbf{R}}^\top) \\ * & \mathbf{0} \end{bmatrix}}_{\triangleq \mathbf{D}} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} \preceq (\Phi^\top \Phi + \Psi^\top \Psi) \bar{\lambda}(\mathbf{D}).$$

To compute the spectrum of  $\mathbf{D}$ , let  $\mathbf{E} = 2\mathbf{I} - \tilde{\mathbf{R}}^\top - \tilde{\mathbf{R}}$ . Then

$$\begin{aligned} \det(x\mathbf{I} - \mathbf{D}) &= \det(x^2\mathbf{I} - (\mathbf{I} - \tilde{\mathbf{R}}^\top)^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top)) \\ &= \prod_i (x^2 - \lambda_i((\mathbf{I} - \tilde{\mathbf{R}}^\top)^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top))) \\ &= \prod_i (x^2 - \lambda_i(\mathbf{E})) \\ &= \prod_i (x - \sqrt{\lambda_i(\mathbf{E})})(x + \sqrt{\lambda_i(\mathbf{E})}), \end{aligned}$$

where the last equality holds if  $\lambda_i(\mathbf{E}) \geq 0$  for all  $i$  (which we can indeed show to be the case). For any unit vector  $\mathbf{w} \perp \boldsymbol{\nu}$ , we have that  $\mathbf{w}^\top \tilde{\mathbf{R}}\mathbf{w} = \mathbf{w}^\top \tilde{\mathbf{R}}^\top \mathbf{w} = \cos(\alpha)$ , thus

$$\mathbf{w}^\top (2\mathbf{I} - \tilde{\mathbf{R}} - \tilde{\mathbf{R}}^\top) \mathbf{w} = \begin{cases} (\mathbf{w}^\top \mathbf{w}) 2(1 - \cos(\alpha)) & \text{if } \mathbf{w} \perp \boldsymbol{\nu} \\ (\mathbf{w}^\top \mathbf{w}) 0 & \text{if } \mathbf{w} \parallel \boldsymbol{\nu} \end{cases}$$

as illustrated in Fig. 9. The spectrum of  $2\mathbf{I} - \tilde{\mathbf{R}} - \tilde{\mathbf{R}}^\top$  is characterized by one eigenvalue  $2(1 - \cos(\alpha))$  of multiplicity 2, and one eigenvalue 0 of multiplicity 1. As a result, the spectrum of  $\mathbf{D}$  has eigenvalues of  $\pm \sqrt{2(1 - \cos(\alpha))}$  of multiplicity 2, and 0 with multiplicity 2. Thus,  $\bar{\lambda}(\mathbf{D}) = \sqrt{2(1 - \cos(\alpha))}$ , concluding the proof. ■

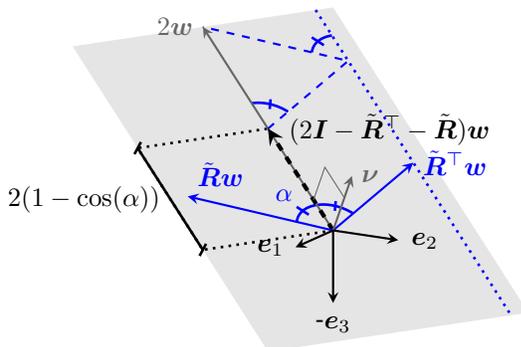


Fig. 9. Geometric intuition of the proof of Lemma 1.

## B. Proof of Lemma 2

**Proof 2** Under the assumptions

$$\sum_{h=1}^N \zeta_h \begin{bmatrix} \bar{\mathbf{K}} & \mathbf{K}_h^\top \\ \mathbf{K}_h & \mathbf{I} \end{bmatrix} \succeq \mathbf{0}, \quad \forall \zeta_h : \zeta_h \geq 0, \sum_{h=1}^N \zeta_h = 1,$$

and as a consequence,

$$\begin{bmatrix} \bar{\mathbf{K}} & \mathbf{K}^\top \\ \mathbf{K} & \mathbf{I} \end{bmatrix} \succeq \mathbf{0} \quad \forall \mathbf{K} \in \mathcal{K}. \quad (23a)$$

By taking the Schur complement, we obtain

$$\bar{\mathbf{K}} \succeq \mathbf{K}^\top \mathbf{K} \quad \forall \mathbf{K} \in \mathcal{K}$$

concluding the proof. ■

## C. Proof of Proposition 1

**Proof 3** For simplicity, we drop the sub-index of the reference  $\mathbf{r}_k$  and let  $\mathbf{x}_e = (\mathbf{p} - \mathbf{r}; \mathbf{v})$ . The error dynamics are

$$\dot{\mathbf{x}}_e = \bar{\mathbf{A}}\mathbf{x}_e + \mathbf{B}\Delta, \quad (24)$$

where

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\tilde{\mathbf{R}}^\top \mathbf{K}_p & -\tilde{\mathbf{R}}^\top \mathbf{K}_v \end{bmatrix}, \quad \mathbf{K} \in \mathcal{K}. \quad (25)$$

Let  $\mathbf{A} = \bar{\mathbf{A}}$  when  $\tilde{\mathbf{R}} \equiv \mathbf{I}$ , we can consider the effect of the attitude tracking error as a perturbation of some nominal dynamics,

$$\mathbf{P}\bar{\mathbf{A}} = \mathbf{P}\mathbf{A} + \mathbf{N}^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top) \mathbf{K}, \quad (26)$$

where

$$\mathbf{N}^\top = \begin{bmatrix} \mathbf{P}_{p\mathbf{v}} \\ \mathbf{P}_{v\mathbf{v}} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} = \mathbf{P}\mathbf{B}. \quad (27)$$

Differentiating the Lyapunov function  $V = \|\mathbf{x}_e\|_{\mathbf{P}}^2$  along (3),

$$(\mathrm{d}/\mathrm{d}t)V \leq \underbrace{G(\mathbf{x}_e, \Delta)}_{\text{nominal}} + \underbrace{\mathbf{x}_e^\top \mathbf{M} \mathbf{x}_e}_{\text{att.pert.}}$$

where the nominal part  $G(\mathbf{x}_e, \Delta)$  is similar to the expressions with perfect attitude tracking, and the component that is due to the bounded attitude tracking error is

$$\mathbf{M} = (\mathbf{N}^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top) \mathbf{K})^\top + \mathbf{N}^\top (\mathbf{I} - \tilde{\mathbf{R}}^\top) \mathbf{K}. \quad (28)$$

We use Lemma 1 to bound  $\mathbf{M}$  for all  $\mathbf{K} \in \mathcal{K}$  and  $\alpha \leq \alpha_{\max}$ ,

$$\begin{aligned} \mathbf{M} &\preceq [\mathbf{N}^\top \mathbf{N} + \mathbf{K}^\top \mathbf{K}] \sqrt{2(1 - \cos(\alpha))} \\ &\preceq \beta (\mathbf{P}^\top \mathbf{B} \mathbf{B}^\top \mathbf{P} + \mathbf{K}^\top \mathbf{K}), \end{aligned}$$

where  $\beta = \sqrt{2(1 - \cos(\alpha_{\max}))}$ , which holds for any  $\tilde{\mathbf{R}} \in \mathrm{SO}(3)$  with a rotation angle  $\alpha \leq \alpha_{\max} < \pi/2$ . The dependence on  $\alpha$  can be eliminated as  $\sqrt{2(1 - \cos(\alpha))}$  increases monotonically in  $\alpha \in [0, \pi]$ , attaining a maximum at  $\alpha = \alpha_{\max}$ . By Lemma 2, we obtain

$$\mathbf{M} \preceq \beta (\mathbf{P}^\top \mathbf{B} \mathbf{B}^\top \mathbf{P} + \bar{\mathbf{K}}), \quad (29)$$

for any  $\bar{\mathbf{K}} \in \mathbb{S}_{++}^6$  that satisfies

$$\begin{bmatrix} \bar{\mathbf{K}} & \mathbf{K}_h^\top \\ \mathbf{K}_h & \mathbf{I} \end{bmatrix} \succeq \mathbf{0} \quad \forall h \in \mathbb{Z}_{[1, N]}. \quad (30)$$

Using (30) and taking the Schur complement we obtain

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}_h^\top \mathbf{P} + \mathbf{P} \mathbf{A}_h + \mathbf{P} + \beta \bar{\mathbf{K}} & \mathbf{P} \mathbf{B} & \sqrt{\beta} \mathbf{P} \mathbf{B} \\ \mathbf{B}^\top \mathbf{P} & -\gamma \mathbf{I} & \mathbf{0} \\ \sqrt{\beta} \mathbf{B}^\top \mathbf{P} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \preceq \mathbf{0} \forall h, \\ & \Rightarrow \begin{bmatrix} \mathbf{A}_h^\top \mathbf{P} + \mathbf{P} \mathbf{A}_h + \mathbf{P} + \mathbf{M} & \mathbf{P} \mathbf{B} \\ \mathbf{B}^\top \mathbf{P} & -\gamma \mathbf{I} \end{bmatrix} \preceq \mathbf{0} \forall h \end{aligned} \quad (31)$$

Assume that  $(\mathbf{P}, \gamma) \in \mathbb{S}_{++}^6 \times \mathbb{R}_{>0}$  solves (9). Given the definition of  $\mathcal{K}$  in Assumption A2, there exists a set  $\{\mathbf{A}_h\}_{h=1}^N$  such that  $\mathbf{A} = \sum_h \zeta_h \mathbf{A}_h$  for some  $\zeta_h > 0$  such that  $\sum_h \zeta_h = 1$ . Differentiating the Lyapunov function  $V = \|\mathbf{x}_e\|_{\mathbf{P}}^2$  along the solutions of (3),

$$\begin{aligned} (d/dt)V(\mathbf{p}, \mathbf{v}, \mathbf{r}) &= \mathbf{x}_e^\top \mathbf{P} \dot{\mathbf{x}}_e + \dot{\mathbf{x}}_e^\top \mathbf{P} \mathbf{x}_e \\ &= \mathbf{x}_e^\top \mathbf{P} (\bar{\mathbf{A}} \mathbf{x}_e + \mathbf{B} \Delta) + (\bar{\mathbf{A}} \mathbf{x}_e + \mathbf{B} \Delta)^\top \mathbf{P} \mathbf{x}_e \\ &= \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix}^\top \begin{bmatrix} \bar{\mathbf{A}}^\top \mathbf{P} + \mathbf{P} \bar{\mathbf{A}} & \mathbf{P} \mathbf{B} \\ \mathbf{B}^\top \mathbf{P} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix}^\top \begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{M} & \mathbf{P} \mathbf{B} \\ \mathbf{B}^\top \mathbf{P} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix} \\ &= \sum_{h=1}^N \zeta_h \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix}^\top \begin{bmatrix} \mathbf{A}_h^\top \mathbf{P} + \mathbf{P} \mathbf{A}_h + \mathbf{M} & \mathbf{P} \mathbf{B} \\ \mathbf{B}^\top \mathbf{P} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix} \\ &\leq -\sum_{h=1}^N \zeta_h \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix}^\top \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & -\gamma \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \Delta \end{bmatrix} \\ &= -\mathbf{x}_e^\top \mathbf{P} \mathbf{x}_e + \gamma \Delta^\top \Delta \\ &\leq -V(\mathbf{p}, \mathbf{v}, \mathbf{r}) + \gamma \Delta_{\max}^2, \end{aligned}$$

where the first inequality follows by (31). Thus,  $\mathcal{O}_{\mathcal{U}}$  is an ultimate set as per Definition 4 with  $\rho_{\mathcal{U}} = \gamma \Delta_{\max}^2$  for all constant  $\mathbf{r}$ . ■



**Marcus Greiff** (Member, IEEE) received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from Lund University in 2017 and 2021, respectively. He was a visiting research scientist with Mitsubishi Electric Research Labs until 2024, and is now a Research Scientist with Toyota Research Institute. His research topics concern nonlinear control, motion planning, and estimation theory in the context of quad-rotor UAV and automotive control. He has written/coauthored more than 50 peer-reviewed papers and patents, and

is the winner of two separate paper awards at the IEEE Conference on Control Technology and Applications (CCTA) in 2020 and 2022, respectively.



**Himani Sinhmar** received the Ph.D. degree specializing in robotics from Cornell University in 2024. She earned her Bachelor and Master of Technology from IIT Bombay in 2018 and 2019 respectively. Her research focuses on developing certifiable-safe controllers and motion planners for robotic systems with limited onboard capabilities. Sinhmar's interdisciplinary approach integrates control theory, collective intelligence, and formal methods. Her overarching goal is to advance both practical and theoretically grounded solutions, enhancing the safety and guaranteed task completion of robotic systems in unstructured, dynamic, and unknown environments.

anteed task completion of robotic systems in unstructured, dynamic, and unknown environments.



**Avishai Weiss** (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 2008 and 2009, respectively, and the Ph.D. degree in aerospace engineering from the University of Michigan, Ann Arbor, MI, USA, in 2013. He is currently a Senior Principal Research Scientist at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA. His main research interests and contributions are in the areas of spacecraft orbital and attitude control, constrained control, model predictive control, motion planning, and time-varying systems, in which he has authored/coauthored more than 90 peer-reviewed papers and patents.



**Karl Berntorp** (Senior Member, IEEE) received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from Lund University, Lund, Sweden, in 2009 and 2014, respectively. In 2008, he was a Visiting Researcher with Daimler AG, Sindelfingen, Germany. Since 2014, he has been working with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, where he is currently a Senior Principal Research Scientist and a Team Leader. He has authored or coauthored more than 110 peer-reviewed papers in journals and

conferences and 50 patents. His work includes the design and implementation of estimation, constrained control, motion planning, and learning algorithms. His research interests include statistical signal processing, Bayesian inference and learning, sensor fusion, and optimization-based control, with applications to automotive, transportation, positioning, navigation, and communication systems. Dr. Berntorp is an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY and the CCTA Editorial Board.



**Stefano Di Cairano** (Senior Member, IEEE) received the Master's (Laurea) and the Ph.D. degrees in information engineering in 2004 and 2008, respectively, from the University of Siena, Italy. During 2008-2011, he was with Powertrain Control R&A, Ford Research and Advanced Engineering, Dearborn, MI, USA. Since 2011, he is with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, where he is currently a Deputy Director, and a Distinguished Research Scientist. His research focuses on optimization-based control and decision-

making strategies for complex mechatronic systems, in automotive, factory automation, transportation systems, and aerospace. His research interests include model predictive control, constrained control, path planning, hybrid systems, optimization, and particle filtering. He has authored/coauthored more than 250 peer-reviewed papers in journals and conference proceedings and 85 patents. Dr. Di Cairano was the Chair of the IEEE CSS Technical Committee on Automotive Controls and of the IEEE CSS Standing Committee on Standards. He is the inaugural Chair of the IEEE CCTA Editorial Board and was an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY.