

# Safe Navigation under Uncertain Obstacle Dynamics using Control Barrier Functions and Constrained Convex Generators

Hugo Matias, Daniel Silvestre

arXiv:2601.07715v1 [eess.SY] 12 Jan 2026

**Abstract**—This paper presents a sampled-data framework for the safe navigation of controlled agents in environments cluttered with obstacles governed by uncertain linear dynamics. Collision-free motion is achieved by combining Control Barrier Function (CBF)-based safety filtering with set-valued state estimation using Constrained Convex Generators (CCGs). At each sampling time, a CCG estimate of each obstacle is obtained using a finite-horizon guaranteed estimation scheme and propagated over the sampling interval to obtain a CCG-valued flow that describes the estimated obstacle evolution. However, since CCGs are defined indirectly—as an affine transformation of a generator set subject to equality constraints, rather than as a sublevel set of a scalar function—converting the estimated obstacle flows into CBFs is a nontrivial task. One of the main contributions of this paper is a procedure to perform this conversion, ultimately yielding a CBF via a convex optimization problem whose validity is established by the Implicit Function Theorem. The resulting obstacle-specific CBFs are then merged into a single CBF that is used to design a safe controller through the standard Quadratic Program (QP)-based approach. Since CCGs support Minkowski sums, the proposed framework also naturally handles rigid-body agents and generalizes existing CBF-based rigid-body navigation designs to arbitrary agent and obstacle geometries. While the main contribution is general, the paper primarily focuses on agents with first-order control-affine dynamics and second-order strict-feedback dynamics. Simulation examples demonstrate the effectiveness of the proposed method.

**Index Terms**—Nonlinear Systems, Constrained Control, State Estimation, Linear System Observers, Control Barrier Functions, Constrained Convex Generators

## I. INTRODUCTION

THE collision-free motion of controlled agents in dynamic and uncertain environments is a fundamental requirement in many real-world applications. In cooperative scenarios, such as fleets of robots sharing a workspace or teams of autonomous

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through the LARSyS FCT funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, and 10.54499/UIDB/50009/2020), and through the COPELABS, Lusófona University project UIDB/04111/2020.

Hugo Matias is with the Center of Technology and Systems (UNINOVACTS), NOVA School of Science and Technology (NOVA-FCT), 2829-516 Caparica, Portugal, and also with the Institute for Systems and Robotics (ISR-Lisbon), Instituto Superior Técnico (IST), 1049-001 Lisbon, Portugal (email: h.matias@campus.fct.unl.pt).

Daniel Silvestre is with the Center of Technology and Systems (UNINOVACTS), NOVA School of Science and Technology (NOVA-FCT), 2829-516 Caparica, Portugal, with the Institute for Systems and Robotics (ISR-Lisbon), Instituto Superior Técnico (IST), 1049-001 Lisbon, Portugal, and also with the COPELABS, Lusófona University, 1749-024 Lisbon, Portugal (e-mail: dsilvestre@fct.unl.pt).

vehicles coordinating maneuvers, agents may exchange information, yet remain subject to disturbances, measurement noise, and communication delays [1], [2]. Conversely, in noncooperative or adversarial settings, where the environment is cluttered with uncontrolled objects and independent autonomous agents, these entities follow their trajectories without disclosing state information or intentions to the navigating agent [3]. Across these situations, safe navigation hinges on accurately capturing uncertainty, ideally while avoiding excessive conservatism.

In such contexts, guaranteed (set-valued) state estimation is a powerful alternative to stochastic filtering. Instead of relying on probabilistic assumptions, set-valued estimation algorithms compute sets that are guaranteed to contain the true state of a dynamical system, given bounds on the system inputs and the measurement noise [4]. As highlighted in recent surveys, this framework has strong theoretical appeal and has sparked a lot of interest from the research community [5], [6].

A key consideration when designing guaranteed state estimation algorithms is the set representation. For linear systems, numerous methods have been developed based on different set representations, evolving from intervals [7], [8] and ellipsoids [9], [10] to more expressive structures such as zonotopes [11], [12] and Constrained Zonotopes (CZs) [13], [14], which have a reduced wrapping effect compared to intervals and ellipsoids. For nonlinear systems, these techniques can still be employed, provided that the dynamics are approximated by a linear model to allow set propagation, as performed in [15], [16], [17], [18], and [19], for each of the mentioned set classes.

Most recently, Constrained Convex Generators (CCGs) have emerged as a unifying set representation capable of capturing highly general convex sets, subsuming all previously discussed set classes and greatly reducing the need for approximations. CCGs are sets described through an affine transformation of a generator set in a (typically) higher-dimensional space, subject to linear equality constraints, enabling expressive shapes while supporting key operations such as affine mappings, Minkowski sums, and generalized intersections. Such features make CCGs particularly well suited for set-based estimation, enabling tight enclosures even when the system dynamics and measurement model induce mixtures of polytopic and ellipsoidal geometry. Consequently, CCGs currently represent the state of the art in convex set-valued estimation [20], [21], [22], [23], [24].

It is important to note, however, that a central challenge in guaranteed state estimation is the growth of the underlying data structures over time, requiring the use of order reduction methods to maintain a fixed computational load. Several order

reduction techniques have been developed for specific set representations, including zonotopes [25], ellipsotopes [26], CZs [13], and CCGs [27]. More recently, [28] introduces a CCG finite-horizon scheme that refines an estimate computed by an ellipsoidal observer using a limited history of measurements, thereby removing the need for order reduction methods.

Combining Model Predictive Control (MPC) with CCG-based state estimation for collision-free motion in the presence of obstacles with uncertain dynamics is relatively straightforward, as safety constraints concerning estimated CCG obstacle sets can be directly derived from the definition of a CCG. Since a CCG is expressed as an affine map of a generator set subject to linear equality constraints, these conditions can be encoded into the MPC formulation to enforce obstacle avoidance, with the generator variable of the CCG included as an optimization variable. However, this approach leads to an MPC formulation with nonconvex constraints, an issue that becomes even more severe when the agent dynamics are nonlinear [29]. As a result, despite the predictive advantages of MPC, it adds a significant computational burden for real-time implementation, even when resorting to convexification techniques [30], [31], [32].

Over the last decade, Control Barrier Functions (CBFs) have emerged as a powerful tool for designing safe controllers for nonlinear systems [33], [34], [35]. One of the primary utilities of CBFs is their ability to serve as a safety filter for a nominal controller, which may not have been designed to ensure safety. Such safety filters are typically instantiated through Quadratic Programs (QPs), which can be efficiently executed in real time, to minimize the deviation from the nominal controller while satisfying a Lyapunov-like condition that guarantees forward invariance of a designated safe set [36], [37], [38]. However, in contrast to MPC, combining CBF-based safety filtering with CCG-based state estimation for safe navigation in the presence of obstacles with uncertain dynamics is a nontrivial task. Since CCGs describe sets indirectly—as an affine map of a generator set subject to equality constraints, rather than as a sublevel set of a scalar function—estimated CCG obstacle sets cannot be directly translated into CBFs. To the best of our knowledge, no existing work addresses the problem of integrating CBF-based control with guaranteed state estimation based on CCGs or related set representations such as CZs.

For deterministic environments, however, some CBF-based methods have been proposed for safe navigation under different agent and obstacle geometries. The main challenge tackled by this line of research is that ensuring safety for a rigid-body agent requires maintaining the entire agent set in a safe region, while classical CBF-based techniques only ensure safety for a single point. Examples include CBF-based methods for robotic arms avoiding point obstacles [39], circular agents navigating around ellipses, cardioids, diamonds, and squares via discrete barrier states and differential dynamic programming [40], and distributed multi-agent navigation based on circular agent and obstacle geometries [41]. Further examples include CBF-based techniques for manipulators interacting with humans modeled as capsules [42], and manipulators moving under more general obstacle geometries using the signed distance function [43].

Additionally, a few studies have explored CBF-based strategies for safe navigation under polytopic geometries. For exam-

ple, [44] navigates polygonal agents in elliptical environments based on the polygon-ellipse distance, [45] addresses polytope-polytope avoidance via discrete-time CBF constraints, and [46] designs polygonal cone CBFs based on the vertices of polygon obstacles. More recently, [47] proposes an optimization-based CBF that directly considers the exact signed distance function between a polytopic agent and polytopic obstacles, while [48] presents an optimization-free alternative for polytope-polytope navigation based on smooth approximations of the maximum and minimum functions.

Despite recent progress, the aforementioned approaches are still quite specific, relying on constructions that do not readily extend to richer set classes, such as mixtures of polytopes and ellipsoids or sets defined by general  $\ell_p$ -norms. In this context, CCGs—beyond their original role in state estimation—provide a natural and unifying set representation for rigid-body navigation. Since CCGs are closed under Minkowski sums, rigid-body agents can be equivalently treated as single points while the obstacles are enlarged with the agent geometry, enabling a systematic treatment of general agent and obstacle shapes once a mechanism for translating CCGs into CBFs is developed.

#### A. Main Contributions and Organization

The main goal of this paper is to introduce the theoretical framework for integrating CBF-based control with guaranteed state estimation based on CCGs, providing the foundations for further developments in this direction. To this end, we address the safe navigation of a controlled agent in an environment populated with obstacles governed by uncertain linear dynamics, with measurements obtained at discrete sampling instants.

At each sampling instant, a CCG estimate of each obstacle is computed using a finite-horizon guaranteed estimation scheme that generalizes the approach from [28]. Each estimate is then propagated over the sampling interval to obtain a CCG-valued flow that describes the estimated obstacle evolution, being the basis for safety enforcement. The central challenge then lies in translating these CCG-valued flows into (time-varying) CBFs. The main contribution of the paper is a procedure that enables this conversion, which ultimately produces a CBF via a convex optimization problem whose validity is established through the Implicit Function Theorem under mild regularity conditions. The resulting obstacle-specific CBFs are then combined into a single CBF using a smooth approximation of the minimum function, and the overall CBF is employed to synthesize a safe controller via the standard QP-based approach.

Since CCGs support Minkowski sums, the proposed framework naturally handles rigid-body agents, generalizing existing CBF-based designs for rigid-body navigation to arbitrary agent and obstacle geometries. Also, while the main contribution is general, the paper initially focuses on agents with first-order control-affine dynamics and then extends the method to agents with second-order strict-feedback dynamics via backstepping.

The remainder of the paper is structured as follows. Section II provides essential mathematical background, and Section III formulates the safe navigation problem addressed in the paper. Sections IV, V, and VI describe the proposed approach, with simulation results in Section VII. Finally, Section VIII draws conclusions and outlines future directions.

## B. Notation and General Definitions

$\mathbb{N}$  is the nonnegative integers set.  $\mathbb{R}$ ,  $\mathbb{R}_{\geq 0}$ , and  $\mathbb{R}_{> 0}$  are the sets of real, nonnegative, and positive numbers, respectively.  $\mathbb{R}^n$  is the  $n$ -dimensional euclidean space, and  $\mathbb{S}^{n-1}$  is the unit sphere in  $\mathbb{R}^n$ .  $\mathbb{R}^{n \times m}$  is the set of  $n \times m$  real matrices,  $\mathbb{R}_{> 0}^{n \times n}$  is the set of positive-definite matrices of size  $n$ , and  $\text{SO}(n)$  is the special orthogonal group in  $\mathbb{R}^n$ . For a given set  $\mathcal{S} \subseteq \mathbb{R}^n$ ,  $\text{int}(\mathcal{S})$  and  $\partial\mathcal{S}$  are the interior and boundary of  $\mathcal{S}$ , respectively. The  $p$ -norm of a vector  $\mathbf{x} \in \mathbb{R}^n$  is denoted  $\|\mathbf{x}\|_p$  ( $\|\mathbf{x}\| = \|\mathbf{x}\|_2$ ), and for two vectors  $\mathbf{x}_1 \in \mathbb{R}^{n_1}$ ,  $\mathbf{x}_2 \in \mathbb{R}^{n_2}$ , we often use the notation  $(\mathbf{x}_1, \mathbf{x}_2) = [\mathbf{x}_1^\top \mathbf{x}_2^\top]^\top \in \mathbb{R}^{n_1+n_2}$ . For a differentiable function  $h : \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}$  and  $\mathbf{G} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ , we consider the Lie-derivative notation  $L_{\mathbf{G}} h(\mathbf{x}, t) = \frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}, t) \mathbf{G}(\mathbf{x})$ , and  $\dot{\mathbf{x}}$  is the time derivative of  $\mathbf{x} \in \mathbb{R}^n$ . Additionally,  $\mathbf{0}_{n \times m}$  is the  $n \times m$  matrix of zeros, and  $\mathbf{I}_n$  is the identity matrix of size  $n$  (dimensions are often omitted when clear from context). The Cartesian product is represented by  $\times$ , the Minkowski sum by  $\oplus$ , and the generalized set intersection by  $\cap_{\mathbf{R}}$ . Finally, given the matrices  $\mathbf{A}_1, \dots, \mathbf{A}_M$ ,  $\text{diag}(\mathbf{A}_1, \dots, \mathbf{A}_M)$  yields a block diagonal matrix whose diagonal blocks are  $\mathbf{A}_1, \dots, \mathbf{A}_M$ .

**Definition 1** (Extended Class- $\mathcal{K}/\mathcal{K}_\infty$  Function). A continuous function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  is said to be an extended class- $\mathcal{K}$  function if it is strictly increasing with  $\alpha(0) = 0$ , and it is an extended class- $\mathcal{K}_\infty$  function if, additionally,  $\lim_{s \rightarrow \pm\infty} \alpha(s) = \pm\infty$ .

**Definition 2** (Graph of a Set-Valued Flow). Consider a set-valued flow  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$ . The set  $\mathcal{G}(\mathcal{D})$ , defined as

$$\mathcal{G}(\mathcal{D}) = \{(\mathbf{x}, t) \in \mathbb{R}^n \times [t_0, t_f] : \mathbf{x} \in \mathcal{D}(t)\}, \quad (1)$$

is said to be the graph of  $\mathcal{D}$ .

## II. MATHEMATICAL BACKGROUND

This section provides essential mathematical background on CBFs. Since we will deal with dynamic obstacle avoidance, we present the concepts in a time-varying fashion, as the standard static formulations naturally arise as particular cases.

Consider a nonlinear control-affine system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}, \quad (2)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the system state,  $\mathbf{u} \in \mathbb{R}^m$  is the control input, and the fields  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $\mathbf{G} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$  are locally Lipschitz. Let now  $\mathbf{k} : \mathcal{G}(\mathcal{D}) \rightarrow \mathbb{R}^m$  be a feedback controller, defined over the graph of a set-valued flow  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$ . Applying  $\mathbf{k}$  to (2) yields the time-varying closed-loop system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{k}(\mathbf{x}, t). \quad (3)$$

As the functions  $\mathbf{f}$  and  $\mathbf{G}$  are locally Lipschitz, if the controller  $\mathbf{k}$  is locally Lipschitz in  $\mathbf{x}$  and continuous in  $t$ , then, for every initial condition  $\mathbf{x}_0 \in \mathcal{D}(t_0)$ , there exists a unique continuously differentiable solution  $\varphi : I(\mathbf{x}_0) \rightarrow \mathbb{R}^n$  satisfying

$$\begin{aligned} \dot{\varphi}(t) &= \mathbf{f}(\varphi(t)) + \mathbf{G}(\varphi(t))\mathbf{k}(\varphi(t), t), \\ \varphi(t_0) &= \mathbf{x}_0, \end{aligned} \quad (4)$$

for all  $t \in I(\mathbf{x}_0)$ , where  $I(\mathbf{x}_0) \subseteq [t_0, t_f]$  denotes the maximal interval of existence for the solution [49]. In the remainder of this paper, we assume that  $I(\mathbf{x}_0) = [t_0, t_f]$  for convenience. Next, we revisit the standard definition of forward invariance for sets and extend it to set-valued flows.

**Definition 3** (Forward Invariance (Set)). A set  $\mathcal{C} \subset \mathbb{R}^n$  is said to be forward invariant for the system (3) if, for every initial condition  $\mathbf{x}_0 \in \mathcal{C}$ , we have  $\varphi(t) \in \mathcal{C}$  for all  $t \in [t_0, t_f]$ .

**Definition 4** (Forward Invariance (Set-Valued Flow)). A set-valued flow  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$  is said to be forward invariant for the system (3) if, for every initial condition  $\mathbf{x}_0 \in \mathcal{C}(t_0)$ , we have  $\varphi(t) \in \mathcal{C}(t)$  for all  $t \in [t_0, t_f]$ .

## A. Control Barrier Functions

We intend to ensure that, at every time instant, the solution of the closed-loop system (3) lies within a safe set  $\mathcal{C}(t) \subset \mathbb{R}^n$ , which amounts to ensuring forward invariance of a set-valued flow  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$  for system (3). Particularly, we consider a set-valued flow  $\mathcal{C}$  defined for all  $t \in [t_0, t_f]$  as

$$\mathcal{C}(t) = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, t) \geq 0\}, \quad (5)$$

where  $h : \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}$  is a continuously differentiable function with  $\frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}, t) \neq \mathbf{0}$  when  $h(\mathbf{x}, t) = 0$ . This regularity condition implies that, for all  $t \in [t_0, t_f]$ ,

$$\begin{aligned} \partial\mathcal{C}(t) &= \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, t) = 0\}, \\ \text{int}(\mathcal{C}(t)) &= \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}, t) > 0\}. \end{aligned} \quad (6)$$

If  $h$  has the properties of a CBF, then it can be used to design safe controllers for the system (2).

**Definition 5** (CBF [33]). Let  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$  be a set-valued flow defined by (5), for a continuously differentiable function  $h : \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}$  with  $\frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}, t) \neq \mathbf{0}$  when  $h(\mathbf{x}, t) = 0$ . The function  $h$  is a (zeroing) CBF for the system (2) if there exists a set-valued flow  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$  with  $\mathcal{C}(t) \subseteq \mathcal{D}(t)$  for all  $t \in [t_0, t_f]$  and an extended class- $\mathcal{K}_\infty$  function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  such that, for all  $(\mathbf{x}, t) \in \mathcal{G}(\mathcal{D})$ ,

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \dot{h}(\mathbf{x}, t, \mathbf{u}) > -\alpha(h(\mathbf{x}, t)), \quad (7)$$

where the function  $\dot{h}$  is defined as

$$\dot{h}(\mathbf{x}, t, \mathbf{u}) = L_{\mathbf{f}} h(\mathbf{x}, t) + L_{\mathbf{G}} h(\mathbf{x}, t) \mathbf{u} + \frac{\partial h(\mathbf{x}, t)}{\partial t}. \quad (8)$$

Such a definition means that a CBF is allowed to decrease in the interior of the safe set but not on its boundary. Given a CBF  $h$  for (2) and a corresponding extended class- $\mathcal{K}_\infty$  function  $\alpha$ , we define the pointwise set of controls

$$K_{\text{CBF}}(\mathbf{x}, t) = \left\{ \mathbf{u} \in \mathbb{R}^m : \dot{h}(\mathbf{x}, t, \mathbf{u}) \geq -\alpha(h(\mathbf{x}, t)) \right\}. \quad (9)$$

This yields the following main result concerning CBFs.

**Theorem 1** (Safe Controller [33]). Let  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$  be a set-valued flow defined by (5), for a continuously differentiable function  $h : \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}$  such that  $\frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}, t) \neq \mathbf{0}$  when  $h(\mathbf{x}, t) = 0$ . If the function  $h$  is a CBF for the system (2) on a set-valued flow  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$ , then the set  $K_{\text{CBF}}(\mathbf{x}, t)$  is nonempty for all  $(\mathbf{x}, t) \in \mathcal{G}(\mathcal{D})$ , and every feedback controller  $\mathbf{k} : \mathcal{G}(\mathcal{D}) \rightarrow \mathbb{R}^m$  that is locally Lipschitz in  $\mathbf{x}$ , continuous in  $t$ , and with  $\mathbf{k}(\mathbf{x}, t) \in K_{\text{CBF}}(\mathbf{x}, t)$  for all  $(\mathbf{x}, t) \in \mathcal{G}(\mathcal{D})$  renders  $\mathcal{C}$  forward invariant for the resulting closed-loop system.

**Remark 1.** The strictness of the inequality (7) enables proving that optimization-based controllers relying on CBFs are locally Lipschitz continuous in  $\mathbf{x}$  [50], [36].

### B. Safety Filters

One of the major utilities of CBFs is their ability to serve as a safety filter for a nominal controller  $\mathbf{k}_d : \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}^m$ . A safety filter is a controller that modifies  $\mathbf{k}_d$ , preferably in a minimally invasive fashion, such that the resulting closed-loop system is safe. Given a CBF  $h : \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}$  for (2) on  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^n$ , the typical approach for designing a safety filter  $\mathbf{k} : \mathcal{G}(\mathcal{D}) \rightarrow \mathbb{R}^m$  is through the following QP:

$$\mathbf{k}(\mathbf{x}, t) = \arg \min_{\mathbf{u} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{u} - \mathbf{k}_d(\mathbf{x}, t)\|^2 \quad (10)$$

$$\text{subject to } \dot{h}(\mathbf{x}, t, \mathbf{u}) \geq -\alpha(h(\mathbf{x}, t)),$$

where  $\alpha$  is an extended class- $\mathcal{K}_\infty$  function associated with the CBF. As per the Karush-Kuhn-Tucker (KKT) conditions, the controller defined in (10) can be expressed in closed form as

$$\mathbf{k}(\mathbf{x}, t) = \mathbf{k}_d(\mathbf{x}, t) + \mu(\mathbf{x}, t) L_{\mathbf{G}} h(\mathbf{x}, t)^\top, \quad (11)$$

where  $\mu(\mathbf{x}, t)$  is the KKT multiplier associated with the CBF constraint, defined for all  $(\mathbf{x}, t) \in \mathcal{G}(\mathcal{D})$  as

$$\mu(\mathbf{x}, t) = \begin{cases} \bar{\mu}(\mathbf{x}, t), & \text{if } \dot{h}(\mathbf{x}, t, \mathbf{k}_d(\mathbf{x}, t)) \leq -\alpha(h(\mathbf{x}, t)), \\ 0, & \text{if } \dot{h}(\mathbf{x}, t, \mathbf{k}_d(\mathbf{x}, t)) > -\alpha(h(\mathbf{x}, t)), \end{cases} \quad (12)$$

where the expression corresponding to the first case is

$$\bar{\mu}(\mathbf{x}, t) = -\frac{\dot{h}(\mathbf{x}, t, \mathbf{k}_d(\mathbf{x}, t)) + \alpha(h(\mathbf{x}, t))}{\|L_{\mathbf{G}} h(\mathbf{x}, t)\|^2}. \quad (13)$$

Also, the controller defined in (10) is locally Lipschitz in  $\mathbf{x}$  and continuous in  $t$ , provided that the CBF gradient field is locally Lipschitz in  $\mathbf{x}$ , the function  $\alpha$  is locally Lipschitz, and the nominal controller is locally Lipschitz in  $\mathbf{x}$  and continuous in  $t$  [50], [36]. When input bounds must be taken into account, an enhanced formulation is required, such as the optimal-decay QP introduced in [51] and further studied in [52].

### III. PROBLEM FORMULATION

We consider a rigid-body agent whose configuration at any time instant is described by the set

$$\mathcal{P}(\mathbf{p}) = \mathbf{p} + \bar{\mathcal{P}}, \quad (14)$$

where  $\mathbf{p} \in \mathbb{R}^p$  is the position of the agent, matching its center, and  $\bar{\mathcal{P}} \subset \mathbb{R}^p$  is a compact convex set with nonempty interior that defines the shape of the agent relative to its center. Also, we consider two types of dynamics for the agent, with different relative degrees. In the first scenario, the agent dynamics are described by a first-order control-affine model of the form

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}) + \mathbf{G}(\mathbf{p})\mathbf{z}, \quad (15)$$

where the state matches the position of the agent,  $\mathbf{z} \in \mathbb{R}^z$  is the control input, the functions  $\mathbf{f} : \mathbb{R}^p \rightarrow \mathbb{R}^p$  and  $\mathbf{G} : \mathbb{R}^p \rightarrow \mathbb{R}^{p \times z}$  are locally Lipschitz, and the matrix  $\mathbf{G}(\mathbf{p})$  has full row rank for all  $\mathbf{p} \in \mathbb{R}^p$ . In the second scenario, the agent dynamics are extended to a second-order strict-feedback model of the form

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{f}(\mathbf{p}) + \mathbf{G}(\mathbf{p})\mathbf{z}, \\ \dot{\mathbf{z}} &= \mathbf{f}_1(\mathbf{p}, \mathbf{z}) + \mathbf{G}_1(\mathbf{p}, \mathbf{z})\mathbf{u}, \end{aligned} \quad (16)$$

where the state is now the pair  $(\mathbf{p}, \mathbf{z}) \in \mathbb{R}^p \times \mathbb{R}^z$ ,  $\mathbf{u} \in \mathbb{R}^m$  is the control input, the additional fields  $\mathbf{f}_1 : \mathbb{R}^p \times \mathbb{R}^z \rightarrow \mathbb{R}^z$  and  $\mathbf{G}_1 : \mathbb{R}^p \times \mathbb{R}^z \rightarrow \mathbb{R}^{z \times m}$  are also locally Lipschitz, and the gain matrix  $\mathbf{G}_1(\mathbf{p}, \mathbf{z})$  has full row rank for all  $(\mathbf{p}, \mathbf{z}) \in \mathbb{R}^p \times \mathbb{R}^z$ .

The agent will operate in an environment cluttered with  $M$  dynamic obstacles, indexed by  $i \in \mathcal{I} = \{1, \dots, M\}$ . For each  $i \in \mathcal{I}$ , the obstacle  $i$  is represented by the set

$$\mathcal{O}_i(\mathbf{o}_i) = \mathbf{o}_i + \bar{\mathcal{O}}_i, \quad (17)$$

where  $\mathbf{o}_i \in \mathbb{R}^p$  is the position of the obstacle, which matches its center, and  $\bar{\mathcal{O}}_i \subset \mathbb{R}^p$  is a compact convex set with nonempty interior that defines the geometry of the obstacle relative to its center. From the standpoint of the agent, the obstacles evolve according to uncertain linear dynamics. More specifically, the motion of obstacle  $i$  is governed by the linear model

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{F}_i \mathbf{x}_i + \mathbf{w}_i, \\ \mathbf{o}_i &= \mathbf{E}_i \mathbf{x}_i, \end{aligned} \quad (18)$$

where  $\mathbf{x}_i \in \mathbb{R}^{n_i}$  is the state,  $\mathbf{w}_i \in \mathbb{R}^{n_i}$  represents an uncertain input,  $\mathbf{F}_i \in \mathbb{R}^{n_i \times n_i}$  is a known matrix that describes the free dynamics, and  $\mathbf{E}_i \in \mathbb{R}^{p \times n_i}$  is an auxiliary matrix that extracts the position from the state. The initial state  $\mathbf{x}_{i,0}$  and input  $\mathbf{w}_i$  are known to belong to compact convex sets  $\mathcal{X}_{i,0}, \mathcal{W}_i \subset \mathbb{R}^{n_i}$ .

At each sampling instant  $t_k = kT_s$ , for  $k \in \mathbb{N}$  and sampling period  $T_s \in \mathbb{R}_{>0}$ , the agent obtains a measurement  $\mathbf{y}_{i,k} \in \mathbb{R}^{y_i}$  of the state of each obstacle  $i \in \mathcal{I}$ . Each measurement follows a linear observation model of the form

$$\mathbf{y}_{i,k} = \mathbf{C}_i \mathbf{x}_{i,k} + \mathbf{v}_{i,k}, \quad (19)$$

where  $\mathbf{C}_i \in \mathbb{R}^{y_i \times n_i}$  is the observation matrix, and  $\mathbf{v}_{i,k} \in \mathbb{R}^{y_i}$  is the measurement noise, which belongs to a compact convex set  $\mathcal{V}_i \subset \mathbb{R}^{y_i}$ . Also,  $\mathbf{x}_{i,k}$  denotes the state of the  $i$ th obstacle at the sampling instant  $t_k$ . With this setup in place, we are now ready to formally state the problem addressed in this paper.

**Problem Statement.** Building on the previous setup, consider also a nominal controller for the agent, defined for all  $t \in \mathbb{R}_{\geq 0}$ , that encodes a desired control objective. Then, design a control algorithm that modifies the nominal controller, preferably in a minimally invasive way, to ensure that the agent never collides with any obstacle. Formally, the algorithm must guarantee that

$$\mathcal{P}(\varphi(t)) \cap \left( \bigcup_{i \in \mathcal{I}} \mathcal{O}_i(\varphi_i(t)) \right) = \emptyset \quad (20)$$

for all  $t \in \mathbb{R}_{\geq 0}$ , where  $\varphi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^p$  is the trajectory of the agent, and  $\varphi_i : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^p$  is the trajectory of obstacle  $i$ .

**Remark 2** (Practical Scenarios). This problem setup captures several practical scenarios of interest:

- Cooperative agents: an obstacle may correspond to another autonomous agent that shares information, such as its control law and estimates. In such cases,  $\mathbf{F}_i$  represents the resulting closed-loop dynamics, while  $\mathbf{w}_i$  accounts for bounded disturbances and modeling errors.
- Noncooperative agents and uncontrolled objects: Additionally, an obstacle may correspond to a noncooperative agent or uncontrolled object for which only input bounds are known. In such cases, the control input and disturbances are absorbed into the uncertain input  $\mathbf{w}_i$ , i.e.,

$$\mathbf{w}_i = \mathbf{B}_i \mathbf{u}_i + \mathbf{d}_i, \quad (21)$$

where  $\mathbf{u}_i$  is a control input known to belong to a compact convex set  $\mathcal{U}_i$  and  $\mathbf{d}_i$  is a disturbance that belongs to a

compact convex set  $\mathcal{D}_i$ , and consequently

$$\mathcal{W}_i = \mathbf{B}_i \mathcal{U}_i \oplus \mathcal{D}_i. \quad (22)$$

Additionally, measurements are limited to onboard sensing affected by bounded noise.

**Remark 3** (Agent Geometry). Note that we consider that the agent moves without rotation, so its attitude is not modeled and the body set  $\bar{\mathcal{P}}$  is constant. This assumption is made only for clarity of exposition and because agent rotation is not central to the focus of this paper. The approach proposed herein readily extends to rotating agents, in which case  $\bar{\mathcal{P}}$  becomes explicitly dependent on the orientation of the agent. Also, the position  $\mathbf{p}$  does not actually need to be the physical center of the agent; it may be any fixed point on the body. Under a coordinate change  $\mathbf{p}' = \mathbf{p} + \mathbf{t}$ , the body set of the agent becomes  $\bar{\mathcal{P}}' = \bar{\mathcal{P}} - \mathbf{t}$ .

**Remark 4** (Agent Dynamics). The full row rank of the gain matrices from the models in (15) and (16) means that the agent is fully actuated. Nevertheless, we highlight that these models are quite general. Even if the dynamics do not originally match any of these forms, they can often be transformed to do so via a suitable change of coordinates. For instance, in the case of the unicycle model, controlling a point slightly displaced from the original control point yields a first-order control-affine system with a full-row-rank gain matrix [53]. This trick is commonly used for controlling underactuated vehicles [54], [55]. Also, we emphasize that these models capture most cases of practical interest, such as ground, marine, and aerial vehicles [56], [57], [58], spacecraft [59], [60], and others. Despite the generality of the systems under consideration, we emphasize that these serve only to allow a more systematic control design and are not directly related to the main contributions of this paper.

#### IV. PROPOSED SOLUTION

This section provides a high-level description of our solution approach, which relies on two main components: a guaranteed state estimation algorithm based on CCGs (Section V), and a conversion procedure from CCGs to CBFs (Section VI).

For convenience, we begin by observing that we can equivalently treat the agent as a point-mass agent and account for its body by enlarging the obstacle sets accordingly. Specifically, for each  $i \in \mathcal{I}$ , we introduce the enlarged obstacle set

$$\mathcal{O}_i^+(\mathbf{o}_i) = \mathbf{o}_i + \bar{\mathcal{O}}_i^+ = \mathbf{o}_i + \bar{\mathcal{O}}_i \oplus (-\bar{\mathcal{P}}), \quad (23)$$

which incorporates the agent body. With such a definition, the obstacle avoidance condition can be equivalently expressed as

$$\varphi(t) \notin \bigcup_{i \in \mathcal{I}} \mathcal{O}_i^+(\varphi_i(t)) \quad (24)$$

for all  $t \in \mathbb{R}_{\geq 0}$ . From now on, we adopt this equivalent point-mass standpoint for describing the proposed solution.

Our approach builds upon a state-of-the-art guaranteed state estimation algorithm based on CCGs, whose implementation is detailed in Section V. This algorithm is used for obtaining set-valued estimates of the obstacle states, which are guaranteed to contain the actual states. Specifically, at each sampling instant  $t_k$ , we apply this algorithm to each obstacle  $i \in \mathcal{I}$ , obtaining a CCG state estimate  $\hat{\mathcal{X}}_{i,k} \subset \mathbb{R}^{n_i}$  that contains the actual state  $\mathbf{x}_{i,k}$ , based on the measurements collected up to that time.

During the sampling interval  $[t_k, t_{k+1})$ , we can then obtain guaranteed estimates of the obstacle sets by propagating the obstacle-position estimates through the obstacle dynamics and inflating them by the obstacle bodies. To capture the estimates of the  $i$ th obstacle during the sampling interval, we introduce the set-valued flow  $\hat{\mathcal{O}}_{i,k}^+ : [t_k, t_{k+1}) \rightrightarrows \mathbb{R}^p$ , which assigns to each time  $t \in [t_k, t_{k+1})$  an estimated obstacle set  $\hat{\mathcal{O}}_{i,k}^+(t)$  that includes the true obstacle set. Such an estimate is constructed by projecting the reachable set of states onto the position space and adding the obstacle body set as follows:

$$\hat{\mathcal{O}}_{i,k}^+(t) = \mathbf{E}_i \left( \Phi_i(t - t_k) \hat{\mathcal{X}}_{i,k} \oplus \Gamma_i(t - t_k) \tilde{\mathcal{W}}_i \right) \oplus \bar{\mathcal{O}}_i^+, \quad (25)$$

where  $\Phi_i(s)$  denotes the state-transition matrix, given by

$$\Phi_i(s) = \exp(\mathbf{F}_i s). \quad (26)$$

Also, if the input  $\mathbf{w}_i$  is known to be constant over the sampling interval, the function  $\Gamma_i : \mathbb{R} \rightarrow \mathbb{R}^{n_i \times n_i}$  is given exactly by

$$\Gamma_i(s) = \int_0^s \Phi_i(\tau) d\tau, \quad (27)$$

and the set  $\tilde{\mathcal{W}}_i$  matches  $\mathcal{W}_i$ . If the evolution of the input over the sampling interval is not known, we can construct an over-approximation of the reachable set by defining  $\Gamma_i$  as

$$\Gamma_i(s) = \frac{\exp(\|\mathbf{F}_i\|s) - 1}{\|\mathbf{F}_i\|} \max_{\mathbf{w}_i \in \mathcal{W}_i} \|\mathbf{w}_i\|, \quad (28)$$

in which case the set  $\tilde{\mathcal{W}}_i$  is defined as a unit ball centered at the origin for the euclidean norm (see, e.g., [61]).

For each  $i \in \mathcal{I}$ , we then convert the set-valued flow  $\hat{\mathcal{O}}_{i,k}^+$  into a CBF  $h_{i,k} : \mathbb{R}^p \times [t_k, t_{k+1}) \rightarrow \mathbb{R}$  for (15), which defines a safe set-valued flow  $\mathcal{C}_{i,k} : [t_k, t_{k+1}) \rightrightarrows \mathbb{R}^p$  for the agent as

$$\mathcal{C}_{i,k}(t) = \{\mathbf{p} \in \mathbb{R}^p : h_{i,k}(\mathbf{p}, t) \geq 0\}, \quad (29)$$

such that  $\mathcal{C}_{i,k}(t) \subseteq \mathbb{R}^p \setminus \hat{\mathcal{O}}_{i,k}^+(t)$  for all  $t \in [t_k, t_{k+1})$ . However, the challenge here is that the estimated obstacle set  $\hat{\mathcal{O}}_{i,k}^+(t)$  will be a CCG, rendering the conversion nontrivial. The proposed conversion procedure is presented in Section VI.

At each time  $t \in [t_k, t_{k+1})$ , the overall safe set for the agent position is the intersection of the individual safe sets,

$$\bigcap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t) = \{\mathbf{p} \in \mathbb{R}^p : h_{i,k}(\mathbf{p}, t) \geq 0 \text{ for all } i \in \mathcal{I}\}, \quad (30)$$

which can be compactly written using the minimum function:

$$\bigcap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t) = \left\{ \mathbf{p} \in \mathbb{R}^p : \min_{i \in \mathcal{I}} h_{i,k}(\mathbf{p}, t) \geq 0 \right\}. \quad (31)$$

However, as the minimum function is not differentiable, it cannot be directly used to define a CBF candidate. To address this, we consider a smooth approximation of the minimum function, adopting the LogSumExp approach from [62]. Specifically, we define an overall safe set-valued flow  $\mathcal{C}_k : [t_k, t_{k+1}) \rightrightarrows \mathbb{R}^p$  as

$$\mathcal{C}_k(t) = \{\mathbf{p} \in \mathbb{R}^p : h_k(\mathbf{p}, t) \geq 0\}, \quad (32)$$

where the overall CBF candidate  $h_k : \mathbb{R}^p \times [t_k, t_{k+1}) \rightarrow \mathbb{R}$  is defined for all  $(\mathbf{p}, t) \in \mathbb{R}^p \times [t_k, t_{k+1})$  as

$$h_k(\mathbf{p}, t) = -\frac{1}{\beta_k} \ln \left( \sum_{i \in \mathcal{I}} \exp(-\beta_k h_{i,k}(\mathbf{p}, t)) \right) - \frac{b}{\beta_k}, \quad (33)$$

with tuning parameters  $\beta_k, b \in \mathbb{R}_{>0}$ . This construction ensures

that, for all  $(\mathbf{p}, t) \in \mathbb{R}^p \times [t_k, t_{k+1}]$ , we have

$$h_k(\mathbf{p}, t) < \min_{i \in \mathcal{I}} h_{i,k}(\mathbf{p}, t), \quad (34)$$

implying that  $\mathcal{C}_k(t) \subset \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t))$ . Also, as  $\beta_k \rightarrow \infty$ , the smooth approximation converges to the exact intersection:

$$\lim_{\beta_k \rightarrow \infty} \mathcal{C}_k(t) = \bigcap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t). \quad (35)$$

Finally, we note that the gradient of  $h_k$  can be expressed as

$$\frac{\partial h_k(\mathbf{p}, t)}{\partial (\mathbf{p}, t)} = \sum_{i \in \mathcal{I}} \pi_i(\mathbf{p}, t) \frac{\partial h_{i,k}(\mathbf{p}, t)}{\partial (\mathbf{p}, t)}, \quad (36)$$

where each weight  $\pi_i(\mathbf{p}, t)$  is defined by

$$\pi_i(\mathbf{p}, t) = \exp(-\beta_k(h_{i,k}(\mathbf{p}, t) - h_k(\mathbf{p}, t)) + b). \quad (37)$$

If  $h_k$  is a valid CBF for (15), then it can be used to design a safe controller for the sampling interval  $[t_k, t_{k+1}]$ .

**Proposition 1** (Valid CBF for (15)). Let  $h : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}$  be a continuously differentiable function with  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) \neq \mathbf{0}$  when  $h(\mathbf{p}, t) = 0$ . If when  $\frac{\partial}{\partial t} h(\mathbf{p}, t) = \mathbf{0}$  and  $h(\mathbf{p}, t) > 0$ ,

$$\frac{\partial h(\mathbf{p}, t)}{\partial t} \geq 0, \quad (38)$$

then  $h$  is a CBF for the system (15), for an arbitrary associated extended class- $\mathcal{K}_\infty$  function  $\alpha$ .

*Proof.* Let  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  be the set-valued flow associated with  $h$ , defined for all  $t \in [t_0, t_f]$  as

$$\mathcal{C}(t) = \{\mathbf{p} \in \mathbb{R}^p : h(\mathbf{p}, t) \geq 0\}. \quad (39)$$

Given the differentiability and regularity assumptions, showing that  $h$  is a CBF for (15) amounts to verifying the existence of a set-valued flow  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  with  $\mathcal{C}(t) \subseteq \mathcal{D}(t)$  for all  $t \in [t_0, t_f]$  and an extended class- $\mathcal{K}_\infty$  function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  so that, for all  $(\mathbf{p}, t) \in \mathcal{G}(\mathcal{D})$ , we have

$$\sup_{\mathbf{z} \in \mathbb{R}^z} \dot{h}(\mathbf{p}, t, \mathbf{z}) > -\alpha(h(\mathbf{p}, t)). \quad (40)$$

We begin by noting that when  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) \neq \mathbf{0}$ , the value of  $\dot{h}$  can be made arbitrarily large through input, meaning that

$$\sup_{\mathbf{z} \in \mathbb{R}^z} \dot{h}(\mathbf{p}, t, \mathbf{z}) = \infty > -\alpha(h(\mathbf{p}, t)), \quad (41)$$

for any function  $\alpha$ . When  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) = \mathbf{0}$  and  $h(\mathbf{p}, t) > 0$ ,

$$\dot{h}(\mathbf{p}, t, \mathbf{z}) = \frac{\partial h(\mathbf{p}, t)}{\partial t} \geq 0 > -\alpha(h(\mathbf{p}, t)), \quad (42)$$

for any extended class- $\mathcal{K}_\infty$  function  $\alpha$ . Hence, for an arbitrary extended class- $\mathcal{K}_\infty$  function  $\alpha$ , the inequality in (40) holds for at least all  $(\mathbf{p}, t) \in \mathcal{G}(\mathcal{C})$ , completing the proof. ■

**Remark 5.** If the inequality in (38) does not hold at all points where  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) = \mathbf{0}$  and  $h(\mathbf{p}, t) > 0$ , then we must find an extended class- $\mathcal{K}_\infty$  function  $\alpha$  such that

$$\frac{\partial h(\mathbf{p}, t)}{\partial t} > -\alpha(h(\mathbf{p}, t)) \quad (43)$$

whenever  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) = \mathbf{0}$ ,  $h(\mathbf{p}, t) > 0$ , and  $\frac{\partial}{\partial t} h(\mathbf{p}, t) < 0$  in order for  $h$  to be a valid CBF for (15) under Definition 5. In practice, however, a simpler approach is to locally relax the CBF constraint around such points in the controller design.

---

### Algorithm 1 Safe Navigation Scheme

---

**Require:** agent, obstacles, and measurement models

```

1: for  $k \in \mathbb{N}$  do
2:   for  $i \in \mathcal{I}$  do
3:     collect measurement  $\mathbf{y}_{i,k}$ 
4:     compute state estimate  $\hat{\mathcal{X}}_{i,k}$  (Section V)
5:     convert  $\hat{\mathcal{O}}_{i,k}$  into a CBF  $h_{i,k}$  for (15) (Section VI)
6:   end for
7:   design overall CBF  $h_k / h_{k,1}$  for (15) / (16)
8:   apply safe controller during  $[t_k, t_{k+1}]$ 
9: end for
```

---

When our agent has first-order dynamics as in (15) and  $h_k$  is a valid CBF for it on a set-valued flow  $\mathcal{D}_k : [t_k, t_{k+1}] \rightrightarrows \mathbb{R}^p$ , we can then construct a safety filter  $\mathbf{k}_k : \mathcal{G}(\mathcal{D}_k) \rightarrow \mathbb{R}^z$  for the sampling interval  $[t_k, t_{k+1}]$  through the following QP:

$$\begin{aligned} \mathbf{k}_k(\mathbf{p}, t) = \arg \min_{\mathbf{z} \in \mathbb{R}^z} \frac{1}{2} \|\mathbf{z} - \mathbf{k}_d(\mathbf{p}, t)\|^2 \\ \text{subject to } \dot{h}_k(\mathbf{p}, t, \mathbf{z}) \geq -\alpha(h_k(\mathbf{p}, t)), \end{aligned} \quad (44)$$

where  $\mathbf{k}_d : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^z$  denotes the nominal controller for the agent. If the position of the agent at the instant  $t_k$ , denoted as  $\mathbf{p}_k$ , satisfies  $\mathbf{p}_k \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t_k))$ , then we can select the parameter  $\beta_k$  so that  $h_k(\mathbf{p}_k, t_k) \geq 0$ , thereby ensuring safety over the time interval  $[t_k, t_{k+1}]$ . For instance, given a nominal value  $\bar{\beta} \in \mathbb{R}_{>0}$ , the smoothing parameter  $\beta_k$  can be chosen as

$$\beta_k = \max \{\bar{\beta}, \beta_{\min,k} + \epsilon\}, \quad (45)$$

where  $\beta_{\min,k}$  is the value of  $\beta_k$  for which  $h_k(\mathbf{p}_k, t_k) = 0$ , and  $\epsilon \in \mathbb{R}_{\geq 0}$  is a buffer that ensures  $h_k(\mathbf{p}_k, t_k) > 0$  when positive. Global safety for all times  $t \in \mathbb{R}_{\geq 0}$  is then achieved if, at every sampling time  $t_k$ , it always holds that  $\mathbf{p}_k \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t_k))$ .

For agents with second-order dynamics as in (16), the state  $\mathbf{z}$  cannot be directly controlled; thus, we must *backstep* through  $\mathbf{z}$  to design a CBF  $h_{k,1} : \mathbb{R}^p \times \mathbb{R}^z \times [t_k, t_{k+1}] \rightarrow \mathbb{R}$  for (16). The next subsection elaborates on this additional design step, and Algorithm 1 summarizes the overall navigation scheme.

#### A. CBF Backstepping

This subsection describes the additional backstepping-based design required for agents with second-order dynamics. To this end, we begin by extending the CBF backstepping result from [63] to time-dependent CBFs.

**Proposition 2** (CBF Backstepping). Let  $h : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}$  be a CBF for (15) on a set-valued flow  $\mathcal{D} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$ , and let  $\mathbf{k} : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}^z$  denote a continuously differentiable feedback controller such that, for all  $(\mathbf{p}, t) \in \mathcal{G}(\mathcal{D})$ ,

$$\dot{h}(\mathbf{p}, t, \mathbf{k}(\mathbf{p}, t)) > -\alpha(h(\mathbf{p}, t)), \quad (46)$$

where  $\alpha$  is an extended class- $\mathcal{K}_\infty$  function. Then, the function  $h_1 : \mathbb{R}^p \times \mathbb{R}^z \times [t_0, t_f] \rightarrow \mathbb{R}$ , defined as

$$h_1(\mathbf{p}, \mathbf{z}, t) = h(\mathbf{p}, t) - \frac{1}{2\sigma} \|\mathbf{z} - \mathbf{k}(\mathbf{p}, t)\|^2 \quad (47)$$

for all  $(\mathbf{p}, \mathbf{z}, t) \in \mathbb{R}^p \times \mathbb{R}^z \times [t_0, t_f]$ , with  $\sigma \in \mathbb{R}_{>0}$ , is a CBF for the system (16), for an associated extended class- $\mathcal{K}_\infty$  function  $\alpha_1$  such that  $\alpha_1(s) \geq \alpha(s)$  for all  $s \in \mathbb{R}$ .

*Proof.* Let  $\mathcal{C}_1 : [t_0, t_f] \rightrightarrows \mathbb{R}^p \times \mathbb{R}^z$  denote the set-valued flow associated with  $h_1$ , defined for all  $t \in [t_0, t_f]$  as

$$\mathcal{C}_1(t) = \{(\mathbf{p}, \mathbf{z}) \in \mathbb{R}^p \times \mathbb{R}^z : h_1(\mathbf{p}, \mathbf{z}, t) \geq 0\}. \quad (48)$$

To show that  $h_1$  is a CBF for the system (16), we have to verify that it satisfies the following properties: (i)  $h_1$  is continuously differentiable, (ii)  $\frac{\partial}{\partial(\mathbf{p}, \mathbf{z})} h_1(\mathbf{p}, \mathbf{z}, t) \neq \mathbf{0}$  when  $h_1(\mathbf{p}, \mathbf{z}, t) = 0$ , and (iii) there exists a set-valued flow  $\mathcal{D}_1 : [t_0, t_f] \rightrightarrows \mathbb{R}^p \times \mathbb{R}^z$  with  $\mathcal{C}_1(t) \subseteq \mathcal{D}_1(t)$  for all  $t \in [t_0, t_f]$  and an extended class- $\mathcal{K}_\infty$  function  $\alpha_1$  such that, for all  $(\mathbf{p}, \mathbf{z}, t) \in \mathcal{G}(\mathcal{D}_1)$ ,

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \dot{h}_1(\mathbf{p}, \mathbf{z}, t, \mathbf{u}) > -\alpha_1(h_1(\mathbf{p}, \mathbf{z}, t)). \quad (49)$$

The properties (i) and (ii) are straightforward to verify since  $\mathbf{k}$  is continuously differentiable and  $h$  is a CBF for the system (15). To show (iii), we note that for  $\mathbf{z} \neq \mathbf{k}(\mathbf{p}, t)$ , the value of  $h_1$  can be made arbitrarily large through input, meaning that

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \dot{h}_1(\mathbf{p}, \mathbf{z}, t, \mathbf{u}) = \infty > -\alpha(h_1(\mathbf{p}, \mathbf{z}, t)), \quad (50)$$

and for  $\mathbf{z} = \mathbf{k}(\mathbf{p}, t)$  and  $(\mathbf{p}, t) \in \mathcal{G}(\mathcal{D})$ , we have

$$\begin{aligned} \dot{h}_1(\mathbf{p}, \mathbf{k}(\mathbf{p}, t), t, \mathbf{u}) &> -\alpha(h(\mathbf{p}, t)) \\ &= -\alpha(h_1(\mathbf{p}, \mathbf{k}(\mathbf{p}, t), t)). \end{aligned} \quad (51)$$

Hence, the condition in (49) holds for all  $(\mathbf{p}, t, \mathbf{z}) \in \mathcal{G}(\mathcal{D}) \times \mathbb{R}^z$  and an extended class- $\mathcal{K}_\infty$  function  $\alpha_1$  with  $\alpha_1(s) \geq \alpha(s)$  for all  $s \in \mathbb{R}$ , which concludes the proof. ■

**Remark 6.** The CBF  $h_1$  can be employed to render the set-valued flow  $\mathcal{C}_1$  forward invariant for the system (16). Therefore, to render the set-valued flow  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  associated with  $h$  forward invariant for the top-level subsystem of (16), the initial state  $(\mathbf{p}_0, \mathbf{z}_0)$  must satisfy  $h_1(\mathbf{p}_0, \mathbf{z}_0, t_0) \geq 0$ . If the initial position lies in the interior of  $\mathcal{C}(t_0)$ , i.e.,  $h(\mathbf{p}_0, t_0) > 0$ , then this requirement can be satisfied by selecting

$$\sigma \geq \frac{1}{2h(\mathbf{p}_0, t_0)} \|\mathbf{z}_0 - \mathbf{k}(\mathbf{p}_0, t_0)\|^2. \quad (52)$$

Building on the previous result, we can then design a CBF  $h_{k,1} : \mathbb{R}^p \times \mathbb{R}^z \times [t_k, t_{k+1}] \rightarrow \mathbb{R}$  for (16) as

$$h_{k,1}(\mathbf{p}, \mathbf{z}, t) = h_k(\mathbf{p}, t) - \frac{1}{2\sigma_k} \|\mathbf{z} - \mathbf{k}_k(\mathbf{p}, t)\|^2, \quad (53)$$

where  $\sigma_k \in \mathbb{R}_{>0}$ . It is important to note, however, that in this formulation the top-level controller  $\mathbf{k}_k$  must be continuously differentiable. Hence,  $\mathbf{k}_k$  cannot be designed by means of a QP as in (44) because QPs typically only guarantee local Lipschitz continuity. To ensure smoothness, we adopt the technique from [64], which constructs a smooth controller based on Gaussian-weighted centroids. Specifically, consider the set

$$K_{h_k}(\mathbf{p}, t) = \{\mathbf{z} \in \mathbb{R}^z : \dot{h}_k(\mathbf{p}, t, \mathbf{z}) \geq -\alpha(h_k(\mathbf{p}, t))\}. \quad (54)$$

Following [64], we can design a top-level controller  $\mathbf{k}_k$  as

$$\mathbf{k}_k(\mathbf{p}, t) = \boldsymbol{\mu}(K_{h_k}(\mathbf{p}, t)), \quad (55)$$

where  $\boldsymbol{\mu}$  is the Gaussian-weighted centroid function, given by

$$\boldsymbol{\mu}(\mathcal{S}) = \frac{\int_{\mathcal{S}} \mathbf{z} \exp(-\|\mathbf{z}\|^2/(2\varsigma)) d\mathbf{z}}{\int_{\mathcal{S}} \exp(-\|\mathbf{z}\|^2/(2\varsigma)) d\mathbf{z}}, \quad (56)$$

for  $\mathcal{S} \subseteq \mathbb{R}^z$  and  $\varsigma \in \mathbb{R}_{>0}$ . This controller can be expressed in closed-form, and it is smooth if the agent dynamics, the CBF gradient field, and the function  $\alpha$  are smooth [64].

When our agent has second-order dynamics as in (16) and  $h_{k,1}$  is a valid CBF for it on  $\mathcal{D}_{k,1} : [t_k, t_{k+1}] \rightrightarrows \mathbb{R}^p \times \mathbb{R}^z$ , we can then construct a safe controller  $\mathbf{k}_{d,1} : \mathcal{G}(\mathcal{D}_{k,1}) \rightarrow \mathbb{R}^m$  for the sampling interval  $[t_k, t_{k+1}]$  through the following QP:

$$\begin{aligned} \mathbf{k}_{d,1}(\mathbf{p}, \mathbf{z}, t) &= \arg \min_{\mathbf{u} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{u} - \mathbf{k}_{d,1}(\mathbf{p}, \mathbf{z}, t)\|^2 \\ \text{subject to } \dot{h}_{k,1}(\mathbf{p}, \mathbf{z}, t, \mathbf{u}) &\geq -\alpha_1(h_{k,1}(\mathbf{p}, \mathbf{z}, t)), \end{aligned} \quad (57)$$

where  $\mathbf{k}_{d,1} : \mathbb{R}^p \times \mathbb{R}^z \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  is the nominal controller for the agent. If the position of the agent at the time instant  $t_k$  satisfies  $\mathbf{p}_k \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t_k))$ , then selecting a positive  $\epsilon$  in (45) guarantees  $h_k(\mathbf{p}_k, t_k) > 0$ . As a result, we can then select the gain  $\sigma_k$  such that  $h_{k,1}(\mathbf{p}_k, \mathbf{z}_k, t_k) \geq 0$ , thereby ensuring safety over the sampling interval  $[t_k, t_{k+1}]$ . For instance, given a nominal value  $\bar{\sigma} \in \mathbb{R}_{>0}$ , the gain  $\sigma_k$  can be selected as

$$\sigma_k = \max \left\{ \bar{\sigma}, \frac{1}{2h_k(\mathbf{p}_k, t_k)} \|\mathbf{z}_k - \mathbf{k}_k(\mathbf{p}_k, t_k)\|^2 \right\}. \quad (58)$$

Global safety for all times  $t \in \mathbb{R}_{\geq 0}$  is then achieved if, at every sampling time  $t_k$ , it always holds that  $\mathbf{p}_k \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t_k))$ , as in the first-order case (see Remark 9, Section VI).

## V. GUARANTEED OBSTACLE-STATE ESTIMATION USING CONSTRAINED CONVEX GENERATORS

This section details the first main component of our solution, which is the estimation algorithm used for obtaining guaranteed estimates of the obstacle states. For notational simplicity, in this section we omit the obstacle index  $i$  from (18)-(19) and consider a generic linear model of the form

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{w}, \quad (59)$$

with  $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$ ,  $\mathbf{F} \in \mathbb{R}^{n \times n}$ , and the measurement model is

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \quad (60)$$

with  $\mathbf{C} \in \mathbb{R}^{y \times n}$  and  $\mathbf{v}_k \in \mathbb{R}^y$ . The subproblem addressed in this section can then be summarized as follows.

**Subproblem 1** (Guaranteed Obstacle-State Estimation). Consider the linear system (59) and the measurement model (60). Then, design an estimation algorithm that, at the sampling time  $t_k$ , computes a state estimate  $\hat{\mathbf{x}}_k$  such that  $\mathbf{x}_k \in \hat{\mathcal{X}}_k$ , given the collected measurements and the compact convex sets  $\mathcal{X}_0$ ,  $\mathcal{W}$ , and  $\mathcal{V}$ , such that  $\mathbf{x}_0 \in \mathcal{X}_0$ ,  $\mathbf{w} \in \mathcal{W}$ , and  $\mathbf{v}_k \in \mathcal{V}$ .

### A. Constrained Convex Generators

Prior to presenting the main results of this section, we first review the CCG class of sets, which constitutes the state-of-the-art framework for guaranteed state estimation.

**Definition 6** (CCG [20]). A set  $\mathcal{Z} \subset \mathbb{R}^n$  is a CCG if there exists a tuple  $(\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}) \in \mathbb{R}^{n \times \xi} \times \mathbb{R}^n \times \mathbb{R}^{c \times \xi} \times \mathbb{R}^c$  and a set  $\mathfrak{G} = \mathcal{G}_1 \times \mathcal{G}_2 \times \dots \times \mathcal{G}_G \subset \mathbb{R}^\xi$  such that

$$\mathcal{Z} = \{\mathbf{G}\xi + \mathbf{c} : \mathbf{A}\xi = \mathbf{b}, \xi \in \mathfrak{G}\}, \quad (61)$$

where, for each  $j \in \{1, \dots, G\}$ , the generator set  $\mathcal{G}_j$  is defined as the zero-sublevel set of a convex function  $g_j : \mathbb{R}^{\xi_j} \rightarrow \mathbb{R}$ :

$$\mathcal{G}_j = \{\xi_j \in \mathbb{R}^{\xi_j} : g_j(\xi_j) \leq 0\}. \quad (62)$$

For a CCG  $\mathcal{Z}$  defined as in (61), we introduce the shorthand notation  $\mathcal{Z} = (\mathbf{G}, \mathbf{c}, \mathbf{A}, \mathbf{b}, \mathfrak{G}) \subset \mathbb{R}^n$ . Given this formulation,

the following proposition asserts that CCGs are closed under three fundamental set operations (affine map, Minkowski sum, and generalized intersection) and that these operations can be performed in closed form through simple identities that follow almost directly from the definition of CCG.

**Proposition 3** (Set Operations with CCGs [20]). Consider a matrix  $\mathbf{R} \in \mathbb{R}^{y \times n}$ , a vector  $\mathbf{t} \in \mathbb{R}^y$ , and three CCGs:

- $\mathcal{Z} = (\mathbf{G}_z, \mathbf{c}_z, \mathbf{A}_z, \mathbf{b}_z, \mathfrak{G}_z) \subset \mathbb{R}^n$ ;
- $\mathcal{W} = (\mathbf{G}_w, \mathbf{c}_w, \mathbf{A}_w, \mathbf{b}_w, \mathfrak{G}_w) \subset \mathbb{R}^n$ ;
- $\mathcal{V} = (\mathbf{G}_v, \mathbf{c}_v, \mathbf{A}_v, \mathbf{b}_v, \mathfrak{G}_v) \subset \mathbb{R}^y$ .

Then, regarding the affine map, Minkowski sum, and generalized intersection operations, the following identities hold:

$$\begin{aligned} \mathbf{R}\mathcal{Z} + \mathbf{t} &= (\mathbf{R}\mathbf{G}_z, \mathbf{R}\mathbf{c}_z + \mathbf{t}, \mathbf{A}_z, \mathbf{b}_z, \mathfrak{G}_z), \\ \mathcal{Z} \oplus \mathcal{W} &= \left( [\mathbf{G}_z \mathbf{G}_w], \mathbf{c}_z + \mathbf{c}_w, \begin{bmatrix} \mathbf{A}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_w \end{bmatrix}, \begin{bmatrix} \mathbf{b}_z \\ \mathbf{b}_w \end{bmatrix}, \mathfrak{G}_z \times \mathfrak{G}_w \right), \\ \mathcal{Z} \cap_{\mathbf{R}} \mathcal{V} &= \left( [\mathbf{G}_z \mathbf{0}], \mathbf{c}_z, \begin{bmatrix} \mathbf{A}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_v \end{bmatrix}, \begin{bmatrix} \mathbf{b}_z \\ \mathbf{b}_v \end{bmatrix}, \mathfrak{G}_z \times \mathfrak{G}_v \right). \end{aligned} \quad (63)$$

From the identities in (63) and the fact that CCGs are convex by construction, it follows that CCGs are well suited for state estimation in linear systems. Moreover, CCGs constitute a very general class of sets, which significantly reduces the need for approximations. Particularly, they generalize many commonly used set classes, such as intervals, ellipsoids, zonotopes, CZs or polytopes, convex cones, ellipsotopes, or AH-polytopes. For additional details on CCGs, the reader is referred to [20].

Consequently, for the linear system (59) and the observation model (60), if the sets  $\mathcal{X}_0$ ,  $\mathcal{W}$ , and  $\mathcal{V}$  are CCGs, the guaranteed state estimation problem can be solved recursively by applying the operation identities in (63). Specifically, at each sampling instant  $t_k$ , the optimal state estimate is given by the recursion

$$\hat{\mathcal{X}}_k = \left( \Phi(T_s)\hat{\mathcal{X}}_{k-1} \oplus \Gamma(T_s)\tilde{\mathcal{W}} \right) \cap_{\mathbf{C}} (\mathbf{y}_k - \mathcal{V}), \quad (64)$$

with  $\Phi$ ,  $\Gamma$ , and  $\tilde{\mathcal{W}}$  defined as in Section IV, and with an initial state estimate determined as

$$\hat{\mathcal{X}}_0 = \mathcal{X}_0 \cap_{\mathbf{C}} (\mathbf{y}_0 - \mathcal{V}). \quad (65)$$

The recursive formula from (64) produces a new state estimate by first propagating the previous estimate over one sampling period and then updating the propagated set by intersecting it with the set of states consistent with the current measurement.

However, this approach has a major drawback. Since each update of the state estimate involves a Minkowski sum and a generalized intersection, the number of generator sets steadily increases over time, leading to a substantial computational load after a certain number of iterations. To address this limitation, the following subsection introduces a more efficient approach that preserves a fixed-length representation.

### B. Explicit Finite-Horizon Estimator

To keep a constant computational load over time, we adopt a finite-horizon estimation strategy that generalizes the approach recently introduced in [28]. The main point is to fix a horizon length  $N \in \mathbb{N}$  and consider an auxiliary conservative estimator which, at each sampling time  $t_k \geq NT_s$ , provides a CCG state estimate  $\bar{\mathcal{X}}_{k-N}$  corresponding to the earlier instant  $t_{k-N}$ . The current estimate  $\hat{\mathcal{X}}_k$  is then obtained by improving the estimate

$\bar{\mathcal{X}}_{k-N} \cap_{\mathbf{C}} (\mathbf{y}_{k-N} - \mathcal{V})$  through  $N$  iterations of (64).

However, note that a straightforward implementation of this approach would still require executing  $N$  recursive iterations at each sampling instant  $t_k \geq NT_s$ , which would be computationally expensive for large  $N$ . The next result addresses this issue by showing that the CCG parameters of the current state estimate  $\hat{\mathcal{X}}_k$  can be explicitly determined as a function of the parameters of the conservative estimate  $\bar{\mathcal{X}}_{k-N}$ , based on fixed auxiliary variables that can be precomputed offline.

**Theorem 2** (Explicit Computation of CCG Estimates). Consider the linear system (59), the measurement model (60), and let the sets  $\tilde{\mathcal{W}}$  and  $\mathcal{V}$  be CCGs, defined as

$$\begin{aligned} \tilde{\mathcal{W}} &= (\mathbf{G}_{\tilde{w}}, \mathbf{c}_{\tilde{w}}, [ ], [ ], \mathfrak{G}_{\tilde{w}}), \\ \mathcal{V} &= (\mathbf{G}_v, \mathbf{c}_v, [ ], [ ], \mathfrak{G}_v). \end{aligned} \quad (66)$$

Additionally, consider a fixed horizon  $N \in \mathbb{N}$ , and let

$$\bar{\mathcal{X}}_{k-N} = (\mathbf{G}_{\bar{x},k-N}, \mathbf{c}_{\bar{x},k-N}, [ ], [ ], \mathfrak{G}_{\bar{x},k-N}) \quad (67)$$

be a state estimate corresponding to the sampling instant  $t_{k-N}$ . Then, the state estimate at the sampling instant  $t_k$ , attained by improving  $\bar{\mathcal{X}}_{k-N} \cap_{\mathbf{C}} (\mathbf{y}_{k-N} - \mathcal{V})$  through  $N$  iterations of (64), is a CCG of the form

$$\hat{\mathcal{X}}_k = (\mathbf{G}_{\hat{x},k}, \mathbf{c}_{\hat{x},k}, \mathbf{A}_{\hat{x},k}, \mathbf{b}_{\hat{x},k}, \mathfrak{G}_{\hat{x},k}), \quad (68)$$

with parameters explicitly given by

$$\begin{aligned} \mathbf{G}_{\hat{x},k} &= [\mathbf{R}_{1,N} \mathbf{G}_{\bar{x},k-N} \mathbf{R}_{2,N}], \\ \mathbf{c}_{\hat{x},k} &= \mathbf{R}_{1,N} \mathbf{c}_{\bar{x},k-N} + \mathbf{t}_{1,N}, \\ \mathbf{A}_{\hat{x},k} &= [\mathbf{R}_{3,N} \mathbf{G}_{\bar{x},k-N} \mathbf{R}_{4,N}], \\ \mathbf{b}_{\hat{x},k} &= \mathbf{R}_{5,N} \mathbf{c}_{\bar{x},k-N} + \mathbf{t}_{2,N} + \mathbf{y}_{k-N:k}, \\ \mathfrak{G}_{\hat{x},k} &= \mathfrak{G}_{\bar{x},k-N} \times \mathfrak{C}_N, \end{aligned} \quad (69)$$

where  $\mathbf{y}_{k-N:k} = (\mathbf{y}_{k-N}, \dots, \mathbf{y}_k)$ , and the auxiliary variables  $\mathbf{R}_{1,N}, \dots, \mathbf{R}_{5,N}$ ,  $\mathbf{t}_{1,N}$ ,  $\mathbf{t}_{2,N}$ , and  $\mathfrak{C}_N$  are determined through  $N$  iterations of the following recursions:

$$\begin{aligned} \mathbf{R}_{1,l+1} &= \Phi(T_s)\mathbf{R}_{1,l}, \\ \mathbf{R}_{2,l+1} &= [\Phi(T_s)\mathbf{R}_{2,l} \ \Gamma(T_s)\mathbf{G}_{\tilde{w}} \ \mathbf{0}], \\ \mathbf{R}_{3,l+1} &= \begin{bmatrix} \mathbf{R}_{3,l} \\ \mathbf{C}\mathbf{R}_{1,l+1} \end{bmatrix}, \\ \mathbf{R}_{4,l+1} &= \begin{bmatrix} \mathbf{R}_{4,l} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}\Phi(T_s)\mathbf{R}_{2,l} & \mathbf{C}\Gamma(T_s)\mathbf{G}_{\tilde{w}} & \mathbf{G}_v \end{bmatrix}, \\ \mathbf{R}_{5,l+1} &= \begin{bmatrix} \mathbf{R}_{5,l} \\ -\mathbf{C}\mathbf{R}_{1,l+1} \end{bmatrix}, \\ \mathbf{t}_{1,l+1} &= \Phi(T_s)\mathbf{t}_{1,l} + \Gamma(T_s)\mathbf{c}_{\tilde{w}}, \\ \mathbf{t}_{2,l+1} &= \begin{bmatrix} \mathbf{t}_{2,l} \\ -\mathbf{C}\mathbf{t}_{1,l+1} - \mathbf{c}_v \end{bmatrix}, \\ \mathfrak{C}_{l+1} &= \mathfrak{C}_l \times \mathfrak{G}_{\tilde{w}} \times \mathfrak{G}_v, \end{aligned} \quad (70)$$

initialized with  $\mathbf{R}_{1,0} = \mathbf{I}$ ,  $\mathbf{R}_{2,0} = \mathbf{0}$ ,  $\mathbf{R}_{3,0} = \mathbf{C}$ ,  $\mathbf{R}_{4,0} = \mathbf{G}_v$ ,  $\mathbf{R}_{5,0} = -\mathbf{C}$ ,  $\mathbf{t}_{1,0} = \mathbf{0}$ ,  $\mathbf{t}_{2,0} = -\mathbf{c}_v$ ,  $\mathfrak{C}_0 = \mathfrak{G}_v$ .

*Proof.* To prove this result, we have to show that the parameters of  $\bar{\mathcal{X}}_{k-N} \cap_{\mathbf{C}} (\mathbf{y}_{k-N} - \mathcal{V})$  can be expressed in the form of (69) and that performing an iteration of (64) produces a CCG with parameters of the same form. For this purpose, let

$$\hat{\mathcal{X}}_{\bar{k}} = \bar{\mathcal{X}}_{\bar{k}} \cap_{\mathbf{C}} (\mathbf{y}_{\bar{k}} - \mathcal{V}), \quad (71)$$

where  $\bar{k} = k - N$ . Using the generalized intersection identity

from (63), the CCG parameters of  $\hat{\mathcal{X}}_{\bar{k}}$  can be expressed as

$$\begin{aligned}\mathbf{G}_{\hat{x}, \bar{k}} &= [\mathbf{G}_{\bar{x}, \bar{k}} \ \mathbf{0}] = [\mathbf{R}_{1,0} \mathbf{G}_{\bar{x}, \bar{k}} \ \mathbf{R}_{2,0}], \\ \mathbf{c}_{\hat{x}, \bar{k}} &= \mathbf{c}_{\bar{x}, \bar{k}} = \mathbf{R}_{1,0} \mathbf{c}_{\bar{x}, \bar{k}} + \mathbf{t}_{1,0}, \\ \mathbf{A}_{\hat{x}, \bar{k}} &= [\mathbf{C} \mathbf{G}_{\bar{x}, \bar{k}} \ \mathbf{G}_v] = [\mathbf{R}_{3,0} \mathbf{G}_{\bar{x}, \bar{k}} \ \mathbf{R}_{4,0}], \quad (72) \\ \mathbf{b}_{\hat{x}, \bar{k}} &= -\mathbf{C} \mathbf{c}_{\bar{x}, \bar{k}} - \mathbf{c}_v + \mathbf{y}_{\bar{k}} = \mathbf{R}_{5,0} \mathbf{c}_{\bar{x}, \bar{k}} + \mathbf{t}_{2,0} + \mathbf{y}_{\bar{k}}, \\ \mathfrak{G}_{\hat{x}, \bar{k}} &= \mathfrak{G}_{\bar{x}, \bar{k}} \times \mathfrak{G}_v = \mathfrak{G}_{\bar{x}, \bar{k}} \times \mathfrak{C}_0,\end{aligned}$$

with the auxiliary variables  $\mathbf{R}_{1,0} = \mathbf{I}$ ,  $\mathbf{R}_{2,0} = \mathbf{0}$ ,  $\mathbf{R}_{3,0} = \mathbf{C}$ ,  $\mathbf{R}_{4,0} = \mathbf{G}_v$ ,  $\mathbf{R}_{5,0} = -\mathbf{C}$ ,  $\mathbf{t}_{1,0} = \mathbf{0}$ ,  $\mathbf{t}_{2,0} = -\mathbf{c}_v$ , and  $\mathfrak{C}_0 = \mathfrak{G}_v$ . Applying now an iteration of (64) to  $\hat{\mathcal{X}}_{\bar{k}}$ , using the operation identities from (63), yields a state estimate  $\hat{\mathcal{X}}_{\bar{k}+1}$  whose CCG parameters are given by

$$\begin{aligned}\mathbf{G}_{\hat{x}, \bar{k}+1} &= [\Phi(T_s) \mathbf{G}_{\hat{x}, \bar{k}} \ \Gamma(T_s) \mathbf{G}_{\bar{w}} \ \mathbf{0}] \\ &= [\Phi(T_s) \mathbf{R}_{1,0} \mathbf{G}_{\bar{x}, \bar{k}} \ \Phi(T_s) \mathbf{R}_{2,0} \ \Gamma(T_s) \mathbf{G}_{\bar{w}} \ \mathbf{0}] \\ &= [\mathbf{R}_{1,1} \mathbf{G}_{\bar{x}, \bar{k}} \ \mathbf{R}_{2,1}], \\ \mathbf{c}_{\hat{x}, \bar{k}+1} &= \Phi(T_s) \mathbf{c}_{\hat{x}, \bar{k}} + \Gamma(T_s) \mathbf{c}_{\bar{w}} \\ &= \Phi(T_s) \mathbf{R}_{1,0} \mathbf{c}_{\bar{x}, \bar{k}} + \Phi(T_s) \mathbf{t}_{1,0} + \Gamma(T_s) \mathbf{c}_{\bar{w}} \\ &= \mathbf{R}_{1,1} \mathbf{c}_{\bar{x}, \bar{k}} + \mathbf{t}_{1,1}, \\ \mathbf{A}_{\hat{x}, \bar{k}+1} &= \begin{bmatrix} \mathbf{A}_{\hat{x}, \bar{k}} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} \Phi(T_s) \mathbf{G}_{\hat{x}, \bar{k}} & \mathbf{C} \Gamma(T_s) \mathbf{G}_{\bar{w}} & \mathbf{G}_v \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_{3,0} \mathbf{G}_{\bar{x}, \bar{k}} & \mathbf{R}_{4,0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} \mathbf{R}_{1,1} \mathbf{G}_{\bar{x}, \bar{k}} & \mathbf{C} \Phi(T_s) \mathbf{R}_{2,0} & \mathbf{C} \Gamma(T_s) \mathbf{G}_{\bar{w}} & \mathbf{G}_v \end{bmatrix} \quad (73) \\ &= [\mathbf{R}_{3,1} \mathbf{G}_{\bar{x}, \bar{k}} \ \mathbf{R}_{4,1}], \\ \mathbf{b}_{\hat{x}, \bar{k}+1} &= \begin{bmatrix} \mathbf{b}_{\hat{x}, \bar{k}} \\ -\mathbf{C} \mathbf{c}_{\hat{x}, \bar{k}+1} - \mathbf{c}_v + \mathbf{y}_{\bar{k}+1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_{5,0} \mathbf{c}_{\bar{x}, \bar{k}} + \mathbf{t}_{2,0} + \mathbf{y}_{\bar{k}} \\ -\mathbf{C} \mathbf{R}_{1,1} \mathbf{c}_{\bar{x}, \bar{k}} - \mathbf{C} \mathbf{t}_{1,1} - \mathbf{c}_v + \mathbf{y}_{\bar{k}+1} \end{bmatrix} \\ &= \mathbf{R}_{5,1} \mathbf{c}_{\bar{x}, \bar{k}} + \mathbf{t}_{2,1} + \mathbf{y}_{\bar{k}: \bar{k}+1}, \\ \mathfrak{G}_{\hat{x}, \bar{k}+1} &= \mathfrak{G}_{\hat{x}, \bar{k}} \times \mathfrak{G}_{\bar{w}} \times \mathfrak{G}_v \\ &= \mathfrak{G}_{\bar{x}, \bar{k}} \times \mathfrak{C}_0 \times \mathfrak{G}_{\bar{w}} \times \mathfrak{G}_v \\ &= \mathfrak{G}_{\bar{x}, \bar{k}} \times \mathfrak{C}_1,\end{aligned}$$

where the auxiliary variables  $\mathbf{R}_{1,1}, \dots, \mathbf{R}_{5,1}, \mathbf{t}_{1,1}, \mathbf{t}_{2,1}$ , and  $\mathfrak{C}_1$  are given by the recursive formulas in (70). Therefore, since the CCG parameters of  $\hat{\mathcal{X}}_{\bar{k}}$  can be written as in (69), and one iteration of (64) produces a CCG with parameters of the same form, the result follows by induction. ■

**Remark 7.** For simplicity, in (66) and (67), we only consider CCGs with no equality constraints, but the result can be readily extended to CCGs with equality constraints in (66) and (67). Nevertheless, note that the equality constraints can always be eliminated by explicitly writing their solutions and modifying the remaining CCG parameters accordingly.

As shown in the previous result, the recursive formulas in (70) only involve fixed parameters, meaning that the variables  $\mathbf{R}_{1,N}, \dots, \mathbf{R}_{5,N}, \mathbf{t}_{1,N}, \mathbf{t}_{2,N}$ , and  $\mathfrak{C}_N$  can be precomputed for a given horizon  $N$ . Using these precomputed parameters, the state estimate  $\hat{\mathcal{X}}_k$  at each sampling instant  $t_k \geq NT_s$  can then be explicitly computed from the conservative estimate  $\bar{\mathcal{X}}_{k-N}$  using the expressions in (69). During the initial offset phase, when  $t_k < NT_s$ , a plausible approach is to obtain the current estimate directly from the conservative estimator. Algorithms 2 and 3 outline the proposed finite-horizon scheme.

---

**Algorithm 2** Precomputation of the Estimator Parameters

---

**Require:**  $\Phi(T_s), \Gamma(T_s), \mathbf{C}, \mathbf{G}_{\bar{w}}, \mathbf{c}_{\bar{w}}, \mathfrak{G}_{\bar{w}}, \mathbf{G}_v, \mathbf{c}_v, \mathfrak{G}_v, N$

- 1:  $\mathbf{R}_{1,0} \leftarrow \mathbf{I}, \mathbf{R}_{2,0} \leftarrow \mathbf{0}, \mathbf{R}_{3,0} \leftarrow \mathbf{C}, \mathbf{R}_{4,0} \leftarrow \mathbf{G}_v,$
- 2:  $\mathbf{R}_{5,0} \leftarrow -\mathbf{C}, \mathbf{t}_{1,0} \leftarrow \mathbf{0}, \mathbf{t}_{2,0} \leftarrow -\mathbf{c}_v, \mathfrak{C}_0 \leftarrow \mathfrak{G}_v$
- 3: **for**  $l \leftarrow 0$  to  $N-1$  **do**
- 4:     compute  $\mathbf{R}_{1,l+1}, \dots, \mathbf{R}_{5,l+1}, \mathbf{t}_{1,l+1}, \mathbf{t}_{2,l+1}, \mathfrak{C}_{l+1}$  (70)
- 5: **end for**
- 6: **return**  $\mathbf{R}_{1,N}, \dots, \mathbf{R}_{5,N}, \mathbf{t}_{1,N}, \mathbf{t}_{2,N}, \mathfrak{C}_N$

---

**Algorithm 3** Explicit Finite-Horizon Estimator

---

**Require:**  $\mathbf{R}_{1,N}, \dots, \mathbf{R}_{5,N}, \mathbf{t}_{1,N}, \mathbf{t}_{2,N}, \mathfrak{C}_N, \mathbf{y}_{0:k}, k, N$

- 1: **if**  $k < N$  **then**
- 2:     compute  $\hat{\mathcal{X}}_k$  using conservative estimator
- 3:      $\hat{\mathcal{X}}_k \leftarrow \bar{\mathcal{X}}_k$
- 4: **else**
- 5:     compute  $\bar{\mathcal{X}}_{k-N}$  using conservative estimator
- 6:     compute  $\hat{\mathcal{X}}_k$  using (68)-(69)
- 7: **end if**
- 8: **return**  $\hat{\mathcal{X}}_k$

---

Regarding the conservative estimator, a practical choice is to consider an ellipsoidal observer derived from a Luenberger observer, as suggested in [28]. However, alternative approaches are also viable, provided that they produce CCG estimates with a fixed-length representation.

## VI. CONTROL BARRIER FUNCTIONS FOR CONSTRAINED CONVEX GENERATORS

As detailed in Section IV, given a guaranteed estimate  $\hat{\mathcal{X}}_k$  of an obstacle state at the sampling instant  $t_k$ , the evolution of the estimated obstacle set during the sampling interval  $[t_k, t_{k+1})$  can be described by the set-valued flow  $\hat{\mathcal{O}}_k^+ : [t_k, t_{k+1}) \Rightarrow \mathbb{R}^p$ , which is defined for all  $t \in [t_k, t_{k+1})$  as

$$\hat{\mathcal{O}}_k^+(t) = \mathbf{E} \left( \Phi(t - t_k) \hat{\mathcal{X}}_k \oplus \Gamma(t - t_k) \tilde{\mathcal{W}} \right) \oplus \bar{\mathcal{O}}^+, \quad (74)$$

where the obstacle index  $i$  has been omitted for brevity. Following the state estimation approach described in the previous section, the sets  $\tilde{\mathcal{W}}$  and  $\hat{\mathcal{X}}_k$  are CCGs, defined by (66) and (68). Thus, if the body set  $\bar{\mathcal{O}}^+$  is also modeled as a CCG, then the estimated obstacle set  $\hat{\mathcal{O}}_k^+(t)$  is a CCG and can be expressed in closed form using the set-operation identities from (63).

More specifically, let the body set be defined as

$$\bar{\mathcal{O}}^+ = (\mathbf{G}_{\bar{o}}^+, \mathbf{c}_{\bar{o}}^+, [\ ], [\ ], \mathfrak{G}_{\bar{o}}^+). \quad (75)$$

Then, by applying the identities from (63) to (74), we conclude that the estimated obstacle set at each time  $t \in [t_k, t_{k+1})$  is a CCG of the form

$$\hat{\mathcal{O}}_k^+(t) = \left( \mathbf{G}_{\hat{o},k}^+(t), \mathbf{c}_{\hat{o},k}^+(t), \mathbf{A}_{\hat{o},k}^+, \mathbf{b}_{\hat{o},k}^+, \mathfrak{G}_{\hat{o},k}^+ \right), \quad (76)$$

where the CCG parameters are given by

$$\begin{aligned}\mathbf{G}_{\hat{o},k}^+(t) &= [\mathbf{E} \Phi(t - t_k) \mathbf{G}_{\hat{x},k} \ \mathbf{E} \Gamma(t - t_k) \mathbf{G}_{\bar{w}} \ \mathbf{G}_{\bar{o}}^+], \\ \mathbf{c}_{\hat{o},k}^+(t) &= \mathbf{E} \Phi(t - t_k) \mathbf{c}_{\hat{x},k} + \mathbf{E} \Gamma(t - t_k) \mathbf{c}_{\bar{w}} + \mathbf{c}_{\bar{o}}^+, \\ \mathbf{A}_{\hat{o},k}^+ &= [\mathbf{A}_{\hat{x},k} \ \mathbf{0} \ \mathbf{0}], \\ \mathbf{b}_{\hat{o},k}^+ &= \mathbf{b}_{\hat{x},k}, \\ \mathfrak{G}_{\hat{o},k}^+ &= \mathfrak{G}_{\hat{x},k} \times \mathfrak{G}_{\bar{w}} \times \mathfrak{G}_{\bar{o}}^+,\end{aligned} \quad (77)$$

from which it becomes clear that the functions  $\mathbf{G}_{\delta,k}^+$  and  $\mathbf{c}_{\delta,k}^+$  are (infinitely) continuously differentiable. Moreover, since  $\mathcal{W}$ ,  $\hat{\mathcal{X}}_k$ , and  $\bar{\mathcal{O}}^+$  are compact sets, and the body set has a nonempty interior, it follows that the estimated obstacle set is a compact set with nonempty interior.

With the set-valued flow  $\hat{\mathcal{O}}_k^+$  in hand, the next step is then to construct a CBF that captures the safety objective of avoiding the respective obstacle. However, since the estimated obstacle set at each time instant is a CCG, we cannot directly convert  $\hat{\mathcal{O}}_k^+$  into a CBF. Accordingly, the subproblem addressed in this section is stated as follows.

**Subproblem 2** (CBFs for CCGs). Let  $\mathcal{O} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  be a set-valued flow defined for all  $t \in [t_0, t_f]$  as

$$\begin{aligned}\mathcal{O}(t) &= \{\mathbf{G}(t)\xi + \mathbf{c}(t) : \mathbf{A}\xi = \mathbf{b}, \xi \in \mathfrak{G}\} \\ &= (\mathbf{G}(t), \mathbf{c}(t), \mathbf{A}, \mathbf{b}, \mathfrak{G}),\end{aligned}\quad (78)$$

where  $\mathcal{O}(t)$  is a compact set with nonempty interior, the functions  $\mathbf{G} : [t_0, t_f] \rightarrow \mathbb{R}^{p \times \xi}$ ,  $\mathbf{c} : [t_0, t_f] \rightarrow \mathbb{R}^p$  are continuously differentiable,  $\mathbf{A} \in \mathbb{R}^{c \times \xi}$ ,  $\mathbf{b} \in \mathbb{R}^c$ , and  $\mathfrak{G} = \mathcal{G}_1 \times \mathcal{G}_2 \times \dots \times \mathcal{G}_G$ , where, for each  $j \in \mathcal{J} = \{1, \dots, G\}$ , the generator set  $\mathcal{G}_j$  is the zero-sublevel set of a convex function  $g_j : \mathbb{R}^{\xi_j} \rightarrow \mathbb{R}$ . Then, convert  $\mathcal{O}$  into a CBF  $h : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}$  for (15), with an associated safe set-valued flow  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  such that  $\mathcal{C}(t) \subseteq \mathbb{R}^p \setminus \mathcal{O}(t)$  for all  $t \in [t_0, t_f]$ .

The next subsection details the proposed conversion process, using the generic notation in the previous statement for clarity.

#### A. Conversion Procedure

The first step of the conversion process is to eliminate the linear equality constraint from (78). Since  $\mathcal{O}(t)$  has nonempty interior, this linear equation admits infinitely many solutions, which can be expressed in the parametric form

$$\xi = \mathbf{A}^\dagger \mathbf{b} + \mathbf{N}_A \eta, \quad (79)$$

where  $\mathbf{A}^\dagger \in \mathbb{R}^{\xi \times c}$  is the pseudoinverse of  $\mathbf{A}$ ,  $\mathbf{N}_A \in \mathbb{R}^{\xi \times \eta}$  is a matrix whose columns form an orthonormal basis for the null space of  $\mathbf{A}$ , and  $\eta \in \mathbb{R}^\eta$  is a vector of free parameters. As no assumptions are made about the rank of  $\mathbf{A}$ , its pseudoinverse is obtained through the Singular Value Decomposition (SVD):

$$\mathbf{A} = \mathbf{U}_A \Sigma_A \mathbf{V}_A^\top, \quad (80)$$

where the columns of  $\mathbf{U}_A$  and  $\mathbf{V}_A$  form an orthonormal basis for the column and row spaces of  $\mathbf{A}$ , respectively, and  $\Sigma_A$  is a diagonal matrix with the singular values on its entries. From this decomposition, the pseudoinverse follows as

$$\mathbf{A}^\dagger = \mathbf{V}_A \Sigma_A^{-1} \mathbf{U}_A^\top. \quad (81)$$

Using the previous parameterization, the set  $\mathcal{O}(t)$  can then be equivalently expressed with no equality constraint as

$$\mathcal{O}(t) = (\mathbf{G}(t)\mathbf{N}_A, \mathbf{c}(t) + \mathbf{G}(t)\mathbf{A}^\dagger \mathbf{b}, [\cdot], [\cdot], \mathfrak{G}'). \quad (82)$$

Here, the new generator set  $\mathfrak{G}'$  is given by

$$\mathfrak{G}' = \{\eta \in \mathbb{R}^\eta : f_j(\eta) \leq 0 \text{ for all } j \in \mathcal{J}\}, \quad (83)$$

where each function  $f_j : \mathbb{R}^\eta \rightarrow \mathbb{R}$  is defined as

$$f_j(\eta) = g_j(\mathbf{S}_j \mathbf{A}^\dagger \mathbf{b} + \mathbf{S}_j \mathbf{N}_A \eta) \quad (84)$$

for all  $\eta \in \mathbb{R}^\eta$ , where  $\mathbf{S}_j \in \mathbb{R}^{\xi_j \times \xi}$  is an auxiliary matrix that

selects the  $j$ th individual generator vector  $\xi_j$  from the overall generator vector  $\xi = (\xi_1, \dots, \xi_G)$ . Moreover, since  $\mathcal{O}(t)$  has nonempty interior, the matrix  $\mathbf{G}(t)\mathbf{N}_A$  has full row rank.

Observe now that the generator set  $\mathfrak{G}'$  can be equivalently expressed using the maximum function as

$$\mathfrak{G}' = \left\{ \eta \in \mathbb{R}^\eta : \max_{j \in \mathcal{J}} f_j(\eta) \leq 0 \right\}. \quad (85)$$

However, since the maximum function is not differentiable, it cannot be directly used in the CBF design. To overcome this, we apply a smooth underapproximation of the maximum function, adopting the LogSumExp approach once more from [62]. Specifically, we define a set-valued flow  $\tilde{\mathcal{O}} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  as

$$\tilde{\mathcal{O}}(t) = \left( \tilde{\mathbf{G}}(t), \tilde{\mathbf{c}}(t), [\cdot], [\cdot], \tilde{\mathfrak{G}} \right) \quad (86)$$

for all  $t \in [t_0, t_f]$ , with parameters given by

$$\begin{aligned}\tilde{\mathbf{G}}(t) &= \mathbf{G}(t)\mathbf{N}_A, \\ \tilde{\mathbf{c}}(t) &= \mathbf{c}(t) + \mathbf{G}(t)\mathbf{A}^\dagger \mathbf{b}, \\ \tilde{\mathfrak{G}} &= \{\eta \in \mathbb{R}^\eta : f(\eta) \leq 0\},\end{aligned}\quad (87)$$

where the function  $f : \mathbb{R}^\eta \rightarrow \mathbb{R}$  is defined for all  $\eta \in \mathbb{R}^\eta$  as

$$f(\eta) = \frac{1}{\gamma} \ln \left( \sum_{j \in \mathcal{J}} \exp(\gamma f_j(\eta)) \right) - \frac{\ln(G+1)}{\gamma}, \quad (88)$$

with smoothing parameter  $\gamma \in \mathbb{R}_{>0}$ . This construction ensures that, for all  $\eta \in \mathbb{R}^\eta$ , we have

$$f(\eta) < \max_{j \in \mathcal{J}} f_j(\eta), \quad (89)$$

implying that  $\mathcal{O}(t) \subset \text{int}(\tilde{\mathcal{O}}(t))$  for all  $t \in [t_0, t_f]$ , such that

$$\lim_{\gamma \rightarrow \infty} \tilde{\mathcal{O}}(t) = \mathcal{O}(t). \quad (90)$$

Moreover, note that  $f$  is convex, as each function  $f_j$  is convex and  $f$  is defined through convexity-preserving operations.

Since the goal is to obtain a CBF for the agent dynamics in (15), we must now express the set  $\tilde{\mathcal{O}}(t)$  in terms of the agent's position. To this end, note that a point  $\mathbf{p} \in \mathbb{R}^p$  belongs to  $\tilde{\mathcal{O}}(t)$  if and only if there exists some  $\eta \in \mathbb{R}^\eta$  so that  $f(\eta) \leq 0$  and  $\tilde{\mathbf{G}}(t)\eta + \tilde{\mathbf{c}}(t) = \mathbf{p}$ . Formally, this means that

$$\begin{aligned}\tilde{\mathcal{O}}(t) &= \{\mathbf{p} \in \mathbb{R}^p : \exists \eta \in \mathbb{R}^\eta : f(\eta) \leq 0, \\ &\quad \tilde{\mathbf{G}}(t)\eta + \tilde{\mathbf{c}}(t) = \mathbf{p}\}.\end{aligned}\quad (91)$$

Consequently, this set can be equivalently written using a CBF candidate  $h : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}$  as

$$\tilde{\mathcal{O}}(t) = \{\mathbf{p} \in \mathbb{R}^p : h(\mathbf{p}, t) \leq 0\}, \quad (92)$$

where  $h$  is defined via the following optimization problem:

$$\begin{aligned}h(\mathbf{p}, t) &= \min_{\eta \in \mathbb{R}^\eta} f(\eta) \\ \text{subject to } \tilde{\mathbf{G}}(t)\eta + \tilde{\mathbf{c}}(t) &= \mathbf{p},\end{aligned}\quad (93)$$

provided that (93) is solvable for all  $(\mathbf{p}, t) \in \mathbb{R}^p \times [t_0, t_f]$ . The next result now provides a mild sufficient condition for  $h$  to qualify as a valid CBF for the agent dynamics in (15).

**Theorem 3.** Let  $\tilde{\mathcal{O}} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  be a set-valued flow defined as in (92), where  $\tilde{\mathcal{O}}(t)$  has nonempty interior for all  $t \in [t_0, t_f]$ , and  $h : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}$  is defined by (93), where the functions  $\tilde{\mathbf{G}} : [t_0, t_f] \rightarrow \mathbb{R}^{p \times \eta}$ ,  $\tilde{\mathbf{c}} : [t_0, t_f] \rightarrow \mathbb{R}^p$  are continuously

differentiable, and  $\tilde{\mathbf{G}}(t)$  has full row rank for all  $t \in [t_0, t_f]$ . If  $f : \mathbb{R}^\eta \rightarrow \mathbb{R}$  is twice continuously differentiable and strictly convex, then  $h$  is a CBF for the system (15).

*Proof.* Let  $\mathcal{C} : [t_0, t_f] \rightrightarrows \mathbb{R}^p$  be the set-valued flow associated with  $h$ , defined for all  $t \in [t_0, t_f]$  as  $\mathcal{C}(t) = \mathbb{R}^p \setminus \text{int}(\tilde{\mathcal{O}}(t))$ . To show that  $h$  is a CBF for (15), we have to verify that it satisfies the following properties: (i)  $h$  is continuously differentiable, (ii)  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) \neq \mathbf{0}$  when  $h(\mathbf{p}, t) = 0$ , and (iii) there exists an extended class- $\mathcal{K}_\infty$  function  $\alpha$  such that

$$\sup_{\mathbf{z} \in \mathbb{R}^z} \dot{h}(\mathbf{p}, t) > -\alpha(h(\mathbf{p}, t)) \quad (94)$$

for at least all  $(\mathbf{p}, t) \in \mathcal{G}(\mathcal{C})$ .

Since the optimization problem in (93) is convex, its solutions coincide with those of the respective KKT conditions. In this case, the KKT system follows as

$$\psi(\mathbf{p}, t, \boldsymbol{\eta}, \boldsymbol{\lambda}) := \begin{bmatrix} \left( \frac{\partial f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right)^\top + \tilde{\mathbf{G}}(t)^\top \boldsymbol{\lambda} \\ \tilde{\mathbf{G}}(t)\boldsymbol{\eta} + \tilde{\mathbf{c}}(t) - \mathbf{p} \end{bmatrix} = \mathbf{0}, \quad (95)$$

where  $\boldsymbol{\lambda} \in \mathbb{R}^p$  is the Lagrange multiplier associated with the equality constraint. From this setup, the partial derivatives of  $\psi$  with respect to  $(\mathbf{p}, t)$  and  $(\boldsymbol{\eta}, \boldsymbol{\lambda})$  are given by

$$\begin{aligned} \frac{\partial \psi(\mathbf{p}, t, \boldsymbol{\eta}, \boldsymbol{\lambda})}{\partial (\mathbf{p}, t)} &= \begin{bmatrix} \mathbf{0} & \dot{\tilde{\mathbf{G}}}(t)^\top \boldsymbol{\lambda} \\ -\mathbf{I} & \dot{\tilde{\mathbf{G}}}(t)\boldsymbol{\eta} + \dot{\tilde{\mathbf{c}}}(t) \end{bmatrix}, \\ \frac{\partial \psi(\mathbf{p}, t, \boldsymbol{\eta}, \boldsymbol{\lambda})}{\partial (\boldsymbol{\eta}, \boldsymbol{\lambda})} &= \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}^2} & \tilde{\mathbf{G}}(t)^\top \\ \tilde{\mathbf{G}}(t) & \mathbf{0} \end{bmatrix}. \end{aligned} \quad (96)$$

Applying now the Schur complement to the latter yields

$$\begin{aligned} \det \left( \frac{\partial \psi(\mathbf{p}, t, \boldsymbol{\eta}, \boldsymbol{\lambda})}{\partial (\boldsymbol{\eta}, \boldsymbol{\lambda})} \right) &= \\ - \det \left( \frac{\partial^2 f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}^2} \right) \det \left( \tilde{\mathbf{G}}(t) \left( \frac{\partial^2 f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}^2} \right)^{-1} \tilde{\mathbf{G}}(t)^\top \right). \end{aligned} \quad (97)$$

As  $f$  is strictly convex by assumption, meaning that the Hessian matrix of  $f$  is positive definite for all  $\boldsymbol{\eta} \in \mathbb{R}^\eta$ , and  $\tilde{\mathbf{G}}(t)$  has full row rank for all  $t \in [t_0, t_f]$ , the previous determinant is always nonzero. Therefore, by the Implicit Function Theorem, the KKT system defines  $(\boldsymbol{\eta}, \boldsymbol{\lambda})$  as a function of  $(\mathbf{p}, t)$ , i.e.,

$$(\boldsymbol{\eta}, \boldsymbol{\lambda}) = \ell(\mathbf{p}, t), \quad (98)$$

where the implicit function  $\ell : \mathbb{R}^p \times [t_0, t_f] \rightarrow \mathbb{R}^\eta \times \mathbb{R}^p$  is continuously differentiable, with derivative given by

$$\frac{\partial \ell(\mathbf{p}, t)}{\partial (\mathbf{p}, t)} = - \left[ \left( \frac{\partial \psi(\cdot)}{\partial (\boldsymbol{\eta}, \boldsymbol{\lambda})} \right)^{-1} \frac{\partial \psi(\cdot)}{\partial (\mathbf{p}, t)} \right]_{(\boldsymbol{\eta}, \boldsymbol{\lambda})=\ell(\mathbf{p}, t)}. \quad (99)$$

Consequently,  $h$  can be equivalently expressed as

$$h(\mathbf{p}, t) = f(\mathbf{E}_{\boldsymbol{\eta}} \ell(\mathbf{p}, t)), \quad (100)$$

where  $\mathbf{E}_{\boldsymbol{\eta}} \in \mathbb{R}^{\eta \times (\eta+p)}$  extracts  $\boldsymbol{\eta}$  from  $(\boldsymbol{\eta}, \boldsymbol{\lambda})$ , from which it becomes clear that  $h$  is continuously differentiable, with

$$\frac{\partial h(\mathbf{p}, t)}{\partial (\mathbf{p}, t)} = \frac{\partial f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \Big|_{\boldsymbol{\eta}=\mathbf{E}_{\boldsymbol{\eta}} \ell(\mathbf{p}, t)} \mathbf{E}_{\boldsymbol{\eta}} \frac{\partial \ell(\mathbf{p}, t)}{\partial (\mathbf{p}, t)}, \quad (101)$$

hence proving (i).

Finally, we observe that  $h$  is convex in  $\mathbf{p}$ , as the epigraph of  $h(\cdot, t)$  is the projection of the set

$$\begin{aligned} \mathcal{S}(t) &= \{(\mathbf{p}, s, \boldsymbol{\eta}) \in \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^\eta : f(\boldsymbol{\eta}) \leq s, \\ &\quad \tilde{\mathbf{G}}(t)\boldsymbol{\eta} + \tilde{\mathbf{c}}(t) = \mathbf{p}\} \end{aligned} \quad (102)$$

in the  $(\mathbf{p}, s)$ -space, which is a convex set since  $\mathcal{S}(t)$  is convex. Thus, as  $h$  is convex in  $\mathbf{p}$  and  $\tilde{\mathcal{O}}(t)$  has nonempty interior, it follows that  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) \neq \mathbf{0}$  when  $h(\mathbf{p}, t) \geq 0$ . Consequently, for any extended class- $\mathcal{K}_\infty$  function  $\alpha$ , the inequality in (94) holds for all  $(\mathbf{p}, t) \in \mathcal{G}(\mathcal{C})$ , as  $h$  can be made arbitrarily large when  $\frac{\partial}{\partial \mathbf{p}} h(\mathbf{p}, t) \neq \mathbf{0}$ . Hence, the properties (ii) and (iii) are also satisfied, which completes the proof. ■

The preceding result establishes a mild sufficient condition under which the function  $h$ , defined in (93), qualifies as a CBF for the agent dynamics in (15). Particularly, if each function  $g_j$  is twice continuously differentiable and strictly convex, then  $f$  inherits these properties and, by Theorem 3,  $h$  constitutes a valid CBF for (15). To verify this claim, observe that

$$\begin{aligned} \left( \frac{\partial f_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right)^\top &= \mathbf{N}_A^\top \mathbf{S}_j^\top \frac{\partial g_j(\boldsymbol{\xi}_j)}{\partial \boldsymbol{\xi}_j} \Big|_{\boldsymbol{\xi}_j=\mathbf{s}_j(\boldsymbol{\eta})}, \\ \frac{\partial^2 f_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}^2} &= \mathbf{N}_A^\top \mathbf{S}_j^\top \frac{\partial^2 g_j(\boldsymbol{\xi}_j)}{\partial \boldsymbol{\xi}_j^2} \Big|_{\boldsymbol{\xi}_j=\mathbf{s}_j(\boldsymbol{\eta})} \mathbf{S}_j \mathbf{N}_A, \end{aligned} \quad (103)$$

where  $\mathbf{s}_j(\boldsymbol{\eta}) = \mathbf{S}_j \mathbf{A}^\dagger \mathbf{b} + \mathbf{S}_j \mathbf{N}_A \boldsymbol{\eta}$ , and also note that

$$\frac{\partial f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} = \sum_{j \in \mathcal{J}} \pi_j(\boldsymbol{\eta}) \frac{\partial f_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}, \quad (104)$$

where each weight is defined as

$$\pi_j(\boldsymbol{\eta}) = \exp(\gamma(f_j(\boldsymbol{\eta}) - f(\boldsymbol{\eta})) - \ln(G+1)). \quad (105)$$

Consequently, the Hessian matrix of  $f$  follows as

$$\begin{aligned} \frac{\partial^2 f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}^2} &= \mathbf{N}_A^\top \text{diag} \left( \sum_{j \in \mathcal{J}} \pi_j(\boldsymbol{\eta}) \frac{\partial^2 g_j(\boldsymbol{\xi}_j)}{\partial \boldsymbol{\xi}_j^2} \Big|_{\boldsymbol{\xi}_j=\mathbf{s}_j(\boldsymbol{\eta})} \right) \mathbf{N}_A \\ &\quad + \gamma \sum_{j \in \mathcal{J}} \pi_j(\boldsymbol{\eta}) \left( \frac{\partial(f_j(\boldsymbol{\eta}) - f(\boldsymbol{\eta}))}{\partial \boldsymbol{\eta}} \right)^\top \frac{\partial(f_j(\boldsymbol{\eta}) - f(\boldsymbol{\eta}))}{\partial \boldsymbol{\eta}}. \end{aligned} \quad (106)$$

From this expression, it becomes clear that the Hessian matrix  $\frac{\partial^2}{\partial \boldsymbol{\eta}^2} f(\boldsymbol{\eta})$  is always positive definite when each  $g_j$  is strictly convex, as the first term is always positive definite under strict convexity of each  $g_j$ , while the second term is always positive semi-definite, which confirms the initial claim.

**Remark 8** (Implementation Details). In general, the implicit function  $\ell$  introduced in the proof of Theorem 3, and therefore the CBF  $h$  defined by (93), cannot be expressed in closed form. However, under the conditions of Theorem 3, the CBF  $h$  can be locally approximated using a first-order Taylor expansion. For instance, the first-order Taylor approximation of  $h$  around  $(\mathbf{p}_0, t_0)$  is given by

$$h(\mathbf{p}, t) \simeq h(\mathbf{p}_0, t_0) + \frac{\partial h(\mathbf{p}, t)}{\partial (\mathbf{p}, t)} \Big|_{(\mathbf{p}, t)=(\mathbf{p}_0, t_0)} (\mathbf{p} - \mathbf{p}_0, t - t_0), \quad (107)$$

where the value of  $h$  and its gradient at  $(\mathbf{p}_0, t_0)$  are computed using (99)-(101) with  $\ell(\mathbf{p}_0, t_0) = (\boldsymbol{\eta}_0, \boldsymbol{\lambda}_0)$ , where the vector  $(\boldsymbol{\eta}_0, \boldsymbol{\lambda}_0)$  is obtained by numerically solving the KKT system

at  $(\mathbf{p}_0, t_0)$ . For example,  $(\boldsymbol{\eta}_0, \boldsymbol{\lambda}_0)$  can be obtained by performing the following steps:

$$\begin{aligned}\boldsymbol{\alpha}_0 &= \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^{n-p}} f\left(\tilde{\mathbf{G}}(t_0)^\dagger (\mathbf{p}_0 - \tilde{\mathbf{c}}(t_0)) + \mathbf{N}_{\tilde{\mathbf{G}}(t_0)} \boldsymbol{\alpha}\right), \\ \boldsymbol{\eta}_0 &= \tilde{\mathbf{G}}(t_0)^\dagger (\mathbf{p}_0 - \tilde{\mathbf{c}}(t_0)) + \mathbf{N}_{\tilde{\mathbf{G}}(t_0)} \boldsymbol{\alpha}_0, \\ \boldsymbol{\lambda}_0 &= \left(\tilde{\mathbf{G}}(t_0)^\dagger\right)^\top \left(\frac{\partial f(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \Big|_{\boldsymbol{\eta}=\boldsymbol{\eta}_0}\right)^\top,\end{aligned}\quad (108)$$

where the pseudoinverse of  $\tilde{\mathbf{G}}(t_0)$  can be computed as

$$\tilde{\mathbf{G}}(t_0)^\dagger = \tilde{\mathbf{G}}(t_0)^\top \left(\tilde{\mathbf{G}}(t_0) \tilde{\mathbf{G}}(t_0)^\top\right)^{-1} \quad (109)$$

because  $\tilde{\mathbf{G}}(t_0)$  has full row rank, and  $\mathbf{N}_{\tilde{\mathbf{G}}(t_0)} \in \mathbb{R}^{n \times (n-p)}$  is a matrix whose columns form an orthonormal basis for the null space of  $\tilde{\mathbf{G}}(t_0)$ . The first step (computation of  $\boldsymbol{\alpha}_0$ ) involves solving an unconstrained convex optimization problem, which can be addressed efficiently using appropriate solvers.

While the first-order approximation is adequate for a small sampling period, higher-order expansions can be constructed if  $h$  has a higher smoothness degree [65]. These arise by further differentiating (101) and (99), and then evaluating the resulting expressions at  $(\mathbf{p}_0, t_0)$  with  $\ell(\mathbf{p}_0, t_0) = (\boldsymbol{\eta}_0, \boldsymbol{\lambda}_0)$ .

**Remark 9** (Ensuring Global Safety). As discussed in Section IV, global safety is guaranteed if, at every sampling time  $t_k$ , the condition  $\mathbf{p}_k \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t_k))$  holds. If we employ the infinite-horizon estimation scheme described at the beginning of Section V, then this condition is preserved at all sampling instants if it holds initially at  $t = 0$ . This follows from two facts: (i) if at the time  $t_k$  we have  $\mathbf{p}_k \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t_k))$ , then the trajectory of the agent satisfies  $\varphi(t) \in \text{int}(\cap_{i \in \mathcal{I}} \mathcal{C}_{i,k}(t))$  for all  $t \in [t_k, t_{k+1}]$  (ensured by the strict inequality in (34)), and (ii) the infinite-horizon estimator, along with the CCG-to-CBF conversion process, ensures that  $\mathcal{C}_{i,k}(t_{k+1}^-) \subseteq \mathcal{C}_{i,k+1}(t_{k+1})$  for all  $i \in \mathcal{I}$ . However, using the finite-horizon estimator to keep a fixed computational load means that  $\mathbf{p}_{k+1} \in \text{int}(\mathcal{C}_{i,k}(t_{k+1}^-))$  does not necessarily imply that  $\mathbf{p}_{k+1} \in \text{int}(\mathcal{C}_{i,k+1}(t_{k+1}))$ , due to the conservatism introduced by discarding the measurement from time  $t_{k-N}$ . Several alternatives are possible to still use the finite-horizon estimator while maintaining theoretical guarantees of safety, such as rejecting the new estimate produced by this estimator whenever the condition is not satisfied and continuing the propagation of the estimate from  $t_{k+1}^-$  (before the update) over the next interval. Nevertheless, we note that the finite-horizon estimator in [28], which uses a Luenberger observer as the conservative estimator, can be designed using a *deadbeat* gain to remove the conservatism in  $n_i$  time steps (under an observability assumption), as shown in [66] for fault detection. Overall, this discussion highlights the importance for obstacle estimates to never contain the agent position to preserve theoretical guarantees of global safety.

## VII. SIMULATION RESULTS

This section presents three simulation examples that demonstrate the framework proposed in this paper. The first example considers an agent navigating in a known static environment and serves to illustrate different problem geometries and the CCG-to-CBF conversion procedure. The remaining two exam-

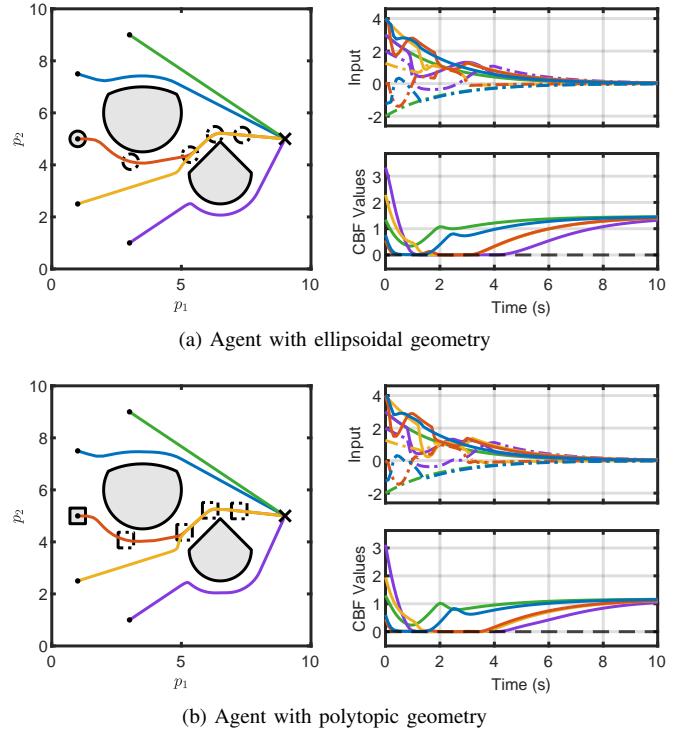


Fig. 1. Navigation of an agent with single-integrator dynamics and ellipsoidal and polytopic geometry around obstacles with mixed ellipsoidal and polytopic geometry. The left-hand plots show trajectories from different initial positions, and the right-hand plots illustrate the respective time evolution of the control inputs and overall CBF values. The colors in the right-hand plots correspond to those of the associated trajectories on the left-hand plots. In the input plots, solid lines represent the first input component and dotted lines the second.

ples involve dynamic obstacles with uncertain linear dynamics and demonstrate the overall framework proposed in the paper.

**Example 1.** Fig. 1 exemplifies the safe navigation of a rigid-body agent in a known static environment with two obstacles, where both the agent and obstacle body sets are CCGs. Two different agent geometries are considered: an ellipsoidal agent, shown in Fig. 1 (a), and a polytopic agent, illustrated in Fig. 1 (b). In this example, the agent has single-integrator dynamics:

$$\dot{\mathbf{p}} = \mathbf{z}, \quad (110)$$

with  $\mathbf{p}, \mathbf{z} \in \mathbb{R}^2$ , and the agent is controlled using the approach proposed in this paper, without the estimation component, to safely guide it to a goal point  $\bar{\mathbf{p}} \in \mathbb{R}^2$ . Convergence to the goal is captured by a nominal controller  $\mathbf{k}_d : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , defined as

$$\mathbf{k}_d(\mathbf{p}) = K(\mathbf{p} - \bar{\mathbf{p}}) \quad (111)$$

for all  $\mathbf{p} \in \mathbb{R}^2$ , where  $K$  is a negative gain.

The obstacles are modeled as CCGs with mixed geometry: one of them is defined as the intersection of a polytope with an ellipsoid, and the other is formed by the intersection of two ellipsoids. An ellipsoid  $\mathcal{E} \subset \mathbb{R}^p$  is a CCG of the form

$$\mathcal{E} = (\mathbf{G}_e, \mathbf{c}_e, [\cdot], [\cdot], \mathfrak{G}_e), \quad (112)$$

where  $\mathfrak{G}_e$  is the unit  $\ell_2$ -ball in  $\mathbb{R}^p$ . The generator set  $\mathfrak{G}_e$  can be described as the zero-sublevel set of a function  $g_e : \mathbb{R}^p \rightarrow \mathbb{R}$ , defined for all  $\xi_e \in \mathbb{R}^p$  as

$$g_e(\xi_e) = \frac{1}{2} \|\xi_e\|^2 - \frac{1}{2}, \quad (113)$$

which is twice continuously differentiable and strictly convex.

A polytope (or CZ)  $\mathcal{Z} \subset \mathbb{R}^p$  is a CCG of the form

$$\mathcal{Z} = (\mathbf{G}_z, \mathbf{c}_z, \mathbf{A}_z, \mathbf{b}_z, \mathfrak{G}_z), \quad (114)$$

where the set  $\mathfrak{G}_z$  is the unit  $\ell_\infty$ -ball in  $\mathbb{R}^{\xi_z}$ . However, since the  $\ell_\infty$ -norm is not differentiable neither strictly convex, we cannot directly derive a single suitable generator function. To address this, we can decompose  $\mathfrak{G}_z$  as  $\mathfrak{G}_z = \mathcal{G}_z^{\xi_z} = \times_{j=1}^{\xi_z} \mathcal{G}_z$ , where  $\mathcal{G}_z$  is the unit  $\ell_2$ -ball in  $\mathbb{R}$ , which can be described as the zero-sublevel set of a function  $g_z : \mathbb{R} \rightarrow \mathbb{R}$  defined as

$$g_z(s) = \frac{1}{2}s^2 - \frac{1}{2}, \quad (115)$$

which is twice continuously differentiable and strictly convex.

The simulations are conducted with a sampling period of  $T_s = 0.1$  s. Over each sampling interval, the safe controller defined in (44) is applied to the agent, with design parameters  $\bar{\beta} = \gamma = 10$  and  $\alpha(s) = \bar{\alpha}s$  for all  $s \in \mathbb{R}$ , with  $\bar{\alpha} = 10$ . The obstacle-specific CBFs are approximated as in Remark 8, by using a first-order Taylor expansion around the position of the agent at each sampling instant. The unconstrained optimization step in (108) is solved using a built-in quasi-Newton method in MATLAB, with an average computation time of 3 ms per obstacle per sampling step. As shown in Fig. 1, collision-free motion is achieved for all the initial agent positions, confirmed by the nonnegativity of the overall CBF values over time. The conservativeness of the resulting trajectories depends on the smoothing parameters  $\bar{\beta}$  and  $\gamma$ , as well as the decay rate  $\bar{\alpha}$ .

**Example 2.** Fig. 2 exemplifies the navigation of an ellipsoidal agent in an environment cluttered with dynamic obstacles with uncertain linear dynamics, showcasing the overall framework proposed in this paper. In this example, the agent also has single-integrator dynamics and is guided toward a target point using the same nominal controller from Example 1.

The environment contains two obstacles with different geometries (one polytopic and one ellipsoidal), and each obstacle is known to evolve with an uncertain velocity. Specifically, the motion of obstacle  $i$  is governed by

$$\dot{\mathbf{o}}_i = \mathbf{w}_i, \quad (116)$$

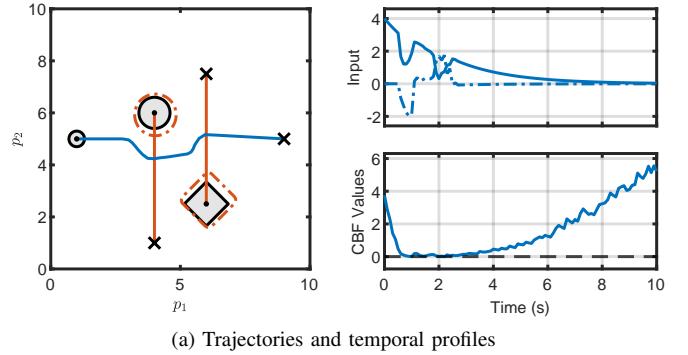
with  $\mathbf{o}_i, \mathbf{w}_i \in \mathbb{R}^2$ , where  $\mathbf{w}_i$  is an uncertain velocity, known to belong to an ellipsoidal set  $\mathcal{W} = (\mathbf{G}_w, \mathbf{0}, [ ], [ ], \mathfrak{G}_w) \subset \mathbb{R}^2$ . In the simulation, the actual velocities are constant and given by  $\mathbf{w}_1 = -\mathbf{w}_2 = [0 \ -0.5]^\top$ . The measurement model is

$$\mathbf{y}_{i,k} = \mathbf{o}_{i,k} + \mathbf{v}_{i,k}, \quad (117)$$

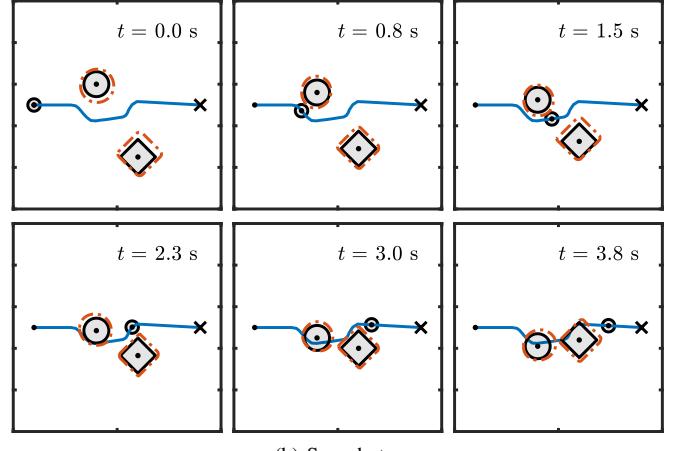
where the measurement noise  $\mathbf{v}_{i,k}$  is a random variable drawn uniformly from an ellipsoid  $\mathcal{V} = (\mathbf{G}_v, \mathbf{0}, [ ], [ ], \mathfrak{G}_v) \subset \mathbb{R}^2$ .

The simulation setup matches that of Example 1, with the additional parameters  $\mathbf{G}_w = 0.5\mathbf{I}$  and  $\mathbf{G}_v = 0.2\mathbf{I}$ . Moreover, the finite-horizon estimation scheme described in Section V is implemented with a horizon of  $N = 5$ , with the conservative estimator being defined as  $\bar{\mathcal{X}}_{i,k} = \mathbf{y}_{i,k} - \mathcal{V}$ . For the considered horizon length, the unconstrained optimization step in (108) takes an average of 5 ms per obstacle per sampling step.

As illustrated in Fig. 2 (a), collision-free motion is achieved throughout the simulation, confirmed by the nonnegative overall CBF values over time. Fig. 2 (b) shows snapshots of the agent and the estimated obstacle sets at selected time instants. The orange dotted contours denote the estimated CCG obstacle



(a) Trajectories and temporal profiles



(b) Snapshots

Fig. 2. Safe navigation of an ellipsoidal agent with single-integrator dynamics around moving obstacles with uncertain linear dynamics. The left-hand plot in subfigure (a) illustrates the agent and obstacle trajectories, and the right-hand plots shows the respective time evolution of the control input and overall CBF values. In the input plot, the solid line represents the first input component and the dotted line the second. Subfigure (b) presents snapshots of the agent and obstacle configurations at selected time instants. The orange dotted contours illustrate the estimated CCG obstacle sets used for collision avoidance.

sets, which enclose the actual obstacle sets. During the first  $N$  sampling steps, the state estimates are obtained directly from the conservative estimator, resulting in looser estimates. After the initial transient, the estimator produces steadier and tighter estimates, with the degree of tightness being determined by the sets  $\mathcal{W}$  and  $\mathcal{V}$ , as well as the horizon length  $N$ .

**Example 3.** Finally, Fig. 3 illustrates a scenario similar to that from Example 2, but where now both the agent and obstacles have second-order dynamics. Specifically, the agent dynamics are inspired by a satellite subject to a gravitational force and are described by the second-order strict-feedback system

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{z}, \\ \dot{\mathbf{z}} &= \mathbf{f}_1(\mathbf{p}) + \mathbf{u}, \end{aligned} \quad (118)$$

with  $\mathbf{p}, \mathbf{z}, \mathbf{u} \in \mathbb{R}^2$ , where  $\mathbf{f}_1(\mathbf{p}) = (\|\mathbf{p}\| + 0.1)^{-3} \mathbf{p}$  represents a gravity-like effect. Similar to the previous example, the agent is controlled using the overall approach proposed in this paper, with the addition of the backstepping-based design in Section IV-A, to safely guide it to a target point  $\bar{\mathbf{p}}$ . Convergence to the goal is captured by a nominal controller  $\mathbf{k}_{d,1} : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , defined for all  $(\mathbf{p}, \mathbf{z}) \in \mathbb{R}^2 \times \mathbb{R}^2$  as

$$\mathbf{k}_{d,1}(\mathbf{p}, \mathbf{z}) = -\mathbf{f}_1(\mathbf{p}) + K_1(\mathbf{z} - K(\mathbf{p} - \bar{\mathbf{p}})), \quad (119)$$

where  $K$  and  $K_1$  are negative gains.

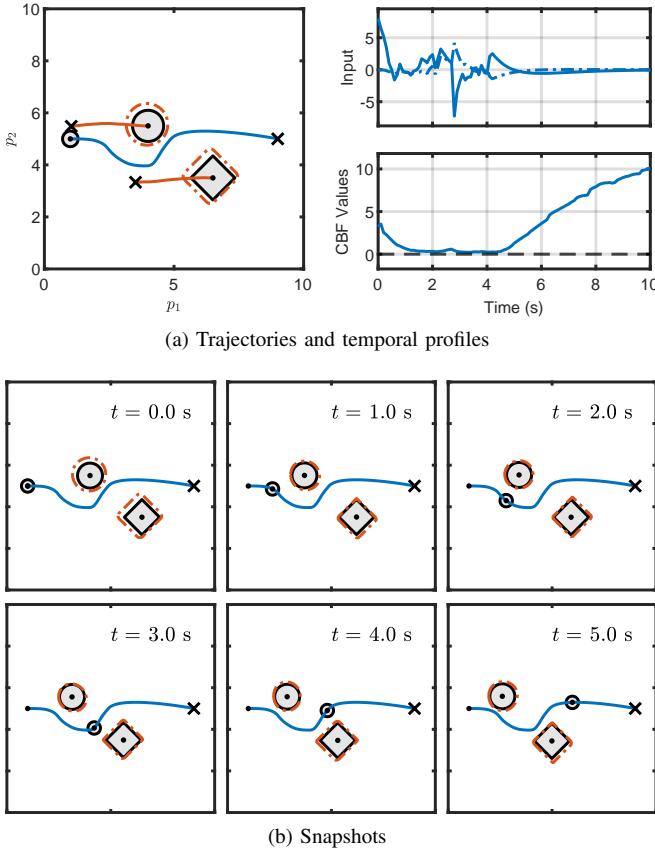


Fig. 3. Navigation of an ellipsoidal agent with second-order strict-feedback dynamics around moving obstacles with uncertain linear dynamics. The left-hand plot in subfigure (a) illustrates the agent and obstacle trajectories, and the right-hand plots show the respective time evolution of the control input and top-level CBF values. In the input plot, the solid line represents the first input component and the dotted line the second. Subfigure (b) presents snapshots of the agent and obstacle configurations at selected time instants. The orange dotted contours illustrate the estimated CCG obstacle sets used for avoidance.

The environment is populated with two obstacles with the same geometries from Example 2, and each obstacle evolves subject to a random acceleration. Specifically, the motion of obstacle  $i$  is governed by

$$\begin{aligned}\dot{\mathbf{o}}_i &= \mathbf{z}_i, \\ \dot{\mathbf{z}}_i &= \mathbf{w}_i,\end{aligned}\quad (120)$$

with  $\mathbf{o}_i, \mathbf{z}_i, \mathbf{w}_i \in \mathbb{R}^2$ , where  $\mathbf{w}_i$  is a random variable drawn uniformly from an ellipsoid  $\mathcal{W} = (\mathbf{G}_w, \mathbf{0}, [\cdot], [\cdot], \mathfrak{G}_w) \subset \mathbb{R}^2$ . The measurement model is of the form

$$\mathbf{y}_{i,k} = (\mathbf{o}_{i,k}, \mathbf{z}_{i,k}) + \mathbf{v}_{i,k}, \quad (121)$$

where the measurement noise  $\mathbf{v}_{i,k}$  is a random variable drawn uniformly from an ellipsoid  $\mathcal{V} = (\mathbf{G}_v, \mathbf{0}, [\cdot], [\cdot], \mathfrak{G}_v) \subset \mathbb{R}^4$ .

The simulation setup matches that of the previous examples, with  $\mathbf{G}_w = 0.5\mathbf{I}_2$ ,  $\mathbf{G}_v = 0.2\mathbf{I}_4$ , the backstepping parameters  $\varsigma = 0.1$ ,  $\bar{\sigma} = 10$ , and  $\epsilon = 10$ , and the CBF rate  $\alpha_1(s) = \bar{\alpha}_1 s$  for all  $s \in \mathbb{R}$ , with  $\bar{\alpha}_1 = 10$ . The finite-horizon estimator is implemented with a horizon of  $N = 5$ , with the conservative estimator again defined as  $\bar{\mathcal{X}}_{i,k} = \mathbf{y}_{i,k} - \mathcal{V}$ . For the considered horizon length, the unconstrained optimization step in (108) takes an average of 8 ms per obstacle per sampling step.

As illustrated in Fig. 3 (a), collision-free motion is achieved throughout the simulation, confirmed by the nonnegative top-

level CBF values over time. Additionally, Fig. 3 (b) presents snapshots of the agent and the estimated CCG obstacle sets at selected time instants. As in Example 2, during the first  $N$  sampling steps, the state estimates are obtained directly from the conservative estimator, resulting in looser estimates. After this initial phase, the finite-horizon estimator yields steadier and tighter estimates. Also, extending to an agent with second-order dynamics results in smoother trajectories.

## VIII. CONCLUSION

This paper introduced a control strategy that combines CBF-based safety filtering with guaranteed state estimation based on CCGs for safe navigation around obstacles with uncertain linear dynamics. At each sampling instant, the approach consists in obtaining a CCG estimate of each obstacle using a finite-horizon estimator, and each estimate is then propagated over the sampling interval to obtain a CCG-valued flow describing the estimated obstacle evolution. To convert these CCG-valued flows into CBFs, we developed a procedure that enables this conversion, which ultimately produces a CBF by means of a convex optimization problem, whose validity is established by the Implicit Function Theorem. The resulting obstacle-specific CBFs are then combined into a single CBF using a smooth approximation of the minimum function, and the overall CBF is used to design a safety filter through the standard QP-based approach. Since CCGs support Minkowski sums, the proposed approach naturally handles rigid-body agents by treating the agent as a point and enlarging the obstacle sets with the agent geometry. While the main contribution is general, our analysis focused on agents with first-order control-affine dynamics and second-order strict-feedback dynamics, and we demonstrated the proposed approach through several simulation examples.

Future directions include extending the methodology to handle obstacles with uncertain nonlinear dynamics—potentially through ReLU neural network models and hybrid zonotopes—and employing the framework for safe cooperative navigation of multi-agent teams. Additionally, performing an experimental validation could be a valuable next step to gather real-world data supporting the effectiveness of the method.

## REFERENCES

- [1] R. A. García, L. Orihuela, P. Millán, F. R. Rubio, and M. G. Ortega, “Guaranteed estimation and distributed control of vehicle formations,” *International Journal of Control*, vol. 93, no. 11, pp. 2729–2742, 2020.
- [2] F. Rego and D. Silvestre, “Cooperative navigation using constrained convex generators,” *European Journal of Control*, vol. 81, p. 101163, 2025.
- [3] G. Bianchin, Y.-C. Liu, and F. Pasqualetti, “Secure navigation of robots in adversarial environments,” *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 1–6, 2019.
- [4] C. Combastel, “An extended zonotopic and gaussian kalman filter (ezgkf) merging set-membership and stochastic paradigms: Toward nonlinear filtering and fault detection,” *Annual Reviews in Control*, vol. 42, pp. 232–243, 2016.
- [5] M. Althoff and J. J. Rath, “Comparison of guaranteed state estimators for linear time-invariant systems,” *Automatica*, vol. 130, p. 109662, 2021.
- [6] D. Silvestre, “Comparison of recent advances in set-membership techniques: Application to state estimation, fault detection and collision avoidance,” *European Journal of Control*, p. 101161, 2024.
- [7] R. E. H. Thabet, T. Raissi, C. Combastel, D. Efimov, and A. Zolghadri, “An effective method to interval observer design for time-varying systems,” *Automatica*, vol. 50, no. 10, pp. 2677–2684, 2014.

- [8] W. Tang, Z. Wang, Y. Wang, T. Raïssi, and Y. Shen, "Interval estimation methods for discrete-time linear time-invariant systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4717–4724, 2019.
- [9] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis: internal approximation," *Systems & control letters*, vol. 41, no. 3, pp. 201–211, 2000.
- [10] F. L. Chernousko, "Ellipsoidal state estimation for dynamical systems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 63, no. 5-7, pp. 872–879, 2005.
- [11] C. Combastel, "A state bounding observer based on zonotopes," in *2003 European control conference (ECC)*, pp. 2589–2594, IEEE, 2003.
- [12] M. U. B. Niazi, A. Alanwar, M. S. Chong, and K. H. Johansson, "Resilient set-based state estimation for linear time-invariant systems using zonotopes," *European Journal of Control*, vol. 74, p. 100837, 2023.
- [13] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, "Constrained zonotopes: A new tool for set-based estimation and fault detection," *Automatica*, vol. 69, pp. 126–136, 2016.
- [14] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, "Stochastic and deterministic fault detection for randomized gossip algorithms," *Automatica*, vol. 78, pp. 46–60, 2017.
- [15] T. Alamo, J. M. Bravo, and E. F. Camacho, "Guaranteed state estimation by zonotopes," *Automatica*, vol. 41, no. 6, pp. 1035–1043, 2005.
- [16] F. Abdallah, A. Gning, and P. Bonnifait, "Box particle filtering for nonlinear state estimation using interval analysis," *Automatica*, vol. 44, no. 3, pp. 807–815, 2008.
- [17] A. A. Julius and G. J. Pappas, "Trajectory based verification using local finite-time invariance," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 223–236, Springer, 2009.
- [18] B. S. Rego, D. M. Raimondo, and G. V. Raffo, "Set-based state estimation of nonlinear systems using constrained zonotopes and interval arithmetic," in *2018 European Control Conference (ECC)*, pp. 1584–1589, IEEE, 2018.
- [19] J. Wan, S. Sharma, and R. Sutton, "Guaranteed state estimation for nonlinear discrete-time systems via indirectly implemented polytopic set computation," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4317–4322, 2018.
- [20] D. Silvestre, "Constrained convex generators: A tool suitable for set-based estimation with range and bearing measurements," *IEEE Control Systems Letters*, vol. 6, pp. 1610–1615, 2021.
- [21] D. Silvestre, "Accurate guaranteed state estimation for uncertain LPVs using constrained convex generators," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 4957–4962, IEEE, 2022.
- [22] D. Silvestre, "Set-valued estimators for uncertain linear parameter-varying systems," *Systems & Control Letters*, vol. 166, p. 105311, 2022.
- [23] D. Silvestre, "Exact set-valued estimation using constrained convex generators for uncertain linear systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9461–9466, 2023.
- [24] D. Silvestre, "Closed-form approximations for the minimal robust positively invariant set using constrained convex generators," *IFAC-PapersOnLine*, vol. 58, no. 25, pp. 126–131, 2024.
- [25] C. Combastel, "Zonotopes and kalman observers: Gain optimality under distinct uncertainty paradigms and robust convergence," *Automatica*, vol. 55, pp. 265–273, 2015.
- [26] S. Kousik, A. Dai, and G. X. Gao, "Ellipsotopes: Uniting ellipsoids and zonotopes for reachability analysis and fault detection," *IEEE Transactions on Automatic Control*, vol. 68, no. 6, pp. 3440–3452, 2022.
- [27] F. Rego and D. Silvestre, "A novel and efficient order reduction for both constrained convex generators and constrained zonotopes," *IEEE Transactions on Automatic Control*, 2025.
- [28] F. Rego and D. Silvestre, "Explicit computation of guaranteed state estimates using constrained convex generators," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 1400–1405, IEEE, 2024.
- [29] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE robotics and automation letters*, vol. 5, no. 4, pp. 6001–6008, 2020.
- [30] D. Silvestre and G. Ramos, "Model predictive control with collision avoidance for unknown environment," *IEEE Control Systems Letters*, 2023.
- [31] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmese, "Successive convexification of non-convex optimal control problems with state constraints," *IFAC-PapersOnline*, vol. 50, no. 1, pp. 4063–4069, 2017.
- [32] P. Taborda, H. Matias, D. Silvestre, and P. Lourenço, "Convex MPC and thrust allocation with deadband for spacecraft rendezvous," *IEEE Control Systems Letters*, 2024.
- [33] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [34] 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)*, pp. 3420–3431, IEEE, 2019.
- [35] H. Matias and D. Silvestre, "Hybrid Lyapunov and barrier function-based control with stabilization guarantees," *arXiv preprint arXiv:2504.09760*, 2025.
- [36] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.
- [37] M. H. Cohen, P. Ong, G. Bahati, and A. D. Ames, "Characterizing smooth safety filters via the implicit function theorem," *IEEE Control Systems Letters*, vol. 7, pp. 3890–3895, 2023.
- [38] M. Alyaseen, N. Atanasov, and J. Cortés, "Continuity and boundedness of minimum-norm CBF-safe controllers," *IEEE Transactions on Automatic Control*, 2025.
- [39] R. Hamatani and H. Nakamura, "Collision avoidance control of robot arm considering the shape of the target system," in *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1311–1316, IEEE, 2020.
- [40] H. Almubarak, K. Stachowicz, N. Sadegh, and E. A. Theodorou, "Safety embedded differential dynamic programming using discrete barrier states," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2755–2762, 2022.
- [41] P. Mestres, C. Nieto-Granda, and J. Cortés, "Distributed safe navigation of multi-agent systems using control barrier function-based controllers," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6760–6767, 2024.
- [42] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfè, and C. Secchi, "Safety barrier functions for human-robot interaction with industrial manipulators," in *2019 18th European Control Conference (ECC)*, pp. 2565–2570, IEEE, 2019.
- [43] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10954–10961, 2022.
- [44] K. Long, K. Tran, M. Leok, and N. Atanasov, "Safe stabilizing control for polygonal robots in dynamic elliptical environments," in *2024 American Control Conference (ACC)*, pp. 312–317, IEEE, 2024.
- [45] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 286–292, IEEE, 2022.
- [46] M. Tayal and S. Kolathaya, "Polygonal cone control barrier functions (PolyC2BF) for safe navigation in cluttered environments," in *2024 European Control Conference (ECC)*, pp. 2212–2217, IEEE, 2024.
- [47] Y.-H. Chen, S. Liu, W. Xiao, C. Belta, and M. Otte, "Control barrier functions via minkowski operations for safe navigation among polytopic sets," *arXiv preprint arXiv:2504.00364*, 2025.
- [48] T. G. Molnar, "Navigating polytopes with safety: A control barrier function approach," *arXiv preprint arXiv:2505.17270*, 2025.
- [49] L. Perko, *Differential equations and dynamical systems*, vol. 7. Springer Science & Business Media, 2013.
- [50] B. Morris, M. J. Powell, and A. D. Ames, "Sufficient conditions for the Lipschitz continuity of QP-based multi-objective control of humanoid robots," in *52nd IEEE Conference on Decision and Control*, pp. 2920–2926, IEEE, 2013.
- [51] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *2021 American Control Conference (ACC)*, pp. 3856–3863, IEEE, 2021.
- [52] P. Ong, M. H. Cohen, T. G. Molnar, and A. D. Ames, "On the properties of optimal-decay control barrier functions," *arXiv preprint arXiv:2507.12717*, 2025.
- [53] P. Glotfelter, I. Buckley, and M. Egerstedt, "Hybrid nonsmooth barrier functions with applications to provably safe and composable collision avoidance for robotic systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1303–1310, 2019.
- [54] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE transactions on automatic control*, vol. 52, no. 8, pp. 1362–1379, 2007.
- [55] J. Reis, W. Xie, D. Cabecinhas, and C. Silvestre, "Nonlinear backstepping controller for an underactuated ASV with model parametric uncertainty: Design and experimental validation," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2514–2526, 2022.

- [56] Z.-P. JIANGdagger and H. Nijmeijer, “Tracking control of mobile robots: A case study in backstepping,” *Automatica*, vol. 33, no. 7, pp. 1393–1399, 1997.
- [57] Y. Yang, G. Feng, and J. Ren, “A combined backstepping and small-gain approach to robust adaptive fuzzy control for strict-feedback nonlinear systems,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 3, pp. 406–420, 2004.
- [58] S.-H. Kim, Y.-S. Kim, and C. Song, “A robust adaptive nonlinear control approach to missile autopilot design,” *Control engineering practice*, vol. 12, no. 2, pp. 149–154, 2004.
- [59] J. Farrell, M. Sharma, and M. Polycarpou, “Backstepping-based flight control with adaptive function approximation,” *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1089–1102, 2005.
- [60] L. Sun, W. Huo, and Z. Jiao, “Adaptive backstepping control of spacecraft rendezvous and proximity operations with input saturation and full-state constraint,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 480–492, 2016.
- [61] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *International workshop on hybrid systems: Computation and control*, pp. 291–305, Springer, 2005.
- [62] T. G. Molnar and A. D. Ames, “Composing control barrier functions for complex safety specifications,” *IEEE Control Systems Letters*, 2023.
- [63] A. J. Taylor, P. Ong, T. G. Molnar, and A. D. Ames, “Safe backstepping with control barrier functions,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5775–5782, IEEE, 2022.
- [64] P. Ong and J. Cortés, “Universal formula for smooth safe stabilization,” in *2019 IEEE 58th conference on decision and control (CDC)*, pp. 2373–2378, IEEE, 2019.
- [65] S. G. Krantz and H. R. Parks, *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.
- [66] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, “Set-based fault detection and isolation for detectable linear parameter-varying systems,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 18, pp. 4381–4397, 2017.



**Hugo Matias** received his B.Sc. and M.Sc. degrees in Electrical and Computer Engineering from the Instituto Superior Técnico, Lisbon, Portugal, in 2021 and 2023, respectively. He is a doctoral candidate in Electrical and Computer Engineering, specializing in Systems, Decision and Control, at the NOVA School of Science and Technology, Caparica, Portugal. His research interests include safety-critical control, state estimation, nonlinear optimization, distributed systems, and computer networks.



**Daniel Silvestre** received his B.Sc. in Computer Networks in 2008 from Instituto Superior Técnico, Lisbon, Portugal, and his M.Sc. in Advanced Computing in 2009 from the Imperial College London, London, United Kingdom. In 2017, he received his Ph.D. (with the highest honors) in Electrical and Computer Engineering from the former university. Currently, he is with the NOVA School of Science and Technology, Caparica, Portugal, and with the Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal. His research interests include fault detection and isolation, distributed systems, guaranteed state estimation, computer networks, optimal control and nonlinear optimization.