

Practical and Safe Navigation Function Based Motion Planning of UAVs

Sinhmar, Himani; Greiff, Marcus; Di Cairano, Stefano;

TR2024-055 May 14, 2024

Abstract

This paper offers a practical method for certifiably safe operations of an unmanned aerial vehicle (UAV) with limited power and computation, useful for real-time operations where the UAV is exposed to significant disturbances in non-convex free space. We propose a motion planning method based on the Explicit Reference Governor (ERG) framework to ensure the safety of a flying quadrotor UAV. From a small set of experiment data and assumptions on modeling errors, a Lyapunov function is synthesized by which an ERG is constructed to modify the UAV set-points. The method can handle polyhedral obstacles and constraints imposed on the maximum thrust of the UAV and its maximum tilt. We demonstrate the approach with extensive simulations and experiments using a Crazyflie 2.1.

IEEE International Conference on Robotics and Automation (ICRA) 2024

Practical and Safe Navigation Function Based Motion Planning of UAVs

Himani Sinhmar¹, Marcus Greiff², Stefano Di Cairano³

Abstract—This paper offers a practical method for certifiably safe operations of an unmanned aerial vehicle (UAV) with limited power and computation, useful for real-time operations where the UAV is exposed to significant disturbances in non-convex free space. We propose a motion planning method based on the Explicit Reference Governor (ERG) framework to ensure the safety of a flying quadrotor UAV. From a small set of experiment data and assumptions on modeling errors, a Lyapunov function is synthesized by which an ERG is constructed to modify the UAV set-points. The method can handle polyhedral obstacles and constraints imposed on the maximum thrust of the UAV and its maximum tilt. We demonstrate the approach with extensive simulations and experiments using a Crazyflie 2.1.

I. INTRODUCTION

A fundamental aspect of motion planning (MP) is that the robot must not only reach its goal but also satisfy a set of constraints at all times in the presence of modeling errors and disturbances. The constraints may include avoiding obstacles, ensuring actuator bounds and other limits on the robot state.

An appealing approach is Model Predictive Control, where optimal control problems are solved in a receding horizon fashion [1], [2]. The main advantage of such optimization-based approaches is that they optimize performance metrics while ensuring constraint satisfaction. However, when implemented on hardware with limited capabilities, the computational cost may become prohibitive if the constraints are non-linear and non-convex in the decision variables [3].

Recently, there has been a rising interest in incorporating Control Barrier functions (CBFs) into planning frameworks for safety-critical applications [4]–[6]. A CBF guarantees that the system stays within a permissible set of states indefinitely when choosing an admissible control input from a given set [7]. However, this leads to one of the major drawbacks of CBFs, which is that the resulting constraints may lack a feasible solution if there are input constraints (see, e.g., [8, Chapter 3]). In contrast, we seek feasibility guarantees to ensure that safety is retained.

For this purpose, set-based path planning methods are attractive. These approaches, akin to sampling-based algorithms [9], [10], tackle path planning through a graph search. Invariant-set motion planners (ISMP) in [11]–[14] incorporate closed-loop system dynamics into the search graph using constraint admissible positive invariant (PI) sets. The computational cost remains low for ISMP since the PI sets

are based on time-invariant closed-loop dynamics, allowing for offline computation of the graph. However, storing the graph may require a prohibitive amount of memory.

An approach that avoids the computational burden of an MPC, potential lack of feasibility of the CBF approach, and memory issues in ISMP is the Reference Governor (RG). This method enforces state and input constraints by dynamically modifying set-points of systems that are pre-stabilized [15]–[18]. In [19], [20] an Explicit Reference Governor (ERG) is proposed to manipulate the rate of change of the applied reference to the stabilizing controller. The ERG design hinges on finding a *dynamic margin* and a *navigation field*, given the knowledge of the environment. The dynamic margin is a non-negative scalar function that determines the maximum allowable rate of change for the reference without violating the constraints. The navigation field specifies the direction in which the reference should progress. In [21] the authors propose to use vector field planners such as artificial potentials [22], [23] and navigation functions [24] to ensure the absence of local minima. Navigation functions [25]–[29] offer theoretical guarantees for global convergence in topologically spherical spaces but struggle with obstacles featuring sharp corners, requiring complex parameter tuning and hindering practical implementation [28, Sec. IV.B].

A major advantage of the ERG is that it can be implemented in real-time on UAV hardware [12], [20]. However, these prior works only consider spherical and cylindrical obstacles in the environment, while treating the boundary as a combination of planar walls. To guarantee the correctness of the ERG in a bounded 3D environment cluttered with polyhedral obstacles and non-spherical boundaries, we refine the notion of safe navigation function. Additionally, we develop the dynamic margin considering uncertainties that arise from modeling errors as well as from partial knowledge of closed-loop system characteristics.

Contributions: In this paper, motivated by system identification, we extend the ERG framework proposed in [20] by considering non-homogeneous controller gains. Furthermore, we introduce a refined practical version of navigation functions tailored for utilization within an environment containing polyhedral obstacles. We validate our proposed control strategy both in simulations and in real-time on a small UAV.

Notation: Vectors are written $\mathbf{x} \in \mathbb{R}^n$, where $[\mathbf{x}]_i$ is the i th element of \mathbf{x} , and \mathbf{e}_i is a unit vector $[\mathbf{e}_i]_i = 1$. $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ and $\mathbf{0}$ are the identity and zero-matrices, respectively. We let $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top \mathbf{x}$ with $\|\mathbf{x}\|_{\mathcal{P}}^2 = \mathbf{x}^\top \mathbf{P} \mathbf{x}$, and take $(\mathbb{S}_+^n) \mathbb{S}_{++}^n$ to be the cone of $n \times n$ positive (semi)definite matrices. The set of rotations is denoted $\text{SO}(3)$. We let $E \sim \mathcal{U}(\mathcal{W})$ be a uniform selection of $E \in \mathcal{W}$, and denote all convex combinations

*The work of H. Sinhmar and M. Greiff was done while at MERL.

¹Cornell University, Ithaca, NY, USA. Email: hs962@cornell.edu

²Toyota Research Institute (TRI), 94022 Los Altos, CA, USA. Email: marcus.greiff@tri.global.

³Mitsubishi Electric Research Labs (MERL), 02139 Cambridge, MA, USA. Email: dicairano@merl.com.

of the elements in \mathcal{W} by $\text{Co}(\mathcal{W})$. The position and velocity of the UAV are $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$, respectively, \mathbf{r} is a reference modified by the ERG, and \mathbf{r}_G is a constant goal.

II. PROBLEM FORMULATION

In this section, the assumptions of the paper are stated with a brief motivation based on experiment data before introducing the problem and giving some mathematical preliminaries.

Assumption 1 *The translation motion of a closed-loop UAV system can be modeled by second-order dynamics,*

$$\ddot{\mathbf{p}} = -\mathbf{K}_p(\mathbf{p} - \mathbf{r}) - \mathbf{K}_v\mathbf{v} + \Delta, \quad (1)$$

where $\Delta \in \mathbb{R}^3$ is an unknown bounded disturbance and the response is characterized by gains $\mathbf{K}_v \in \mathbb{S}_{++}^3$, $\mathbf{K}_p \in \mathbb{S}_{++}^3$.

Unlike [20], we do not assume perfect knowledge of the gains defining the translational subsystem under ideal conditions ($\Delta \equiv \mathbf{0}$), nor do we assume these gains to have uniform diagonals. Instead, we assume the following.

Assumption 2 *The feedback gains are known such that*

$$\mathbf{K} = (\mathbf{K}_p, \mathbf{K}_v) \in \text{Co}(\{(\mathbf{K}_p^i, \mathbf{K}_v^i)\}_{i=1}^N) \triangleq \mathcal{K}, \quad (2)$$

where $\mathbf{K}_p^i, \mathbf{K}_v^i$ are diagonal matrices with positive entries.

To motivate the assumptions experimentally, we consider data from multiple flights with a Crayflie 2.1 following a sequence of target positions in space. We estimate the system response by an extended Kalman filter from IMU and motion capture data, and find the gains a grey-box identification [30],

$$\mathbf{K}_p = \text{diag}(7.78, 7.38, 11.30), \quad (3a)$$

$$\mathbf{K}_v = \text{diag}(3.28, 3.27, 3.75). \quad (3b)$$

Clearly, the system response in $[\mathbf{p}]_3$ is less damped and faster than in the $[\mathbf{p}]_1[\mathbf{p}]_2$ -plane, motivating both assumptions. The modeling errors are bounded, and in this experiment, we note that $\|\Delta\|_2 \leq 1.0$ when ascribing all un-modeled accelerations in (1) to Δ subject to the parameters in (3). However, there is some variability in the system response, motivating uncertainty in the gains as per assumption **A2**.

Problem: *We seek to control a UAV within a world boundary \mathcal{S}_0 containing a set of M disjoint polyhedral obstacles \mathcal{S}_i . The set $\mathbb{F} = \mathcal{S}_0 \setminus \{\mathcal{S}_i\}_{i=1}^M \subseteq \mathbb{R}^3$ is referred to as the free-space, and given assumptions **A1**, **A2**, the problem is to compute a trajectory $\mathbf{r} : \mathbb{R}_+ \mapsto \mathbb{R}^3$ where:*

- $\lim_{t \rightarrow \infty} \mathbf{p}(t) = \mathbf{r}_G$ for any $(\mathbf{p}(0), \mathbf{r}_G) \in \mathbb{F} \times \mathbb{F}$;
- $\mathbf{p}(t) \in \mathbb{F}$ at all times $t \geq 0$, thus retaining safety;
- the UAV thrust and tilt are both bounded at all times.

III. PRELIMINARIES

Assumption **A2** guarantees that each controller in \mathcal{K} is stabilizing [31, Thm. 1]. By converse Lyapunov theory [32],

$$\exists \mathbf{P} \in \mathbb{S}_{++}^3 \text{ such that } V = \|\mathbf{x}_e\|_{\mathbf{P}}^2, \text{ with } \dot{V} \leq -\|\mathbf{x}_e\|_2^2, \quad (4)$$

where $\mathbf{x}_e^\top = (\mathbf{p}^\top - \mathbf{r}^\top; \mathbf{v}^\top)$. Furthermore, by the differential independence of the positions induced by Assumption **A2**, we can find such \mathbf{P} with block-diagonal structure, that is

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{pp} & \mathbf{P}_{pv} \\ \star & \mathbf{P}_{vv} \end{bmatrix}, \quad (5)$$

where $\mathbf{P}_{pp}, \mathbf{P}_{pv}, \mathbf{P}_{vv} \in \mathbb{S}_{++}^3$ are diagonal matrices. Therefore, we compute a common Lyapunov function with block diagonal structure, i.e., a single \mathbf{P} by which (4) holds for all $\mathbf{K} \in \mathcal{K}$. It is well known that the existence of such a function is equivalent to the existence of a $\mathbf{P} \succ \mathbf{0}$ that solves a Lyapunov equation for (1) characterized by the gains $\{(\mathbf{K}_p^i, \mathbf{K}_v^i)\}_{i=1}^N$. The computation of \mathbf{P} can be formulated as a semi-definite program and solved using tools such as CVX [33] similar to [34, Chapter 6]. In the following, V is a common Lyapunov function with a \mathbf{P} structured as in (5).

IV. ERG FOR POLYHEDRAL WORKSPACES

In contrast with common RG schemes, the ERG in [19] manipulates the derivative of the augmented reference $\dot{\mathbf{r}}$ which is the product of two components, the *vector field* and the *dynamic margin*. The *vector field*, $\boldsymbol{\rho}(\mathbf{r}, \mathbf{r}_G)$, indicates the direction towards which $\mathbf{r}(t)$ should evolve to ensure convergence to \mathbf{r}_G . The *dynamic margin* is a function of the current state of the UAV and the augmented reference. Its value indicates how much we can change \mathbf{r} , without violating the constraints. The dynamic margin can be obtained by translating constraints on the states and control signals to constraints on the Lyapunov function, $V \leq \Gamma$. The resulting safety constraint can then be enforced by integrating the reference according to,

$$\dot{\mathbf{r}}(t) = \kappa(\Gamma(\mathbf{r}) - V(\mathbf{p}, \mathbf{v}, \mathbf{r}))\boldsymbol{\rho}(\mathbf{r}, \mathbf{r}_G), \quad (6)$$

where $\kappa \in \mathbb{R}_+$ is an arbitrarily large scalar and affects the stiffness of the resulting differential equation, and $(\Gamma(\mathbf{r}) - V(\mathbf{p}, \mathbf{v}, \mathbf{r}))$ is the dynamic margin. In the following sections, we delve into the details of computing both the dynamic margin and the vector field, while considering the practical implementation for UAVs.

V. VECTOR FIELD USING NAVIGATION FUNCTIONS

The objective of the navigation field is to generate a continuous path in the free space \mathbb{F} that connects the current augmented reference \mathbf{r} to the desired reference \mathbf{r}_G . For path planning on smooth manifolds with boundaries, this is a well-established technique [25]. However, in a polyhedral world, the tangent space is not well defined everywhere [28].

A. Obstacle Inflation

To incorporate the body of the UAV and uncertainty in the UAV control we inflate each obstacle in the environment by a margin $\delta \in \mathbb{R}_+$, which can be obtained by preliminary experiments along the lines of [20]. Performing such inflation enables planning for a point-mass robot.

B. Polyhedral obstacle

To construct a navigation function, the geometric data for all the obstacles in the environment is assumed to be known. We define an *obstacle function* $\beta_i: \mathbb{R}^3 \mapsto \mathbb{R}$ for each obstacle $\mathcal{S}_i = \{\mathbf{r}: \beta_i(\mathbf{r}) \leq 0\}$. This function is zero for any point on the boundary of the associated obstacle, and increases monotonically away from the obstacle. For a smooth navigation function, it is required that the obstacle function is continuous and differentiable. As illustrated in

[35], we can express β_i as a boolean combination of linear inequalities defining the half-planes of the polyhedron. But this becomes prohibitively complex as the vertices of the polyhedron increase. To deal with this [36] introduced a modeling method by approximating a 3D cube as a squirecle resulting in a rounded and inflated cube, which is limiting.

Instead, we propose two different approaches that are not based on an implicit representation of a polyhedron's boundary and the use of so-called purging transformations [36]. The first method is to find the closest point on the boundary of a polyhedron, \mathcal{S}_i from a point $\mathbf{r} \in \mathbb{F}$ by solving a simple quadratic program (QP). The obstacle function is then defined as the Euclidean distance between the two points, that is, $\beta_i(\mathbf{r}) = \min_{\mathbf{q} \in \mathcal{S}_i^\delta} \|\mathbf{r} - \mathbf{q}\|$ where \mathcal{S}_i^δ is the Minkowski sum of the obstacle \mathcal{S}_i and a ball with radius δ . However, to avoid solving the QPs online, we also propose a simpler formulation where we compute the intersection of a ray defined by the center of the polyhedron $\mathbf{c}_i \in \mathcal{S}_i$ and the point \mathbf{r} with the polyhedron boundary, that is, let the intersection point be \mathbf{q}^* , then $\beta_i(\mathbf{r}) = \|\mathbf{r} - \mathbf{q}^*\|$. Both approaches ensure that the obstacle function is simple to compute, and the latter one is faster but lacks the properties required to give formal guarantees [28]. Nonetheless, it will be shown to work well in the forthcoming experiments.

C. Polyhedral environment boundary

In the existing literature [25]–[29], the environment boundary is mostly assumed to be disc-shaped, which works well for workspaces in \mathbb{R}^2 . For bounded workspaces in \mathbb{R}^3 , especially for indoor environments, we need to impose constraints for both the floor and the ceiling. Hence, it makes sense to consider a polyhedral boundary, here denoted by \mathcal{S}_0 . To compute an analogous obstacle function β_0 we need to find the distance between a point in the interior of \mathcal{S}_0 and the boundary of \mathcal{S}_0 , which is a non-convex problem. Instead, we use Euclidean distances resulting from the point-center ray intersection with the boundary, that is, $\beta_0(\mathbf{r}) = \frac{\|\mathbf{r} - \mathbf{q}^*\|}{\|\mathbf{c}_0 - \mathbf{q}^*\|}$.

D. Constructing diffeomorphisms

The navigation functions have an attractive property of being invariant under a coordinate transformation. Therefore, if we can find a diffeomorphism between the polyhedral world and the spherical world, we can use the navigation function, ψ , defined for a spherical world in (7) which is guaranteed to have a single local minima at the goal point for a sufficiently large value of k (see, e.g., [24]),

$$\hat{\psi}(\mathbf{r}) = \frac{\|\mathbf{r} - \mathbf{r}_G\|^2}{(\|\mathbf{r} - \mathbf{r}_G\|^{2k} + \prod_{i=0}^M \beta_i(\mathbf{r}))^{1/k}}. \quad (7)$$

In (7), the navigation function steepens when increasing k , resulting in the critical points gravitating towards the goal point and all the local minima turning into saddle points near the obstacles. However, using a large value of k is not always practically viable as it results in a gradient field that varies abruptly, making it unsuitable for implementation in real world as detailed in [28, Sec. IV.B]. We can avoid this aggressive tuning of the navigation parameters by scaling

down the workspace to reduce the lower bound on k . To this end, let the original polyhedral world be denoted by \mathcal{F}^* , the scaled polyhedral world by \mathcal{F} , and the sphere world by \mathcal{M} .

We propose two diffeomorphisms for implementing a navigation field in a real setting, with h_α and h_λ , such that $\mathcal{F}^* \xrightarrow{h_\alpha} \mathcal{F} \xrightarrow{h_\lambda} \mathcal{M}$, where $\alpha \in (0, 1]$ is a scaling parameter and $h_\alpha(V) = \alpha V$ is a mapping that scales the vertices $V \in \mathcal{F}^*$ of all obstacles and environment boundary. We use the diffeomorphism h_λ as proposed in [29] such that all the polyhedral obstacles are mapped to a Euclidean sphere contained inside the corresponding polyhedron. Similarly, the outer sphere is chosen sufficiently large that it contains the boundary of the polyhedral workspace. We chose the goal point in \mathcal{M} to be \mathbf{r}_G . To summarize our setting,

$$h_\lambda(\mathbf{r}) = \sigma_{\mathbf{r}_G}(\mathbf{r}, \lambda) \mathbf{r}_G + \sum_{i=0}^M \sigma_i(\mathbf{r}, \lambda) \mathbf{T}_i(\mathbf{r}), \quad (8a)$$

where for all $i = 1, \dots, M$,

$$\sigma_i(\mathbf{r}, \lambda) = \frac{\gamma_k(\mathbf{r}) \bar{\beta}_i(\mathbf{r})}{\gamma_k(\mathbf{r}) \bar{\beta}_i(\mathbf{r}) + \lambda \beta_i(\mathbf{r})}, \quad (8b)$$

$$\sigma_{\mathbf{r}_G}(\mathbf{r}, \lambda) = 1 - \sum_{i=0}^M \sigma_i(\mathbf{r}, \lambda), \quad (8c)$$

$$\mathbf{T}_i(\mathbf{r}) = (1 + \beta_i(\mathbf{r}))^{1/2} r_i \frac{(\mathbf{r} - \mathbf{w}_i)}{\|\mathbf{r} - \mathbf{w}_i\|} + \mathbf{w}_i \quad (8d)$$

$$\mathbf{T}_0(\mathbf{r}) = (1 - \beta_0(\mathbf{r}))^{1/2} r_0 \frac{(\mathbf{r} - \mathbf{w}_0)}{\|\mathbf{r} - \mathbf{w}_0\|} + \mathbf{w}_0, \quad (8e)$$

$$\gamma_k(\mathbf{r}) = \|\mathbf{r} - \mathbf{r}_G\|^{2k}, \quad \bar{\beta}_i(\mathbf{r}) = \prod_{j=0, j \neq i}^M \beta_j(\mathbf{r}), \quad (8f)$$

where \mathbf{w}_i is the center of the sphere corresponding to \mathcal{S}_i and r_i is the radius of the corresponding sphere. In principle, the center \mathbf{w}_i can be anywhere inside the polyhedron \mathcal{S}_i , but the placement of the sphere in \mathcal{M} affects the resulting navigation function. We have found that choosing \mathbf{w}_i close to the geometric center \mathbf{c}_i of \mathcal{S}_i results in a sound navigation field in the context of the geometry in Fig. 1.

1) *Example:* Intuitively, σ_i controls the influence of i th obstacle as a function of space. On increasing the value of λ , σ_i vanishes rapidly away from the obstacle as shown in Fig. 1. Hence, the influence of the obstacle is confined to a small neighborhood about its boundaries. Fig. 2 illustrates the two diffeomorphisms for a polyhedral workspace with $\lambda = 0.02$ and $\alpha = 0.2$ using geometries similar to those that are considered in Sec. VII-VIII. The resulting navigation function is then $\psi = (\hat{\psi} \circ h_\lambda \circ h_\alpha)(\mathbf{r})$, and Fig. 3 depicts such a navigation field for a polyhedral world when $k = 2$.

Finally, the vector field as implemented in (6) is $\rho(\mathbf{r}, \mathbf{r}_G) = -\frac{\nabla \psi}{\|\nabla \psi\|}$. As can be seen in Fig. 3, the navigation function can be flat in some regions, therefore, the gradient must be normalized for operating the UAV in practice.

VI. COMPUTING THE DYNAMIC MARGIN

The ERG proposed in [20], considers (i) geometric constraints from obstacles in the environment and (ii) constraints

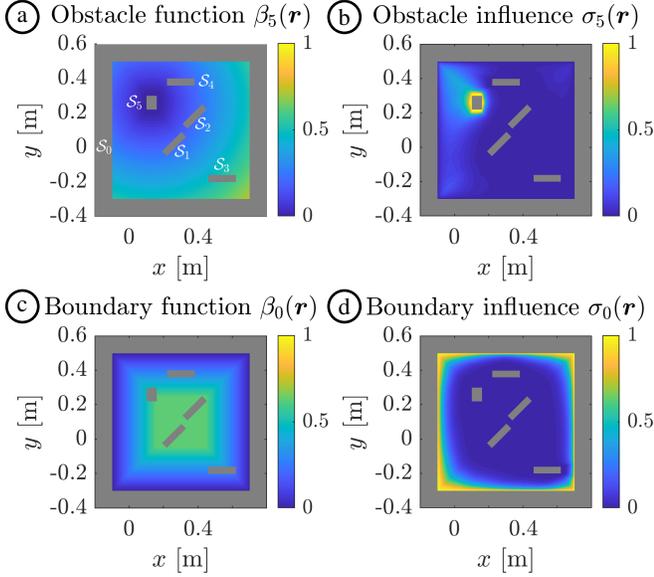


Fig. 1. Obstacle β - and influence σ -functions in the scaled polyhedral world \mathcal{F} at a fixed z with obstacles and boundaries (gray). \textcircled{a} – \textcircled{d} Functions associated with obstacle \mathcal{S}_5 . \textcircled{c} – \textcircled{d} Functions associated with \mathcal{S}_0 . All are two-dimensional sections in a top-down view. See Fig. 5 for the 3D geometry.

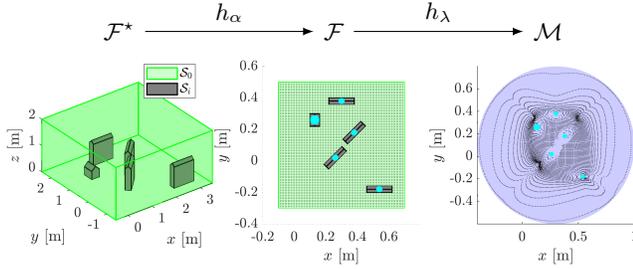


Fig. 2. Diffeomorphisms from the polyhedral world \mathcal{F}^* to the scaled world \mathcal{F} to sphere world \mathcal{M} . A total of 10^4 points (dots) are sampled at $z = 0.4$.

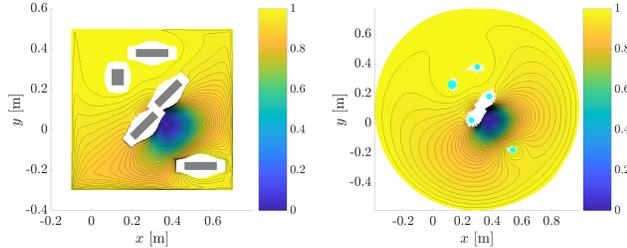


Fig. 3. Level set of navigation field for a polyhedral scaled world of 5 obstacles (left) and the solution in the corresponding sphere world (right). Note that the white regions near the obstacles are inflated due to the contour plot not interpolating properly close to the boundary of the function domain.

on the maximum thrust of the UAV. Both require additional considerations when dealing with convex polyhedral obstacles in \mathbb{R}^3 (Assumption A1) and non-uniform gains (Assumption A2). We also consider (iii) a tilt constraint, limiting the minimum rotation angle of the UAV. The three constraints and their use in computing the dynamic margin of the ERG are discussed in Sec. VI-A–VI-C. We give two preliminary results to clarify the exposition.

Lemma 1 Let $\mathbf{x}_e = (\mathbf{p}^\top - \mathbf{r}^\top; \mathbf{v}^\top)^\top$ and assume that there exists a Lyapunov function $V(\mathbf{p}, \mathbf{v}, \mathbf{r}) = \|\mathbf{x}_e\|_{\mathbf{P}}^2$. Given a

linear constraint $\mathbf{c}_p^\top \mathbf{p} + \mathbf{c}_v^\top \mathbf{v} \leq \mathbf{d}(\mathbf{r})$, if

$$\bar{\Gamma}(\mathbf{r}) = \frac{(\mathbf{c}_p^\top \mathbf{r} - \mathbf{d}(\mathbf{r}))^2}{\begin{bmatrix} \mathbf{c}_p \\ \mathbf{c}_v \end{bmatrix}^\top \mathbf{P}^{-1} \begin{bmatrix} \mathbf{c}_p \\ \mathbf{c}_v \end{bmatrix}} \quad (9)$$

then $V(\mathbf{p}, \mathbf{v}, \mathbf{r}) \leq \bar{\Gamma}(\mathbf{r}) \Rightarrow \mathbf{c}_p^\top \mathbf{p} + \mathbf{c}_v^\top \mathbf{v} \leq \mathbf{d}(\mathbf{r})$.

Proof: This is an adaptation of [19, Proposition 1]. ■

Lemma 1 allows imposing linear constraints on the states and determine the maximal level sets $V \leq \bar{\Gamma}$ in which all trajectories satisfy these constraints at all times, which is key in defining the dynamic margin. However, we also need to consider how to lower-bound the denominator of (9) in the context of the polytopic uncertainty of assumption A2.

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

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix}^\top \bar{\mathbf{K}} \mathbf{P}^{-1} \bar{\mathbf{K}} \begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix} \leq \lambda \quad \forall \mathbf{u} \in \mathcal{S}^2, \quad (10)$$

The smallest value of λ satisfying (10) independent of \mathbf{u} is

$$\min \lambda \quad (11a)$$

$$\begin{bmatrix} \mathbf{P} & \bar{\mathbf{K}}_i \\ \bar{\mathbf{K}}_i & \mathbf{\Lambda} \end{bmatrix} \succeq \mathbf{0}, \quad \forall i = 1, \dots, N, \quad (11b)$$

$$\begin{bmatrix} \lambda_{11} \mathbf{I} & \lambda_{12} \mathbf{I} \\ \star & \lambda_{22} \mathbf{I} \end{bmatrix} = \mathbf{\Lambda}, \quad (11c)$$

$$\lambda_{11} + 2\lambda_{12} + \lambda_{22} = \lambda. \quad (11d)$$

Proof: See appendix. ■

A. Polyhedral Geometries

To compute the dynamic margin for a polyhedral obstacle, we assume a \mathcal{H} -representation $\mathcal{S}_i = \{\mathbf{p} : \mathbf{A}_i^S \mathbf{p} \leq \mathbf{b}_i^S\}$. First the Lyapunov function is projected onto the positional states by taking the Schur complement $\mathbf{Q} = \mathbf{P}_{pp} - \mathbf{P}_{pv} \mathbf{P}_{vv}^{-1} \mathbf{P}_{vp}$. We then introduce a coordinate transform, $\bar{\mathbf{p}} = \mathbf{Q}^{1/2}(\mathbf{p} - \mathbf{r})$, whereby $V(\mathbf{p}, \mathbf{r}, \mathbf{v}) \leq \bar{\Gamma}$ is implied by $\|\bar{\mathbf{p}}\|_2^2 \leq \bar{\Gamma}$. Next, we apply the same linear transformation to the polyhedral obstacles, yielding $\bar{\mathcal{S}}_i = \{\bar{\mathbf{p}} : \bar{\mathbf{A}}_i^S \bar{\mathbf{p}} \leq \bar{\mathbf{b}}_i^S\}$ where

$$\bar{\mathbf{A}}_i^S = \mathbf{A}_i^S \mathbf{Q}^{-1/2}, \quad \bar{\mathbf{b}}_i^S = \mathbf{b}_i^S - \mathbf{A}_i^S \mathbf{r}. \quad (12)$$

In the transformed coordinates, we solve

$$\bar{\mathbf{p}}_i^* = \underset{\bar{\mathbf{p}} \in \bar{\mathcal{S}}_i}{\text{argmin}} \|\bar{\mathbf{p}}\|_2^2, \quad \forall i = 1, \dots, M, \quad (13)$$

and define a level set $\Gamma_i^o = \|\bar{\mathbf{p}}_i^*\|_2^2$ associated with \mathcal{S}_i . If $\mathbf{p}(0) \in \mathbb{F}$ and \mathbf{r} is chosen such that $V \leq \Gamma_i^o$, then $\mathbf{p}(t) \in \mathbb{F} \forall t \geq 0$. Evaluating all Γ_i^o requires solving M QPs on each time step. This can be done independently for each obstacle and the computational cost thus scales linearly in M .

B. Thrust Constraints

The actuation of the UAV depends on the low-level controllers. Here we consider a geometric PD controller implemented as in [37]. When considering point stabilization, the commanded thrust is $f = (mge_3 - \mathbf{K}_p(\mathbf{p} - \mathbf{r}) - \mathbf{K}_d \mathbf{v}) \cdot \mathbf{R}e_3$, where \mathbf{R} is the rotation of the UAV, and the objective is to find an invariant set in which $f \leq f_{\max}$. This can be done along the lines of [20], but becomes complicated if the

VII. SIMULATIONS

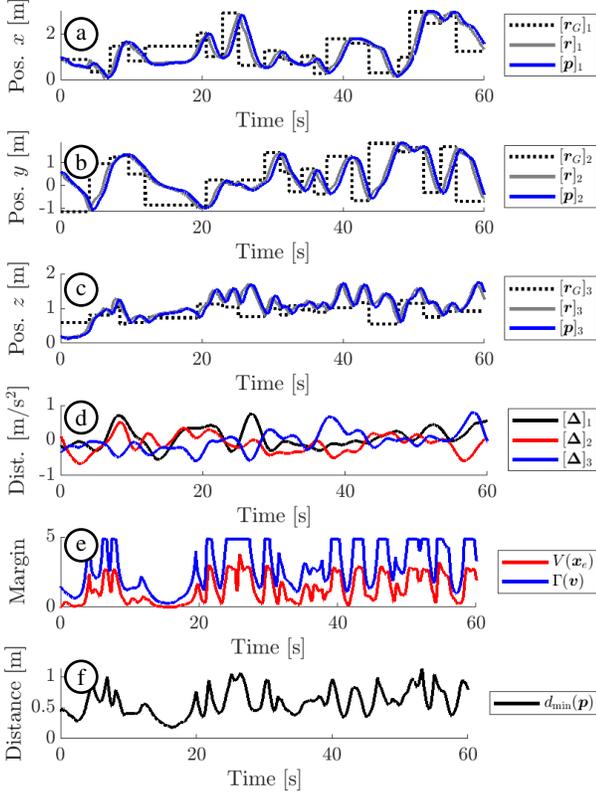


Fig. 4. System response in simulation with disturbances. \odot – \odot Goal positions \mathbf{r}_G (black, dashed), augmented reference \mathbf{r} (gray) and the system response \mathbf{p} (blue) in the three positional dimensions with each time t_k^* at which a new goal is chosen indicated in red. \odot Disturbances acting on the accelerations of the UAV. \odot Maximal safe level set (blue) and Lyapunov function (red), the dynamic margin is the difference of the two and strictly positive. \odot The smallest distance between the UAV and any of the obstacles, never crosses zero, and when approaching the obstacle around $t = 16$ s, the dynamic margin becomes small ensuring safe passage of the UAV.

gains don't have uniform diagonal entries. Following [20] and using Lemma 1 in combination with Lemma 2 we note that if $\Gamma^f = (f_{\max} - mg)^2 m^{-2} (\lambda^*)^{-1}$ then $V \leq \Gamma^f \Rightarrow f(t) < f_{\max}$ for all $t \geq 0$ and any λ^* solving (11).

C. Tilt Constraints

We can bound the tilt of the UAV by the cosine of any rotation angle about a vector in the plane spanned by $\mathbf{e}_1 \mathbf{e}_2$. Let $\theta = \mathbf{e}_3 \cdot \mathbf{R} \mathbf{e}_3 \geq \theta_{\min}$ denote a bound on the cosine angle, then $\Gamma^\theta = (f_{\max} \theta_{\min} - mg)^2 m^{-2} (\lambda^*)^{-1}$ and we have that $V \leq \Gamma^\theta \Rightarrow \theta(t) > \theta_{\min}, \forall t \geq 0$ for any λ^* that solves (11).

D. Dynamic Margin

Finally, the Lyapunov threshold is the most restrictive value, $\Gamma(\mathbf{r}) = \min(\{\Gamma_i^o\}_{i=1}^M \cup \{\Gamma^f, \Gamma^\theta\})$ and the corresponding dynamic margin is then simply computed as $\Gamma(\mathbf{r}) - V(\mathbf{p}, \mathbf{v}, \mathbf{r})$. In the implementation of (12) and (13) we use ProxQP [38] to compute the maximal safe level sets of V with a warm-start of the primal and dual variables to the solution at the previous time step. The computation time for solving the QPs online is minimal ($\approx 3 - 5$ ms) and it increases linearly with the number of obstacles.

To demonstrate the proposed safe navigation function approach, we conduct a simulation using the simplified UAV model in (1) with a polyhedral boundary $\mathcal{S}_0 = [-0.5, 3.5] \times [-1.5, 2.5] \times [0, 2] \subset \mathbb{R}^3$ containing five polyhedral building-like obstacles $\{\mathcal{S}_i\}_{i=1}^5$ that are to be avoided (see Fig. 5.a). The UAV is to navigate to a target $\mathbf{r}_G \in \mathbb{F}$, and upon reaching this goal, a new goal is randomized as $\mathbf{r}_G \sim \mathcal{U}(\mathbb{F})$.

The gain \mathbf{K} is not known, but rather sampled from the set \mathcal{K} containing (3) in its interior, and Gaussian measurement noise is added to the states at each time step, resulting in a measured output $\mathbf{y}_k \sim \mathcal{N}((\mathbf{p}^\top, \mathbf{v}^\top)^\top)$, $\text{diag}(\sigma_p^2 \mathbf{I}_3, \sigma_v^2 \mathbf{I}_3)$, with $\sigma_p = 0.05$ and $\sigma_v = 0.02$. Furthermore, the disturbance acting on the UAV is sampled from a Gaussian process $[\Delta(t)]_i \sim \mathcal{GP}(\mathbf{0}, \mathbf{k}(t, t'))$ with an exponential kernel $\mathbf{k}(t, t') = \exp(0.5|t - t'|^2)$, and each sample is scaled such that $\sup_t \|\Delta(t)\|_2 = 1$. This disturbance is far in excess of what we are likely to see in practice. One of the resulting simulations is shown in Fig. 4 showing the smallest distance between the UAV and the obstacles $d_{\min}(\mathbf{p}) = \min_i \min_{\bar{\mathbf{p}} \in \mathcal{S}_i} \|\bar{\mathbf{p}} - \mathbf{p}\|_2$ in Fig. 4.f.

Discussion: In a large number of simulations, the UAV remained safe at all times with a positive dynamic margin ($V \leq \Gamma$). But in some simulations, the navigation function pushes very close to an obstacle, leading to a small dynamic margin and very slow movement of the UAV (see Fig. 4.e around 18s). This can be modified by tuning the navigation function and further inflating the obstacles. Still, trajectories that seem desirable in the sphere world, \mathcal{M} , may not map to similarly desirable trajectories in the polyhedral world, \mathcal{F} .

VIII. EXPERIMENTS

The ERG was implemented as a ROS node and interfaced with the CrazySwarm driver [39] to enable real-time operation of the Crazyflie 2.1 UAV in Fig. 5.c. To account for uncertainty in the system identification, the set \mathcal{K} was defined by $N = 10$ controllers containing (3) in its interior. Computing the Lyapunov function as per Sec. III, yields

$$\mathbf{P} = \begin{bmatrix} 7.05 & 0 & 0 & 0.59 & 0 & 0 \\ 0 & 6.64 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 11.60 & 0 & 0 & 0.79 \\ 0.59 & 0 & 0 & 1.07 & 0 & 0 \\ 0 & 0.56 & 0 & 0 & 1.06 & 0 \\ 0 & 0 & 0.79 & 0 & 0 & 1.19 \end{bmatrix}, \quad (14)$$

and a maximal safe level set of $\min(\{\Gamma^f, \Gamma^\theta\}) = 3.827$ with respect to the thrust and tilt constraints, when $f_{\max} = 2mg$ [N] and $\theta_{\min} = 0.95$ [rad]. Furthermore, bounding the additive input disturbance at $\|\Delta\|_2 \leq 1$ [m/s²] as motivated by the system identification, we get a positional margin of $\delta = 0.19$ [m] by which the obstacles are inflated. The UAV is set to track a sequence of randomly generated goals, where if $\|\mathbf{p} - \mathbf{r}_G\| < 0.1$, a new goal is sampled in free-space $\mathbf{r}_G \sim \mathcal{U}(\mathbb{F})$. The obstacles used in Sec VII were created physically, and the response from a flight is shown in Fig. 5.

Discussion: Throughout the flight, it is verified the dynamic margin is strictly positive, just as in the simulations, implying

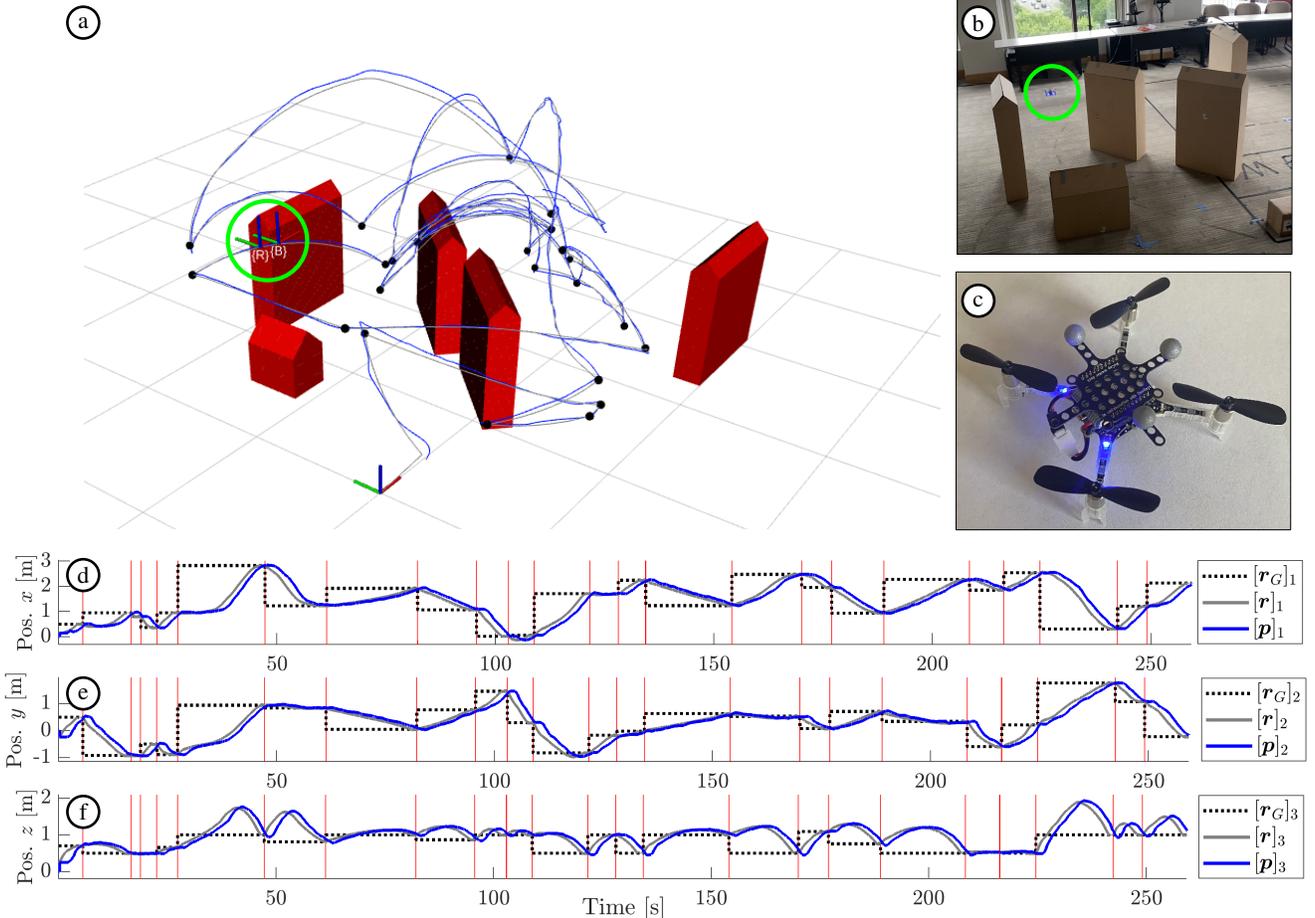


Fig. 5. System response from an experiment with the Crazyflie 2.1 UAV using the proposed ERG with data logged using a uSD card at 100Hz and visualized in RVIZ. © The world consists of known polyhedral obstacles (red) and the UAV (in the green circle) is navigating through a sequence of 23 randomized goal points $\{\mathbf{r}_G(t_k^s) \in \mathbb{F}\}_{k=1}^{23}$ (black). The resulting augmented reference trajectory $\mathbf{r}(t)$ and system response $\mathbf{p}(t)$ are shown in gray and blue, respectively. The coordinate frames of the UAV $\{\mathcal{B}\}$ and the reference $\{\mathcal{R}\}$ are depicted at a single point in time. © Picture of the experiment setup with the UAV (green circle). © Picture of the Crazyflie 2.1 used in the experiments. © – © Goal positions \mathbf{r}_G (black, dashed), augmented reference \mathbf{r} (gray) and the system response \mathbf{p} (blue) in the three positional dimensions with each time t_k^s at which a new goal is chosen indicated in red.

that the system satisfies the constraints at all times. We note that ERG gives rise to transients in the z -direction at certain times, notably about $t \in [40, 60]$ and $t \in [220, 230]$. This behavior is also observed in the simulations and is an artifact of how the navigation function is constructed. A drawback of the method in [28], [29] is that it lacks optimality guarantees, and as such, it may be interesting to explore the proposed ERG mechanism in the context of other trajectories such as potentially un-safe minimum snap trajectories [40], [41]. This is outside the scope of the paper and left as future work.

IX. CONCLUSION

In this paper, we investigate the explicit reference governor (ERG) proposed in [19]–[21] combined with classical navigation functions based on sphere-world solutions [25]. To make these methods viable in practice, we considered the case of polyhedral obstacles and structured uncertainty in the non-uniform diagonal gains governing the dynamics of the UAV, as motivated by a system identification. Both of these developments were necessary to operate the Crazyflie 2.1. The resulting method provides certifiably safe operations of a UAV in non-convex free space with significant input disturbances. Furthermore, it is more computationally

lightweight than an MPC solution (the optimization problems are smaller), comes with guarantees of safety, and has a small memory footprint, unlike the ISMPs.

APPENDIX

Let $\bar{\mathbf{u}} = (\mathbf{u}^\top, \mathbf{u}^\top)^\top$. As $\bar{\mathbf{K}}$ and \mathbf{P} are positive definite and symmetric, $\bar{\mathbf{K}}\mathbf{P}^{-1}\bar{\mathbf{K}} \succ \mathbf{0}$, and there exist a bound $\bar{\mathbf{u}}^\top \bar{\mathbf{K}}\mathbf{P}^{-1}\bar{\mathbf{K}}\bar{\mathbf{u}} \leq \bar{\mathbf{u}}^\top \Lambda \bar{\mathbf{u}} = \lambda$. Furthermore, the largest class of matrices Λ that yields a λ independent of $\mathbf{u} \in \mathbb{S}^2$ is

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix}^\top \underbrace{\begin{bmatrix} \lambda_{11}\mathbf{I}_3 & \lambda_{12}\mathbf{I}_3 \\ * & \lambda_{22}\mathbf{I}_3 \end{bmatrix}}_{\triangleq \Lambda} \begin{bmatrix} \mathbf{u} \\ \mathbf{u} \end{bmatrix} = \|\mathbf{u}\|_2^2 (\lambda_{11} + 2\lambda_{12} + \lambda_{22}) = \lambda. \quad (15)$$

As such, we seek a matrix Λ , which is positive definite as $\mathbf{P} \succ \mathbf{0}$ and $\bar{\mathbf{K}} \succ \mathbf{0}$. By Schur complements and convexity,

$$\bar{\mathbf{K}}\mathbf{P}^{-1}\bar{\mathbf{K}} \preceq \Lambda \Leftrightarrow \Lambda - \bar{\mathbf{K}}\mathbf{P}^{-1}\bar{\mathbf{K}} \succeq \mathbf{0} \quad (16a)$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{P} & \bar{\mathbf{K}} \\ \bar{\mathbf{K}} & \Lambda \end{bmatrix} \succeq \mathbf{0} \quad (16b)$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{P} & \bar{\mathbf{K}}_i \\ \bar{\mathbf{K}}_i & \Lambda \end{bmatrix} \succeq \mathbf{0}, \quad \forall i = 1, \dots, N. \quad (16c)$$

As such, the smallest gain λ can be found by its minimization over (15) and (16), yielding the LMIs in Lemma 2.

REFERENCES

- [1] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5-6, pp. 1327–1349, 2021.
- [2] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382–2387, 2008.
- [3] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
- [4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.
- [5] A. Manjunath and Q. Nguyen, "Safe and robust motion planning for dynamic robotics via control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2122–2128.
- [6] V. Mundheda, H. Kandath *et al.*, "Control barrier function-based predictive control for close proximity operation of UAVs inside a tunnel," *arXiv preprint arXiv:2303.16177*, 2023.
- [7] J. Breeden and D. Panagou, "Safety-critical control for systems with impulsive actuators and dwell time constraints," *IEEE Control Systems Letters*, 2023.
- [8] J. Zeng, "Challenges of control barrier functions: Optimization, control, planning, and navigation," Ph.D. dissertation, University of California, Berkeley, 2022.
- [9] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [10] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [11] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 5986–5991.
- [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] T. Anevclavis, Z. Liu, N. Ozay, and P. Tabuada, "Controlled invariant sets: implicit closed-form representations and applications," *arXiv preprint arXiv:2107.08566*, 2021.
- [15] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.
- [16] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *2014 American Control Conference*. IEEE, 2014, pp. 226–241.
- [17] E. G. Gilbert and I. V. Kolmanovsky, "Set-point control of nonlinear systems with state and control constraints: A Lyapunov-function, reference-governor approach," in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*, vol. 3. IEEE, 1999, pp. 2507–2512.
- [18] E. Gilbert and I. Kolmanovsky, "Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor," *Automatica*, vol. 38, no. 12, pp. 2063–2073, 2002.
- [19] M. M. Nicotra and E. Garone, "Explicit reference governor for continuous time nonlinear systems subject to convex constraints," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 4561–4566.
- [20] M. M. Nicotra, R. Naldi, and E. Garone, "A robust explicit reference governor for constrained control of unmanned aerial vehicles," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 6284–6289.
- [21] M. M. Nicotra and E. Garone, "The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems," *IEEE Control Systems Magazine*, vol. 38, no. 4, pp. 89–107, 2018.
- [22] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, vol. 1. IEEE, 2000, pp. 256–263.
- [23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
- [24] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in applied mathematics*, vol. 11, no. 4, pp. 412–442, 1990.
- [25] E. Rimon, *Exact robot navigation using artificial potential functions*. Yale University, 1990.
- [26] S. G. Loizou, "Navigation functions in topologically complex 3D workspaces," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 4861–4866.
- [27] L. Fan and J. Liu, "Mobile robot navigation in complex polygonal workspaces using conformal navigation transformations," *arXiv preprint arXiv:2208.09635*, 2022.
- [28] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [29] E. Rimon and D. E. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Transactions of the American Mathematical Society*, vol. 327, no. 1, pp. 71–116, 1991.
- [30] L. Ljung and T. Glad, *Modeling of dynamic systems*. Prentice-Hall, Inc., 1994.
- [31] D. S. Bernstein and S. P. Bhat, "Lyapunov Stability, Semistability, and Asymptotic Stability of Matrix Second-Order Systems," *Journal of Mechanical Design*, vol. 117, no. B, pp. 145–153, 06 1995. [Online]. Available: <https://doi.org/10.1115/1.2836448>
- [32] H. K. Khalil, "Nonlinear systems third edition," *Patience Hall*, vol. 115, 2002.
- [33] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014.
- [34] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [35] O. Zenkin, "Analytical description of geometrical shapes," *Cybernetics*, vol. 6, no. 4, pp. 481–489, 1970.
- [36] C. Li, "Star-world navigation functions for convergence to a time-varying destination manifold," Ph.D. dissertation, University of Delaware, 2017.
- [37] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.
- [38] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-qp: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.
- [39] J. A. Preiss*, W. Hönig*, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304, software available at <https://github.com/USC-ACTLab/crazyswarm>. [Online]. Available: <https://doi.org/10.1109/ICRA.2017.7989376>
- [40] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: The 16th International Symposium ISRR*. Springer, 2016, pp. 649–666.
- [41] M. Greiff, A. Vinod, S. Nabi, and S. Di Cairano, "Quadrotor motion planning in stochastic wind fields," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 4619–4625.