# Safe and Dynamically-Feasible Motion Planning using Control Lyapunov and Barrier Functions

Pol Mestres      Carlos Nieto-Granda      Jorge Cortés

*Abstract*—**This paper considers the problem of designing motion planning algorithms for control-affine systems that generate collision-free paths from an initial to a final destination and can be executed using safe and dynamically-feasible controllers. We introduce the `C-CLF-CBF-RRT` algorithm, which produces paths with such properties and leverages rapidly exploring random trees (RRTs), control Lyapunov functions (CLFs) and control barrier functions (CBFs). We show that `C-CLF-CBF-RRT` is computationally efficient for linear systems with polytopic and ellipsoidal constraints, and establish its probabilistic completeness. We showcase the performance of `C-CLF-CBF-RRT` in different simulation and hardware experiments.**

## I. INTRODUCTION

Motion planning refers to the problem of computing a collision-free trajectory for a mobile agent to go from an initial state to a goal state. Motion planning algorithms are the backbone of many robotics applications, but their implementation remains challenging for robots with complex dynamics and environments with irregular obstacles. Even in scenarios where the robot dynamics and the environment obstacles are known, obtaining motion plans is in general a challenging task. Most motion planning algorithms generate high-level plans, consisting of sequences of waypoints in the configuration space, and assume the availability of low-level controllers that can follow such waypoints while avoiding collisions with obstacles. An example of low-level controllers frequently used in applications requiring collision-free navigation are those based on control barrier functions (CBFs) for safety and control Lyapunov functions (CLFs) for stability. However, controllers that simultaneously address safety and stability of the different waypoints might in general be not well-defined. This work is motivated by the need to bridge the gap between motion planner implementations and low-level CLF-CBF controllers that produce dynamically-feasible safe trajectories.

*Literature Review: Trajectory optimization* methods in motion planning [1]–[3] seek to directly design trajectories from the initial state to the goal state that take into account the robot dynamics. These methods usually formulate the planning problem as a high-dimensional nonconvex problem, which can be difficult to solve efficiently by off-the-shelf solvers. To address this, it is common to restrict the problem to a parametric class: [4], [5] uses the so-called MINVO basis, [6] uses B-splines, and [7], [8] use polynomial basis.

P. Mestres and J. Cortés are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, {pomestre,cortes}@ucsd.edu. Carlos Nieto-Granda is with the DEV-COM US Army Research Laboratory (ARL), Adelphi, Maryland, carlos.p.nieto2.civ@army.mil.

Even with the restriction to such parametric classes, the trajectory optimization problems remain nonconvex, and their complexity scales with the dimension of the parameter space. One exception is the recent work [9], which formulates the trajectory optimization problem as a shortest path problem in Graphs of Convex Sets [10], an optimization framework that allows the trajectory optimization problem to be formulated as a mixed-integer convex program for trajectories parameterized by Bernstein polynomials. Despite the low runtimes that this algorithm exhibits in a variety of different robotic systems, it requires a partition of the environment in convex sets, which needs to be precomputed offline.

Regardless of the computational complexity, the restriction of trajectory optimization methods to parametric classes means that they are only guaranteed to produce dynamically feasible solutions for special classes of systems (for example [4]–[7] work for quadrotors, and [8] for feedback linearizable systems). Furthermore, since the controllers needed to track these trajectories are generally open loop, they do not possess the inherent robustness properties associated with feedback control (an exception is [3], which uses model predictive control to generate optimal trajectories).

Our work is more closely aligned with *sampling-based motion planning* methods [11], which seeks to find a collision-free path from an initial state to a goal state through randomly sampling the state space. Despite its simplicity, it has been shown to be a practical solution for efficiently finding feasible paths even for high-dimensional problems. Rapidly-exploring random trees (RRTs) [12] and its variants [13], [14] are a family of sampling-based motion planning algorithms that are simple to implement and are probabilistically complete, meaning that a feasible path (if it exists) is found with probability one as the number of samples goes to infinity. RRTs build a tree rooted at a starting configuration and efficiently explore the configuration space by adding more samples. Despite the widespread use of RRT and the variants outlined above, their performance in systems with general differential constraints and dynamics remains limited, since they rely on the ability to connect any neighboring nodes of the tree with a dynamically feasible trajectory. This requires solving a *two-point boundary value problem* (BVP) [15, Chapter 14], which in general is challenging. Different works [16], [17] address this problem by developing algorithms that achieve optimality guarantees for different classes of systems without requiring the use of a BVP solver. On the one hand, [16] considers controllable linear systems, for which the explicit solution of the BVP can be computed, and [17] focuses on non-holonomic systems where *Chow's condition* holds, whose

accessibility properties can also be used to sidestep the use of a BVP solver. Alternatively, other works introduce heuristics that approximate the solution of the BVP: [18], [19] do it using the linear quadratic regulator, and [20] leverages bang-bang controllers. Other works circumvent solving the BVP by using learning-based approaches. For instance, [21], [22] introduces an offline machine learning phase that learns the solution of the BVP, [23] refines the generation of the dataset used in this offline phase, and [24] learns the solution of the BVP using reinforcement learning techniques. There are also approaches [25] that combine the benefits of trajectory optimization methods with RRT, by constructing a tree of optimized trajectories along with tubes defining their regions of attraction, derived with sum-of-squares programming [26].

Here we bypass the need to solve the BVP or to optimize over sets of trajectories by using two sets of well-established tools: control Lyapunov functions (CLFs) [27], for designing stabilizing controllers for nonlinear systems, and control barrier functions (CBFs) [28], [29], for rendering safe a desired set. In applications where safety and stability specifications need to be met simultaneously, the CLF and CBF conditions can be combined in a variety of different formulations including a quadratic program with a relaxation variable [30], safety filters [31] (where the CBF condition acts on top of a stabilizing nominal controller), or designs based on penalty methods [32]. Even though these control designs have shown great success in applications such as adaptive cruise control [33] and bipedal walking [34], different works have shown that, when combined, they can lead to the existence of undesired equilibria [35]–[37], which can even be asymptotically stable and have large regions of attraction, or the lack of feasibility [32], [38], [39] between the CBF and CLF conditions.

There exist a few works in the literature [40]–[43] that combine the effectiveness of RRT-based algorithms with the safety guarantees and computational efficiency provided by CBFs and CLFs, hence also bypassing the need to compute the solution of a BVP. However, these approaches require the simulation of trajectories derived from a CLF-CBF-based controller in order to determine whether new candidate nodes should be added to the tree. The repeated simulation of such trajectories can significantly slow down the search for a feasible path and compromise the computational efficiency of the resulting algorithm. Moreover, these existing works can be prone to safety violations as a consequence of the numerical errors introduced when simulating these trajectories, and do not formally ensure that the low-level CLF and CBF-based controller possesses both safety and stability guarantees. Finally, [44] introduces `LQR-CBF-RRT⋆`, which is asymptotically optimal and also leverages CBFs to ensure collision-free trajectories. Moreover, this method does not require simulating trajectories obtained with a CLF-CBF-based controller. However, the CBF condition is only verified at a finite sequence of points along a trajectory, which might compromise safety in-between such sampled points. Furthermore, the reference trajectory is generated through an LQR-based controller of a linearized model, which might also not be stabilizing for the original nonlinear system.

*Statement of Contributions:* We consider the problem of designing motion planning algorithms that generate collision-free paths from an initial to a final destination for systems with control-affine dynamics. To ensure that the sequence of waypoints generated by the sampling-based algorithm can be tracked by a controller while ensuring safety and stability, we leverage the theory of CBFs and CLFs. Our contributions are:

(i) We introduce a result of independent interest which shows that the problem of verifying whether a CLF and a CBF are compatible in a set of interest can be solved by solving an optimization problem;

(ii) Although in general such optimization problem is non-convex, we show that for linear systems and CBFs of polytopic or ellipsoidal obstacles, it reduces to a quadratically constrained quadratic program (QCQP), and for CBFs of circular obstacles it can be solved in closed form;

(iii) We leverage the results on compatibility checking of a CLF-CBF pair to develop Compatible CLF-CBF-RRT (or `C-CLF-CBF-RRT` for short), a sampling-based motion planning algorithm that is a variant of RRT. We show that, by construction, `C-CLF-CBF-RRT` generates collision-free paths that can be executed with a CLF-CBF-based controller, and formally establish it is probabilistically complete;

(iv) We show how our proposed approach can be generalized to systems where safety constraints have a high relative degree;

(v) We illustrate our results in simulation and hardware experiments for differential drive robots and compare them with the literature, showing that `C-CLF-CBF-RRT` can generate safe and stable paths with a better average execution time.

Noteworthy properties of `C-CLF-CBF-RRT` as compared to the literature are: it does not require generating closed-loop trajectories at every sampling step because of the compatibility verification of CLF-CBF pairs; it avoids the potential safety violations that occur as a consequence of the numerical errors introduced when simulating trajectories; its computational complexity is tractable provided that the optimization problem verifying the compatibility of the CLF and CBF is tractable; and it ensures by construction that the sequence of generated waypoints can be robustly asymptotically tracked by a safe controller, without introducing unwanted dynamical behaviors such as undesired equilibria, and while ensuring that the optimization problem defining such controller is recursively feasible.

*Notation:* We denote by $\mathbb{Z}_{>0}$, $\mathbb{R}$, and $\mathbb{R}_{\geq 0}$ the set of positive integers, real, and nonnegative real numbers, resp. For $N \in \mathbb{Z}_{>0}$, we denote $[N] = \{1, 2, \ldots, N\}$. Given $x \in \mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. Given matrix $B \in \mathbb{R}^{n \times m}$, $\text{Im}(B)$ denotes its image. The symbols $\mathbb{I}_n$, $\mathbb{0}_n$ denote the identity and zero matrices of dimension $n \in \mathbb{Z}_{>0}$, and $\mathbf{0}_n$ is the zero vector of dimension $n$. Given a set $\mathcal{S} \subset \mathbb{R}^n$, we denote its boundary by $\partial \mathcal{S}$ and its closure by $\text{Cl}(\mathcal{S})$. Given an arbitrary set $A$, we let $\mathcal{P}(A)$ be the power set of $A$, i.e., the set of all subsets of $A$, including the empty set and $A$ itself. We denote by $\mathcal{B}(x, \delta)$ the Euclidean closed ball of center $x \in \mathbb{R}^n$

and radius $\delta > 0$, i.e., $\mathcal{B}(x, \delta) := \{y \in \mathbb{R}^n \ : \ \|y - x\| \leq \delta\}$. Given $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ and a smooth $W : \mathbb{R}^n \to \mathbb{R}$, the notation $L_f W : \mathbb{R}^n \to \mathbb{R}$ (resp., $L_g W : \mathbb{R}^n \to \mathbb{R}^m$) denotes the Lie derivative of $W$ with respect to $f$ (resp., $g$), that is $L_f W = \nabla W^T f$ (resp., $\nabla W^T g$). A function $\beta : \mathbb{R} \to \mathbb{R}$ is extended class $\mathcal{K}_\infty$ if it is continuous, $\beta(0) = 0$, $\beta$ is strictly increasing and $\lim_{s \to \pm\infty} \beta(s) = \pm\infty$. A function $V : \mathbb{R}^n \to \mathbb{R}$ is positive definite with respect to $q \in \mathbb{R}^n$ if $V(q) = 0$ and $V(x) > 0$ for $x \neq q$. Given a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$, its generalized gradient at $x \in \mathbb{R}^n$ is $\partial f(x) = \mathrm{co}\{\lim_{i \to \infty} \nabla f(x_i) \ : \ x_i \to x, x_i \notin S \cup \Gamma_f\}$, where $\Gamma_f$ is the zero-measure set where $f$ is non-differentiable and $S$ is any set of measure zero. An undirected graph $\mathcal{M}$ is a pair $\mathcal{M} = (V, \mathcal{E})$, where $V = \{1, \ldots, N\}$ is a finite set called the vertex set, $\mathcal{E} \subset V \times V$ is called the edge set where $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. A path in $\mathcal{M}$ is a sequence of vertices $v_1, \ldots, v_k$, with $k \in \mathbb{Z}_{>0}$, such that for all $i \in [k-1]$, $(v_i, v_{i+1}) \in \mathcal{E}$. A tree is an undirected graph in which there exists a single path between any pair of vertices.

## II. PRELIMINARIES

Here we review control Lyapunov functions, control barrier functions, and rapidly exploring random trees. For reference, we provide a summary table of the symbols used throughout the paper in Table III.

### A. Control Lyapunov and Control Barrier Functions

Consider a control-affine system

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz functions, with $x \in \mathbb{R}^n$ the state and $u \in \mathbb{R}^m$ the input. Throughout the paper, and without loss of generality, we assume $f(0) = 0$, so that the origin $x = 0$ is the desired equilibrium point of the (unforced) system.

We start by recalling the notion of Control Lyapunov function (CLF) [45], [46].

**Definition II.1.** *(Control Lyapunov Function): Given an open set $\mathcal{D} \subseteq \mathbb{R}^n$, a point $q \in \mathbb{R}^n$ with $q \in \mathcal{D}$, a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is a **CLF** with respect to $q$ in $\mathcal{D}$ for the system (1) if*

- *$V$ is proper in $\mathcal{D}$, i.e., $\{x \in \mathcal{D} \ : \ V(x) \leq c\}$ is a compact set for all $c > 0$,*
- *$V$ is positive definite with respect to $q$,*
- *there exists a continuous positive definite function $W : \mathbb{R}^n \to \mathbb{R}$ with respect to $q$ such that, for each $x \in \mathcal{D}$, there exists a control $u \in \mathbb{R}^m$ satisfying*

$$L_f V(x) + L_g V(x)u \leq -W(x). \tag{2}$$

CLFs provide a way to guarantee asymptotic stability of the origin. Namely, if a Lipschitz controller $u_{\mathrm{st}} : \mathbb{R}^n \to \mathbb{R}^m$ is such that, for every $x \in \mathcal{D}$, $u = u_{\mathrm{st}}(x)$ satisfies (2), then the origin is asymptotically stable for the closed-loop system [45]. Such controllers can be synthesized by means of the pointwise minimum-norm (PMN) control optimization [46, Chapter 4.2],

$$u(x) = \arg \min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2$$

s.t. (2) holds.

Note that, at each $x \in \mathbb{R}^n$, this is a quadratic program in $u$.

Next we define the notion of Boolean Nonsmooth Control Barrier Function (BNCBF), adapted from [47, Definition II.8].

**Definition II.2.** *(BNCBF [47, Definition II.8]): Given $N \in \mathbb{Z}_{>0}$, let $h_i : \mathbb{R}^n \to \mathbb{R}$, for $i \in [N]$, be continuously differentiable functions. Let $h(x) = \max_{i \in [N]} h_i(x)$ and*

$$\mathcal{C} = \{x \in \mathbb{R}^n \ : \ h(x) \geq 0\}, \tag{3a}$$
$$\partial\mathcal{C} = \{x \in \mathbb{R}^n \ : \ h(x) = 0\}. \tag{3b}$$

*Suppose that the set $\mathcal{C}$ is nonempty. Then, $h$ is a BNCBF of $\mathcal{C}$ for (1) if there exists a locally Lipschitz extended class $\mathcal{K}_\infty$ function $\alpha : \mathbb{R} \to \mathbb{R}$ such that for every $x \in \mathcal{C}$ there exists $u \in \mathbb{R}^m$ such that,*

$$\min_{v \in \partial h(x)} v^T(f(x) + g(x)u) \geq -\alpha(h(x)).$$

In case $N = 1$, Definition II.2 reduces to the standard notion of Control Barrier Function [28, Definition 2]. Given $x \in \mathbb{R}^n$, let $\mathcal{I}(x) := \{i \in [N] \ : \ h(x) = h_i(x)\}$ denote the set of *active* functions. The following result is adapted from [47, Theorem III.6] and provides a sufficient condition for $h$ to be a BNCBF.

**Proposition II.3.** *(Sufficient Condition for BNCBF): Suppose there is an extended class $\mathcal{K}_\infty$ function $\alpha : \mathbb{R} \to \mathbb{R}$ such that, for all $x \in \mathbb{R}^n$, there exists $u \in \mathbb{R}^m$ with*

$$L_f h_i(x) + L_g h_i(x)u \geq -\alpha(h(x)), \tag{4}$$

*for all $i \in \mathcal{I}(x)$. Then, $h$ is a BNCBF of $\mathcal{C}$.*

If a measurable and locally bounded controller $u_s : \mathbb{R}^n \to \mathbb{R}^m$ is such that, for every $x \in \mathbb{R}^n$, $u = u_s(x)$ satisfies (4), then $u_s$ renders $\mathcal{C}$ forward invariant (cf. [47, Theorem II.7, Definition II.8]).

When dealing with both safety and stability specifications, it is important to note that an input $u$ might satisfy (2) but not (4), or vice versa. The following notion, adapted from [39, Definition 2.3], captures when a CLF $V$ and a BNCBF $h$ are compatible.

**Definition II.4.** *(Compatibility of CLF-BNCBF pair): Let $\mathcal{D} \subseteq \mathbb{R}^n$ be open, $\mathcal{C} \subset \mathcal{D}$ be closed, $V$ a CLF on $\mathcal{D}$ and $h$ a BNCBF of $\mathcal{C}$. Then, $V$ and $h$ are compatible in a set $\tilde{\mathcal{D}} \subset \mathcal{D}$ if there exist a positive definite function $W : \mathbb{R}^n \to \mathbb{R}$ and an extended class $\mathcal{K}_\infty$ function $\alpha : \mathbb{R} \to \mathbb{R}$ such that, for all $x \in \tilde{\mathcal{D}}$, there exists $u \in \mathbb{R}^m$ satisfying (2) and (4) for all $i \in \mathcal{I}(x)$ simultaneously.*

If $V$ and $h$ are compatible in a set $\tilde{\mathcal{D}}$, we can define the minimum norm controller that satisfies the CLF and BNCBF conditions $u^* : \tilde{\mathcal{D}} \to \mathbb{R}^m$ as follows:

$$u^*(x) := \arg \min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2 \tag{5}$$

$$\text{s.t. } L_f V(x) + L_g V(x)u \leq -W(x),$$
$$L_f h_i(x) + L_g h_i(x)u \geq -\alpha(h(x)), \ \forall \ i \in \mathcal{I}(x).$$

If $u^*$ is locally Lipschitz, then it ensures that $\mathcal{C}$ is forward invariant and that the origin is asymptotically stable for the closed-loop system.

## B. Rapidly-exploring Random Trees (RRTs)

Here, we review `GEOM-RRT` [13], cf. Algorithm 1, a version of RRT [12] upon which we rely later. The input for

---

**Algorithm 1** `GEOM-RRT`

---

1: **Parameters**: $x_{\text{init}}, \mathcal{X}_{\text{goal}}, k, \eta$
2: $\mathcal{T}$.init($x_{\text{init}}$)
3: **for** $i \in [1, \dots, k]$ **do**
4:     $x_{\text{rand}} \leftarrow$ `RANDOM_STATE`
5:     $x_{\text{near}} \leftarrow$ `NEAREST_NEIGHBOR`($x_{\text{rand}}, \mathcal{T}$)
6:     $x_{\text{new}} \leftarrow$ `NEW_STATE`($x_{\text{rand}}, x_{\text{near}}, \eta$)
7:     **if** `COLLISION_FREE`($x_{\text{near}}, x_{\text{new}}$) **then**
8:         $\mathcal{T}$.add_vertex($x_{\text{new}}$)
9:         $\mathcal{T}$.add_edge($x_{\text{near}}, x_{\text{new}}$)
10:         **if** $x_{\text{new}} \in \mathcal{X}_{\text{goal}}$ **then**
11:             **return** $\mathcal{T}$
12:         **end if**
13:     **end if**
14: **end for**
15: **return** $\mathcal{T}$

---

`GEOM-RRT` consists of a state space $\mathcal{X}$, an initial configuration $x_{\text{init}}$, goal region $\mathcal{X}_{\text{goal}}$, number of iterations $k$, and a steering parameter $\eta$ whose use is defined in the sequel. The algorithm builds a tree $\mathcal{T}$ by executing $k$ iterations of the following form:

At each iteration, a new random sample $x_{\text{rand}}$ is obtained by uniformly sampling $\mathcal{X}$ using `RANDOM_STATE`(). The function `NEAREST_NEIGHBOR`($x_{\text{rand}}, \mathcal{T}$) returns the vertex $x_{\text{near}}$ from $\mathcal{T}$ that is closest in the Euclidean distance to $x_{\text{rand}}$. Next, a new configuration $x_{\text{new}} \in \mathcal{X}$ is returned by the `NEW_STATE` function such that $x_{\text{new}}$ is on the line segment between $x_{\text{near}}$ and $x_{\text{rand}}$ and the distance $\|x_{\text{near}} - x_{\text{new}}\|$ is at most $\eta$. Finally, the function `COLLISION_FREE`($x_{\text{near}}, x_{\text{new}}$) checks whether the straight line from $x_{\text{near}}$ and $x_{\text{new}}$ is collision free. If this is the case, $x_{\text{new}}$ is added as a vertex to $\mathcal{T}$ and is connected by an edge from $x_{\text{near}}$. If $x_{\text{new}} \in \mathcal{X}_{\text{goal}}$, there exists a single path in $\mathcal{T}$ from $x_{\text{init}}$ to $x_{\text{new}}$.

A notable property of `GEOM-RRT` is that it is *probabilistically complete*, meaning that the probability that the algorithm will return a collision-free path from the initial state to the goal state (if one exists) approaches one as the number of iterations tends to infinity [48].

## III. PROBLEM STATEMENT

Let $\mathcal{R}$ be a compact and convex set in $\mathbb{R}^n$ containing $M$ known obstacles $\{\mathcal{O}_l\}_{l=1}^M$, with $\text{Int}(\mathcal{O}_i) \cap \text{Int}(\mathcal{O}_j) = \emptyset$ for all $i \neq j \in [M]$. Let $\mathcal{F} := \mathcal{R} \backslash \cup_{l=1}^M \mathcal{O}_l$ denote the *safe* space. For each $l \in [M]$, we assume that there exists a positive integer $N_l \in \mathbb{Z}_{>0}$ and known continuously differentiable functions $\{h_{i,l} : \mathbb{R}^n \to \mathbb{R}\}_{i \in [N_l]}$ such that $\mathcal{O}_l := \{x \in \mathbb{R}^n : h_l(x) = \max_{i \in [N_l]} h_{i,l}(x) < 0\}$. Even though this imposes a specific structure on the set $\mathcal{O}_l$, one can obtain more complex obstacles by considering sets of the form $\cup_{i \in \mathcal{M}} \mathcal{O}_i$, with $\mathcal{M}$ a subset of $[M]$.

The robot dynamics are control-affine of the form (1), with $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ locally Lipschitz. For each $l \in [M]$, $h_l$ is a BNCBF of $\mathbb{R}^n \backslash \mathcal{O}_l$ for these dynamics, with associated extended class $\mathcal{K}_\infty$ function $\alpha_l$. We also assume

$$\nabla h_{i,l}(x)^T g(x) \neq \mathbf{0}_m, \quad \forall x \in \mathcal{F}, l \in [M], i \in [N_l],$$

i.e., one differentiation of $h_{i,l}$ already makes the input $u$ appear explicitly. Given an initial state $x_{\text{init}} \in \mathcal{R}$ and a final goal set $\mathcal{X}_{\text{goal}} \subset \mathcal{R}$, our aim is to develop a sampling-based motion planning algorithm that constructs a collision-free path $\mathcal{A} := \{x_i\}_{i=1}^{N_a}$ from $x_{\text{init}}$ to $\mathcal{X}_{\text{goal}}$ that is dynamically feasible, i.e., such that for each pair of consecutive waypoints in $\mathcal{A}$, there exists a control law that generates a safe trajectory that connects them. Our approach to solve this problem leverages the theory of CLFs and BNCBFs to design controllers which (i) have safety and stability guarantees by design, and (ii) can be implemented efficiently to help reduce the computational burden of generating dynamically feasible trajectories.

## IV. CLF AND BNCBF COMPATIBILITY VERIFICATION

The key challenge in our proposed approach to the problem outlined in Section III is that the optimization (5) defining the CLF-CBF-based controller has to be feasible at all points along the trajectory. In this section we tackle this problem and show how such a feasibility check can be performed in general, and how it is efficient in two specific cases of interest.

### A. Compatibility Verification for General Dynamics and Obstacles

In this section we consider the problem of verifying that a CLF and a BNCBF are compatible in systems for general dynamics and obstacles. The following result gives a characterization for when a CLF and a BNCBF are compatible in the region $\mathcal{R}$.

**Proposition IV.1.** *(Characterization of CLF-BNCBF Compatibility): Given $q \in \mathcal{F}$, let $V_q : \mathbb{R}^n \to \mathbb{R}$ be a CLF of (1) with respect to q. Let $l \in [M]$ and assume that $h_l$ is a BNCBF of $\mathbb{R}^n \backslash \mathcal{O}_l$. Let $W_q : \mathbb{R}^n \to \mathbb{R}$ be a positive definite function with respect to q and $\alpha_l : \mathbb{R} \to \mathbb{R}$ be an extended class $\mathcal{K}_\infty$ function. For each $\mathcal{J} \subset \mathcal{P}([N_l])$, let $Z_{l,\mathcal{J}} := \{x \in \mathbb{R}^n : \mathcal{I}_l(x) = \mathcal{J}\}$ denote the set of points where the active constraints defining obstacle $\mathcal{O}_l$ correspond to the indices in $\mathcal{J}$. For $\Gamma \subset \mathcal{R}$, define*

$$\zeta_1 = \min_{\substack{x \in \Gamma \\ \{\beta_i \in \mathbb{R}\}_{i \in \mathcal{J}}}} \left\| \sum_{i \in \mathcal{J}} \beta_i L_g h_{i,l}(x) - L_g V_q(x) \right\|^2 \quad (6a)$$

$$s.t. \quad \beta_i \geq 0, \ i \in \mathcal{J}, \quad (6b)$$

$$h_{j,l}(x) \leq h_{i,l}(x), \ \forall j \notin \mathcal{J}, i \in \mathcal{J}, \quad (6c)$$

$$h_l(x) \geq 0. \quad (6d)$$

*If $\zeta_1 \neq 0$, then $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. Otherwise, if $\zeta_1 = 0$, let*

$$\zeta_2 = \min_{\substack{x \in \Gamma \\ \{\beta_i \in \mathbb{R}\}_{i \in \mathcal{J}}}} \Phi(x, \{\beta_i\}_{i \in \mathcal{J}}), \quad (7a)$$

$$s.t. \quad \sum_{i \in \mathcal{J}} \beta_i L_g h_{i,l}(x) = L_g V_q(x), \quad (7b)$$

$$\beta_i \geq 0, \ i \in \mathcal{J}, \tag{7c}$$

$$h_{j,l}(x) \leq h_{i,l}(x), \ \forall j \notin \mathcal{J}, i \in \mathcal{J}, \tag{7d}$$

$$h_l(x) \geq 0, \tag{7e}$$

*for* $\Phi(x, \{\beta_i\}_{i \in \mathcal{J}}) = -W_q(x) - L_f V_q(x) + \sum_{i \in \mathcal{J}} \beta_i(L_f h_{i,l}(x) + \alpha_l(h_{i,l}(x)))$. *If* $\zeta_2 \geq 0$, *then* $V_q$ *and* $h_l$ *are compatible in* $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. *Conversely, if* $V_q$ *and* $h_l$ *are compatible in* $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$ *then there exists an extended class* $\mathcal{K}_\infty$ *function* $\alpha_l$ *and a positive definite function* $W_q$ *with respect to* $q$ *such that either* $\zeta_1 \neq 0$ *or* $\zeta_1 = 0$ *and* $\zeta_2 \geq 0$.

*Proof.* First note that if $\zeta_1 = 0$, the optimization problem (7) is feasible and therefore $\zeta_2$ is well-defined. By Farkas' Lemma [49], $V_q$ and $h_l$ are compatible at $x \in Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$ if and only if for some positive definite function $W_q$ with respect to $q$ and some extended class $\mathcal{K}_\infty$ function $\alpha_l$, there do not exist $\beta_0 \in \mathbb{R}_{\geq 0}$, $\{\beta_i\}_{i \in \mathcal{J}} \subset \mathbb{R}_{\geq 0}$ such that

$$\beta_0 L_g V_q(x) = \sum_{i \in \mathcal{J}} \beta_i L_g h_{i,l}(x), \tag{8a}$$

$$\beta_0(-L_f V_q(x) - W(x)) \\ + \sum_{i \in \mathcal{J}} \beta_i(\alpha_l(h_{i,l}(x)) + L_f h_{i,l}(x)) < 0. \tag{8b}$$

First suppose that for some $W_q$ and $\alpha_l$, either $\zeta_1 \neq 0$ or $\zeta_1 = 0$ and $\zeta_2 \geq 0$. Suppose there exists a solution $s_1^* = (x^*, \beta_0^*, \{\beta_i^*\}_{i \in \mathcal{I}_l(x)})$ of (8) and let us reach a contradiction. If $\beta_0^* = 0$, then, (8) implies that the constraints $L_f h_{i,l}(x) + L_g h_{i,l}(x)u \geq -\alpha_l(h_{i,l}(x))$ are not simultaneously feasible, which means that $h_l$ is not a BNCBF, hence arriving at a contradiction. Therefore, $s_1^*$ must be such that $\beta_0^* > 0$. By taking $\tilde{\beta}_i = \frac{\beta_i}{\beta_0}$ for $i \in \mathcal{J}$, we deduce that $(x^*, \{\tilde{\beta}_i\}_{i \in \mathcal{J}})$ is a solution of (6) with a value of the objective function equal to zero. This means that if $\zeta_1 \neq 0$, the solution $s_1^*$ does not exist and $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. Otherwise, if $\zeta_1 = 0$, then $(x^*, \{\tilde{\beta}_i\}_{i \in \mathcal{J}})$ is a solution of (7) with a strictly negative value of the objective function. This means that if $\zeta_1 = 0$ and $\zeta_2 \geq 0$, the solution $s_1^*$ does not exist and $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. Conversely, suppose that $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. This implies that there exists $W_q$ and $\alpha_l$ such that (8) has no solution. If (8a) has no solution, then $\zeta_1 \neq 0$. If (8a) has a solution but (8b) does not, then $\zeta_1 = 0$ and $\zeta_2 \geq 0$. $\square$

Note that Proposition IV.1 is valid for any set $\Gamma \subset \mathcal{R}$. Intuitively, since the CLF and BNCBF conditions define half-spaces in the control input $u$, (6) checks whether the normal vectors of the hyperplanes defining such half-spaces are linearly independent. If this condition does not hold, (7) checks whether the input-independent terms of the CLF and BNCBF conditions leave enough space for such conditions to be compatible. Additionally, optimization problems (6) and (7) need to be checked for every possible set of active constraints. The constraints (6c) and (7d) ensure that $\mathcal{J}$ is the set of active constraints at $x$. Often, one is interested in verifying the compatibility of a CLF and a BNCBF only in a small subset of $\mathcal{R}$, in which case the flexibility provided by the set $\Gamma$ is useful.

**Remark IV.2.** *(Checking for all Possible Sets of Active Constraints):* Given a subset $\mathcal{J} \subset \mathcal{P}([N_l])$ of functions $\{h_{i,l}\}$, Proposition IV.1 provides a way to verify if the CLF and the BNCBF are compatible at the points in the region of interest $\Gamma \cap \mathcal{F}$ where such functions are active. Let $H_{l,\mathcal{J}} := \{x \in \Gamma : \mathcal{I}_l(x) = \mathcal{J}\}$ be the points in $\Gamma$ where the constraints with index in $\mathcal{J}$ are active, and $\mathcal{S}_l := \{\mathcal{J} \subset \mathcal{P}([N_l]) : H_{l,\mathcal{J}} \neq \emptyset\}$ be the sets of indices for which the above set is nonempty. The class $\mathcal{S}_l$ contains all possible sets of active constraints in $\Gamma$. By checking the condition in Proposition IV.1 for all $\mathcal{J}$ in $\mathcal{S}_l$, we can verify if the CLF and the BNCBF are compatible in $\Gamma \cap \mathcal{F}$. In practice, given a region $\Gamma$ where we are interested in checking the compatibility of $V_q$ and $h_l$, one can often identify the indices that can achieve a maximum value in $\Gamma$ (for example, for polytopic obstacles in the plane, only a few of the functions $h_{i,l}$ have points in $\Gamma$ where they take positive values). This means that the cardinality of $\mathcal{S}_l$ is often small and the number of checks using Proposition IV.1 can be kept small. $\bullet$

**Remark IV.3.** *(Verifying Compatibility for Multiple BNCBFs):* Proposition IV.1 actually provides a way to check whether the optimization problem (5) is feasible at *all* points of $\Gamma$. This can be done as follows: one first finds all $l \in [M]$ such that $\Gamma \cap \mathcal{O}_l \neq \emptyset$. If $\Gamma$ can be expressed as the 0-sublevel set of a convex differentiable function $\gamma$, i.e., $\Gamma := \{x \in \mathbb{R}^n : \gamma(x) \leq 0\}$, and the functions $h_{i,l}$ are convex, then this can be solved efficiently by checking that the solution of the convex problem

$$\min_{x \in \mathbb{R}^n} \gamma(x)$$
$$\text{s.t.} \quad h_{i,l}(x) \leq 0, \quad \forall i \in [N_l]$$

is non-positive. The BNCBF constraints associated with those $l' \in [M]$ such that $\Gamma \cap \mathcal{O}_{l'} = \emptyset$ can be neglected since, given a controller that satisfies all the other BNCBF constraints, it can be shown to also satisfy the BNCBF constraints for such $l' \in [M]$ by taking the corresponding extended class $\mathcal{K}_\infty$ function $\alpha_{l'}$ linear with sufficiently large slope. On the other hand, for $l' \in [M]$ such that $\Gamma \cap \mathcal{O}_{l'} \neq \emptyset$, Proposition IV.1 ensures that there exists a small neighborhood around $\partial \mathcal{O}_{l'}$, not containing points of any other obstacle, where $V$ and $h_{l'}$ are compatible. By taking the extended class $\mathcal{K}_\infty$ functions of the other CBF constraints as linear functions with sufficiently large slope, (5) is feasible in each of these neighborhoods. Finally, for points in $\Gamma$ not belonging to any of these neighborhoods, the extended class $\mathcal{K}_\infty$ functions can also be taken as linear with sufficiently large slope to guarantee that (5) is feasible. $\bullet$

**Remark IV.4.** *(About the Choice of CLF and Class $\mathcal{K}_\infty$ Function):* Note that, when solving the optimization problems (6) and (7) for fixed $V_q$, $\alpha_l$, and $W_q$, it is not guaranteed that $\zeta_1 \neq 0$ or $\zeta_1 = 0$ and $\zeta_2 \geq 0$. If $\tilde{\alpha}$ is an extended class $\mathcal{K}_\infty$ function with $\tilde{\alpha}(s) \geq \alpha(s)$ for all $s \in \mathbb{R}$, the objective function $\Phi$ of (7) does not decrease at any point, which means that the value of $\zeta_1$ remains the same, but the condition $\zeta_2 \geq 0$ becomes easier to satisfy. A similar behavior occurs if $\tilde{W}$ is a positive definite function with $\tilde{W}(x) \leq W(x)$ for all $x \in \mathbb{R}^n$. We leverage these observations in Section V when we introduce our proposed motion planning algorithm. $\bullet$

**Remark IV.5.** *(Regularity Properties of the Controller):* If $V_q$ and $h_l$ are compatible in $\mathcal{R}$ for all $l \in [M]$, the CLF-CBF-based controller (5) is well defined, i.e., the optimization (5) is feasible for all points in $\mathcal{R}$. However, slightly stronger conditions are needed to ensure that such CLF-CBF-based controller is locally Lipschitz and therefore can be used to render $\mathcal{C}$ forward invariant and the origin asymptotically stable. We refer the reader to [50] for a survey on different conditions that ensure continuity, Lipschitzness, and other regularity properties of optimization-based controllers of the form (5). These conditions are often satisfied in practice and are mostly related to the dynamics and the specific obstacles, which in our problem here are given and not subject to design. Therefore, throughout this work, we assume that (5) satisfies at least one of the sufficient conditions outlined in [50] that ensure that the resulting controller is locally Lipschitz. $\bullet$

**Remark IV.6.** *(Input Constraints):* In many applications, one is interested in verifying whether the CLF and BNCBF conditions are simultaneously feasible with a control input $u$ constrained to lie on the set $\{u \in \mathbb{R}^m : C_1 u \leq c_2\}$, with $C_1 \in \mathbb{R}^{c \times m}$, $c_2 \in \mathbb{R}^c$, and $c \in \mathbb{Z}_{>0}$. Equivalently, we seek to verify whether the inequalities

$$L_f h_{j,l}(x) + L_g h_{j,l}(x)u \geq -\alpha_{i,l}(h_{j,l}(x)),$$
$$\forall j \in \mathcal{I}_l(x), l \in [M],$$
$$L_f V_i(x) + L_g V_i(x)u + W_i(x) \leq 0, \qquad (9)$$
$$C_1 u \leq c_2,$$

are simultaneously feasible. This problem can also be treated using Farkas' Lemma [49] to obtain a result analogous to Proposition IV.1. For example, the objective function in (6) should be adjusted to $\left\| \sum_{i \in \mathcal{J}} \beta_j L_g h_{i,l}(x) - L_g V_q(x) - C_1^T \bar{\beta} \right\|$, where $\bar{\beta} \in \mathbb{R}^c$ is an additional optimization variable with entries that are required to be positive. Instead, the objective function in (7) should be adjusted to $\bar{\beta}^T c_2 - W_q(x) - L_f V_q(x) + \sum_{i \in \mathcal{J}} \beta_i(L_f h_{i,l}(x) + \alpha_l(h_{i,l}(x)))$. $\bullet$

Proposition IV.1 shows that the problem of checking whether a CLF and a BNCBF are compatible in a region of interest can be reduced to solving a pair of optimization problems. However, in general, the optimization problems (6) and (7) are not convex and can be computationally intractable. Our forthcoming exposition provides two particular cases of dynamics and obstacles for which these two optimization problems are computationally tractable.

### B. Compatibility Verification for Linear Systems and Polytopic Obstacles

In this section we particularize our discussion to linear dynamics,

$$\dot{x} = Ax + Bu, \qquad (10)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and the obstacles are polytopic (i.e., the functions $h_{i,l}$ are affine). We start by introducing some useful notation. For each $l \in [M]$, let $a_{i,l} \in \mathbb{R}^n$, $b_{i,l} \in \mathbb{R}$ be such that $h_{i,l}(x) = a_{i,l}^T x + b_{i,l}$. We further assume that $h_l$

is a BNCBF, i.e., there exists an extended class $\mathcal{K}_\infty$ function $\alpha_l$ such that, for all $x \in \mathbb{R}^n \backslash \mathcal{O}_l$, there exists $u \in \mathbb{R}^m$ with

$$a_{i,l}^T(Ax + Bu) \geq -\alpha_l(a_{i,l}^T x + b_{i,l})$$

for all $i \in \mathcal{I}_l(x)$. We further assume that given $q \in \mathbb{R}^n$, a quadratic CLF is available, i.e., we have a positive definite matrix $P \in \mathbb{R}^{n \times n}$ such that $V_q : \mathbb{R}^n \to \mathbb{R}$, defined as $V_q(x) = (x - q)^T P(x - q)$, is a CLF with respect to $q$ in $\mathbb{R}^n$ of (10) with associated positive definite function $W_q : \mathbb{R}^n \to \mathbb{R}$.

The following result follows by applying Proposition IV.1 to the case when dynamics are linear and obstacles polytopic.

**Proposition IV.7.** *(CLF-BNCBF Compatibility for Linear Dynamics and Polytopic Obstacles):* Let $\Gamma \subset \mathcal{R}$, $l \in [M]$, $\mathcal{J} \in \mathcal{P}([N_l])$, $q \in \mathcal{F}$, and define

$$\zeta_1 := \min_{\substack{x \in \Gamma \\ \{\beta_i \in \mathbb{R}\}_{i \in \mathcal{J}}}} \left\| \sum_{i \in \mathcal{J}} \beta_i B^T a_{i,l} - B^T P(x - q) \right\|^2 \qquad (11a)$$

$$s.t. \ \beta_i \geq 0, \ \forall i \in \mathcal{J}, \qquad (11b)$$
$$a_{j,l}^T x + b_{j,l} \leq a_{i,l}^T x + b_{i,l}, \ \forall j \notin \mathcal{J}, i \in \mathcal{J}, \quad (11c)$$
$$a_{i,l}^T x + b_{i,l} \geq 0, \ i \in \mathcal{J}. \qquad (11d)$$

*If $\zeta_1 \neq 0$, then $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. Otherwise, if $\zeta_1 = 0$, let*

$$\zeta_2 := \min_{\substack{x \in \Gamma \\ \{\beta_i \in \mathbb{R}\}_{i \in \mathcal{J}}}} \Phi(x, \{\beta_i\}_{i \in \mathcal{J}}) \qquad (12a)$$

$$s.t. \ \sum_{i \in \mathcal{J}} \beta_i B^T a_{i,l} = B^T P(x - q), \qquad (12b)$$
$$\beta_i \geq 0, \ \forall i \in \mathcal{J}, \qquad (12c)$$
$$a_{j,l}^T x + b_{j,l} \leq a_{i,l}^T x + b_{i,l}, \ \forall j \notin \mathcal{J}, i \in \mathcal{J}, \quad (12d)$$
$$a_{i,l}^T x + b_{i,l} \geq 0, \ i \in \mathcal{J}, \qquad (12e)$$

*with $\Phi(x, \{\beta_i\}_{i \in \mathcal{J}}) = -W_q(x) - (x - q)^T PAx + \sum_{i \in \mathcal{J}} \beta_i(\alpha_l(a_{i,l}^T x + b_{i,l}) + a_{i,l}^T Ax)$. If $\zeta_2 \geq 0$, then $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$. Conversely, if $V_q$ and $h_l$ are compatible in $Z_{l,\mathcal{J}} \cap \Gamma \cap \mathcal{F}$, then there exists an extended class $\mathcal{K}_\infty$ function $\alpha_l$ and a positive definite function $W_q$ with respect to $q$ such that either $\zeta_1 \neq 0$ or $\zeta_1 = 0$ and $\zeta_2 \geq 0$.*

We end this section by discussing the tractability of the optimizations (11) and (12). If $W_q$ is a quadratic function (as it is often the case in practice), $\alpha(s) = \alpha_0 s$, with $\alpha_0 > 0$, and $\Gamma$ is given by a sublevel set of a quadratic function (e.g., if it is the sublevel set a quadratic CLF $V_q$), then (11) and (12) both have quadratic objective functions and quadratic constraints, i.e., they are quadratically constrained quadratic programs (QCQPs). Moreover, if $\Gamma$ is the sublevel set of a convex quadratic function, then (11) is a convex QCQP (whereas in general, (12) is non-convex). If instead $\Gamma$ is the sublevel set of a piecewise linear function, both (11) and (12) have affine constraints and therefore are quadratic programs (QPs). Moreover, (11) is a convex QP. In either case, even if the resulting QCQPs or QPs are non-convex, there exist efficient heuristics [51], [52] to solve these programs. Finally, Proposition IV.7 can be applied to settings where obstacles are not polytopic by constructing outer approximations of them using polytopes and considering the resulting union of convex sets.

## C. Compatibility Verification for Linear Systems and Ellipsoidal Obstacles

In this section, we again consider linear dynamics (10), but now assume obstacles are ellipsoidal, i.e., $\mathcal{O}_l = \{x \in \mathbb{R}^n : r_l^2 > (x - c_l)^T R_l(x - c_l)\}$, for some positive definite matrix $R_l \in \mathbb{R}^{n \times n}$, $c_l \in \mathbb{R}^n$, and $r_l > 0$. In this case, we take $h_l(x) = -r_l^2 + (x - c_l)^T R_l(x - c_l)$ (which is continuously differentiable and therefore $N_l = 1$ for all $l \in [M]$) and $V_q(x) = (x - q)^T P(x - q)$, for some positive definite matrix $P \in \mathbb{R}^{n \times n}$. Then the following result follows from applying Proposition IV.1 to the case when dynamics are linear and obstacles are ellipsoidal.

**Proposition IV.8.** *(Sufficient Condition for CLF-BNCBF Compatibility for Linear Dynamics and Ellipsoidal Obstacles): Let $\Gamma \subset \mathcal{R}$, $l \in [M]$, $q \in \mathcal{F}$, $\alpha_l > 0$, and define*

$$\zeta_1 := \min_{x \in \Gamma, y \in \mathbb{R}^n, \beta \in \mathbb{R}} \left\| B^T y - B^T P(x - q) \right\|^2 \tag{13a}$$

$$\text{s.t. } \beta \geq 0, \ h_l(x) \geq 0, \ y = -2\beta R_l(x - c_l). \tag{13b}$$

*If $\zeta_1 \neq 0$, then $V_q$ and $h_l$ are compatible in $\Gamma \cap \mathcal{F}$. Otherwise, if $\zeta_1 = 0$, let*

$$\zeta_2 := \min_{x \in \Gamma, y \in \mathbb{R}^n, \beta \in \mathbb{R}} \Phi(x, y, \beta) \tag{14a}$$

$$\text{s.t. } B^T y = B^T P(x - q), \tag{14b}$$

$$\beta \geq 0, \ h_l(x) \geq 0, \ y = -2\beta R_l(x - c_l), \tag{14c}$$

*with $\Phi(x, y, \beta) = -W_q(x) - (x - q)^T PAx - \beta \alpha_l r_l^2 - \alpha_l(x - c_l)^T \frac{y}{2} + y^T Ax$. If $\zeta_2 \geq 0$, then $V_q$ and $h_l$ are compatible in $\Gamma \cap \mathcal{F}$.*

If $\Gamma$ is the sublevel set of a quadratic function and $W_q$ is quadratic, both (13) and (14) are QCQPs and can therefore be solved efficiently [51], [52]. Let us next further restrict our attention to single-integrator dynamics, i.e.,

$$\dot{x} = u, \tag{15}$$

and circular obstacles, i.e., $\mathcal{O}_l = \{x \in \mathbb{R}^n : \|x - c_l\| < r_l\}$ for some $c_l \in \mathbb{R}^n$ and $r_l > 0$. In this case, we take $h_l(x) = \|x - c_l\|^2 - r_l^2$, $V_q(x) = \|x - q\|^2$, and $W_q(x) = (x - q)^T Q(x - q)$, where $Q \in \mathbb{R}^{n \times n}$ is a positive definite matrix. In this case, the optimization problems in Proposition IV.1 can be solved in closed-form.

**Proposition IV.9.** *(Sufficient Condition for CLF-BNCBF Compatibility for Single Integrator Dynamics and Circular Obstacles): Let $l \in [M]$, $\alpha_l > 0$, $x_0 \in \mathbb{R}^n \backslash \{q\}$, $q \in \mathcal{F}$, $\Gamma := \{x \in \mathbb{R}^n : V_q(x) \leq V_q(x_0)\}$, $B_l := \|q - c_l\|_Q^2 - 2\alpha_l r_l^2$,*

$$\beta_+ := \frac{\sqrt{B_l^2 + 4\alpha_l^2 r_l^2 (\|q - c_l\|^2 - r_l^2)} - B_l}{2\alpha_l r_l^2},$$

*and suppose that one of the following holds:*

- $\|x_0 - q\| - \|c_l - q\| > 0$ *and* $\frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|} > 1 + \frac{\|c_l - q\|}{r_l}$;
- $\|x_0 - q\| - \|c_l - q\| > 0$, $\frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|} \leq 1 + \frac{\|c_l - q\|}{r_l}$ *and* $\beta_+ \geq 1 + \frac{\|c_l - q\|}{r_l}$;
- $\|x_0 - q\| - \|c_l - q\| \leq 0$.

*Then, $V_q$ and $h_l$ are compatible in $\Gamma \cap \mathcal{F}$.*

*Proof.* We rely on Proposition IV.1. In the setting considered here, (6) reads as

$$\zeta_1 := \min_{x \in \Gamma, \beta \in \mathbb{R}} \|2\beta(x - c_l) - 2(x - q)\|^2 \tag{16a}$$

$$\text{s.t. } \beta \geq 0, \tag{16b}$$

$$\|x - c_l\|^2 - r_l^2 \geq 0. \tag{16c}$$

It follows that $\zeta_1 = 0$ if and only if there exists $x \in \Gamma$ and $\beta \in \mathbb{R} \backslash \{1\}$ (note that $\beta = 1$ and $\zeta_1 = 0$ are not possible because $q \in \mathcal{F}$) such that $x = \frac{1}{\beta - 1}(\beta c_l - q)$, $\beta \geq 0$ and $\|x - c_l\|^2 - r_l^2 \geq 0$. Equivalently, $\zeta_1 = 0$ if and only if there exists $\beta \in \mathbb{R} \backslash \{1\}$ such that $\beta \geq 0$, $|\beta - 1| \leq \frac{\|c_l - q\|}{r_l}$ and $\beta(\|x_0 - q\| - \|c_l - q\|) \geq \|x_0 - q\|$. Note that since $q \in \mathcal{F}$, $\|c_l - q\| \geq r_l$, and therefore the condition $\beta \geq 1 - \frac{\|c_l - q\|}{r_l}$ trivially holds if $\beta \geq 0$. Hence, $\zeta_1 = 0$ if and only if there exists $\beta \in \mathbb{R} \backslash \{1\}$ such that $\beta \geq 0$, $\beta \leq 1 + \frac{\|c_l - q\|}{r_l}$, and $\beta(\|x_0 - q\| - \|c_l - q\|) \geq \|x_0 - q\|$. We distinguish two cases: (i) suppose that $\|x_0 - q\| - \|c_l - q\| \leq 0$. Then, since $x_0 \neq q$, it follows that $\beta(\|x_0 - q\| - \|c_l - q\|) \geq \|x_0 - q\|$ can not hold. Therefore, $\zeta_1 \neq 0$ and $V_q$ and $h_l$ are compatible in $\Gamma$; (ii) suppose instead that $\|x_0 - q\| - \|c_l - q\| > 0$. Then, $\zeta_1 = 0$ if and only if $\frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|} \leq 1 + \frac{\|c_l - q\|}{r_l}$. Consequently, if $\frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|} > 1 + \frac{\|c_l - q\|}{r_l}$, then $V_q$ and $h_l$ are compatible in $\Gamma$. Consider then the case when $\frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|} \leq 1 + \frac{\|c_l - q\|}{r_l}$ so that $\zeta_1 = 0$. Then, (7) reads

$$\zeta_2 := \min_{\beta \in \mathbb{R} \backslash \{1\}} \frac{1}{(\beta - 1)^2} \hat{\Phi}(\beta) \tag{17a}$$

$$\text{s.t. } \frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|} \leq \beta \leq 1 + \frac{\|c_l - q\|}{r_l}, \tag{17b}$$

where $\hat{\Phi}(\beta) = \beta(\alpha_l \|q - c_l\|^2 - \alpha_l r_l^2 (1 - \beta)^2 - \beta(q - c_l)^T Q(q - c_l))$. By computing the roots of $\hat{\Phi}(\beta) = 0$, it follows that if $\beta_+ \geq 1 + \frac{\|c_l - q\|}{r_l}$, then $\hat{\Phi}(\beta) \geq 0$ for all $\beta \in [0, \beta_+]$, which implies that $\hat{\Phi}(\beta) \geq 0$ for all $\beta \in [\frac{\|x_0 - q\|}{\|x_0 - q\| - \|c_l - q\|}, 1 + \frac{\|c_l - q\|}{r_l}]$, from which it follows that $\zeta_2 \geq 0$ and $V_q$ and $h_l$ are compatible in $\Gamma$. $\square$

Proposition IV.9 provides a test for compatibility over a Lyapunov level set that only requires checking a set of algebraic conditions. Therefore, checking the compatibility of $V_q = \|x - q\|^2$ and $h_l(x) = \|x - c_l\|^2 - r_l^2$ over a Lyapunov sublevel set for a single integrator system can be done very efficiently.

## D. Compatibility Verification for Higher Relative Degree Systems

Here we extend the results of Section IV-A to a larger class of system dynamics and barrier functions, specifically High-Order Control Barrier Functions (HOCBFs) [53]. Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function defining a safe set of the form (3). Consider the situation where $h$ has to be differentiated $\bar{m} \in \mathbb{Z}_{>0}$ times along the dynamics (1) until the control $u$ appears explicitly (this is referred to as $m$ being the relative degree of $h$ under system (1), cf. [54]).

This means that, in order to ensure that the value of $h$ remains positive at all times (i.e., $\mathcal{C}$ is positively invariant), we need to reason with its higher-order derivatives. To do so, given differentiable extended class $\mathcal{K}_\infty$ functions $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(\bar{m}-1)}$, define a series of functions $\phi_0, \ldots, \phi_{\bar{m}-1} : \mathbb{R}^n \to \mathbb{R}$ as follows: $\phi_0 = h$ and

$$\phi_i(x) = L_f \phi_{i-1}(x) + \alpha^{(i)}(\phi_{i-1}(x)), \quad i \in \{1, \ldots, \bar{m}-1\}.$$

We further define sets $\mathcal{C}_1, \ldots, \mathcal{C}_{\bar{m}}$ as $\mathcal{C}_1 = \mathcal{C}$ and

$$\mathcal{C}_i = \{x \in \mathbb{R}^n : \phi_{i-1}(x) \geq 0\}, \quad i \in \{2, \ldots, \bar{m}\}.$$

The function $h$ is a high-order control barrier function (HOCBF) of $\mathcal{C}$ if one can find differentiable, extended class $\mathcal{K}_\infty$ functions $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(m)}$ such that, for all $x \in \mathcal{C} \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_{\bar{m}}$, there exists $u \in \mathbb{R}^m$ satisfying

$$L_f \phi_{\bar{m}-1}(x) + L_g \phi_{\bar{m}-1}(x)u + \alpha^{(\bar{m})}(\phi_{\bar{m}-1}(x)) \geq 0. \quad (18)$$

If $\bar{m} = 1$, this definition corresponds to the notion of CBF. According to [53, Theorem 5], any locally Lipschitz controller that satisfies (18) at each $x \in \mathcal{C} \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_{\bar{m}}$ renders the set $\mathcal{C} \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_{\bar{m}}$ positively invariant for system (1).

We next give an analogue of Definition II.4 for HOCBFs.

**Definition IV.10.** *(Compatibility of CLF-HOCBF pair): Let $\mathcal{D} \subset \mathbb{R}^n$ be open, $\mathcal{C} \subset \mathcal{D}$ be closed, $V$ a CLF on $\mathcal{D}$ and $h$ a HOCBF of $\mathcal{C}$. Then, $V$ and $h$ are compatible at $x \in \mathcal{C} \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_{\bar{m}}$ if there exists $u \in \mathbb{R}^m$ satisfying (2) and (18) simultaneously. We refer to both functions as compatible in a set $\tilde{\mathcal{D}}$ if they are compatible at every point in $\tilde{\mathcal{D}}$.*

The following result is an analogue of Proposition IV.1 for the case when $h$ is a HOCBF. Its proof follows an analogous argument and we omit it for space reasons.

**Proposition IV.11.** *(Characterization of CLF-HOCBF Compatibility): Given $q \in \mathcal{F}$, let $V_q : \mathbb{R}^n \to \mathbb{R}$ be a CLF of (1) with respect to $q$. Let $h$ be a HOCBF of $\mathcal{C}$ with relative degree $\bar{m} \in \mathbb{Z}_{>0}$. Let $W_q : \mathbb{R}^n \to \mathbb{R}$ be a positive definite function with respect to $q$ and $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(\bar{m})}$ be differentiable extended class $\mathcal{K}_\infty$ functions. For $\Gamma \subset \mathcal{R}$, let*

$$\zeta_1 = \min_{x \in \Gamma, \beta \in \mathbb{R}} \|\beta L_g \phi_{\bar{m}-1}(x) - L_g V_q(x)\|^2, \quad (19a)$$
$$s.t. \quad \beta \geq 0, \ \phi_i(x) \geq 0, \ i \in [\bar{m}-1]. \quad (19b)$$

*If $\zeta_1 \neq 0$, then $V_q$ and $h$ are compatible in $\Gamma \cap \mathcal{C} \cap \mathcal{C}_2 \cap \ldots \mathcal{C}_{\bar{m}}$. Otherwise, if $\zeta_1 = 0$, let*

$$\zeta_2 = \min_{x \in \Gamma, \beta \in \mathbb{R}} \tilde{\Phi}(x, \beta) \quad (20a)$$
$$s.t. \quad \beta \geq 0, \ \phi_i(x) \geq 0, \ i \in [\bar{m}-1], \quad (20b)$$

*where $\tilde{\Phi}(x, \beta) = -W_q(x) - L_f V_q(x) + \beta(L_f \phi_{\bar{m}-1}(x) + \alpha^{(\bar{m})}(\phi_{\bar{m}-1}(x)))$. If $\zeta_2 \geq 0$, then $V_q$ and $h$ are compatible in $\Gamma \cap \mathcal{C} \cap \mathcal{C}_2 \cap \ldots \mathcal{C}_{\bar{m}}$. Conversely, if $V_q$ and $h$ are compatible in $\Gamma \cap \mathcal{C} \cap \mathcal{C}_2 \cap \ldots \mathcal{C}_m$, then there exists a set of differentiable extended class $\mathcal{K}_\infty$ functions $\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(\bar{m})}$ and a positive definite function $W_q$ with respect to $q$ such that either $\zeta_1 \neq 0$ or $\zeta_1 = 0$ and $\zeta_2 \geq 0$.*

To conclude this section, we consider the case of double-integrator dynamics and circular obstacles. The double-integrator dynamics are given by

$$\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \mathbb{0}_k & \mathbb{I}_k \\ \mathbb{0}_k & \mathbb{0}_k \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} \mathbb{0}_k \\ \mathbb{I}_k \end{pmatrix} u, \quad (21)$$

with $k \in \mathbb{Z}_{>0}$ such that $n = 2k$, states $x \in \mathbb{R}^k$ and $v \in \mathbb{R}^k$, and input $u \in \mathbb{R}^k$. As pointed out in [55], only states of the form $(x_f, \mathbf{0}_k) \in \mathbb{R}^n$ are stabilizable for (21), and for any $x_f \in \mathbb{R}^k$, if we let $q = (x_f, \mathbf{0}_n)$, then $V_q : \mathbb{R}^n \to \mathbb{R}$ defined as $V_q(x, v) = \|x - x_f\|^2 + \|v\|^2 + (x - x_f)^T v$ is a CLF with respect to $q$. Next, consider $h : \mathbb{R}^n \to \mathbb{R}$ given by $h(x, v) = \|x - x_c\|^2 - r^2$, for some $x_c \in \mathbb{R}^k$ and $r > 0$. The following result shows that for this choice of $V$ and $h$, (19) and (20) take a tractable form.

**Corollary IV.12.** *(CLF-HOCBF Compatibility for Circular Obstacles and Double Integrator): Consider the double integrator dynamics (21). Let $q = (x_f, \mathbf{0}_k) \in \mathbb{R}^n$, and let $V_q(x, v) = \|x - x_f\|^2 + \|v\|^2 + (x - x_f)^T v$ be a CLF with respect to $q$, $W_q : \mathbb{R}^n \to \mathbb{R}$ a positive definite function with respect to $q$, and $h(x, v) = \|x - x_c\|^2 - r^2$ for some $x_c \in \mathbb{R}^k$, $r > 0$ a HOCBF. Let $\alpha_1 > 0$, $\alpha_2 > 0$, and $\phi_0 : \mathbb{R}^n \to \mathbb{R}$, $\phi_1 : \mathbb{R}^n \to \mathbb{R}$ defined as:*

$$\phi_0(x, v) = h(x),$$
$$\phi_1(x, v) = 2(x - x_c)^T v + \alpha_1(\|x - x_c\|^2 - r^2),$$

*and $\mathcal{C}_1 = \{(x, v) \in \mathbb{R}^{2n} : \phi_1(x, v) \geq 0\}$. For $\Gamma \subset \mathcal{R}$, let*

$$\hat{\zeta}_1 = \min_{x \in \Gamma, \beta \in \mathbb{R}, \tilde{x} \in \mathbb{R}^k} \|2\tilde{x} - 2v - (x - x_f)\|^2, \quad (22a)$$
$$s.t. \quad \beta \geq 0, \ \phi_i(x) \geq 0, \ i \in \{0, 1\}, \quad (22b)$$
$$\beta(x - x_c) - \tilde{x} \leq 0, \ \tilde{x} - \beta(x - x_c) \leq 0. \quad (22c)$$

*If $\hat{\zeta}_1 \neq 0$, then $V_q$ and $h$ are compatible in $\Gamma \cap \mathcal{C} \cap \mathcal{C}_1$. Otherwise, if $\hat{\zeta}_1 = 0$, let*

$$\hat{\zeta}_2 = \min_{\substack{(x,v) \in \Gamma, \beta \in \mathbb{R}, \\ \tilde{x} \in \mathbb{R}^k, \tilde{v} \in \mathbb{R}^k}} \hat{\Phi}(x, v, \tilde{x}, \tilde{v}) \quad (23a)$$
$$s.t. \quad \beta \geq 0, \ \phi_i(x) \geq 0, \ i \in \{0, 1\}, \quad (23b)$$
$$2\tilde{x} - 2v + x - x_f \leq 0, \quad (23c)$$
$$-2\tilde{x} + 2v - (x - x_f) \leq 0, \quad (23d)$$
$$\beta(x - x_c) - \tilde{x} \leq 0, \ \tilde{x} - \beta(x - x_c) \leq 0, \quad (23e)$$
$$\beta v - \tilde{v} \leq 0, \ -\beta v + \tilde{v} \leq 0, \quad (23f)$$

*where $\hat{\Phi}(x, v, \tilde{x}, \tilde{v}) = 2\tilde{v}^T v + \alpha_1 \tilde{x}^T v + 2\alpha_2 \tilde{x}^T v + \alpha_2 \alpha_1 \tilde{x}^T (x - x_c) - \alpha_1 \alpha_2 r^2 \beta - 2(x - x_f)^T v - \|v\|^2 - W_q(x, v)$. If $\hat{\zeta}_2 \geq 0$, then $V_q$ and $h$ are compatible in $\Gamma \cap \mathcal{C} \cap \mathcal{C}_1$.*

*Proof.* The result follows from Proposition IV.11 and by introducing the new variables $\tilde{x} = \beta(x - x_c)$, $\tilde{v} = \beta v$. $\square$

Note that (22) is a QCQP, and if $W_q$ is quadratic, (23) is also a QCQP and can therefore be solved efficiently [52].

## V. C-CLF-CBF-RRT

In this section, we introduce a novel motion planning algorithm, termed Compatible-CLF-CBF-RRT (C-CLF-CBF-RRT), that leverages the compatibility results from Section IV to generate collision-free paths that can be tracked using CLF-CBF based controllers.

### A. CLF-CBF Compatible Paths

We start by defining formally the type of paths that we seek to find using our motion planning algorithm. Intuitively, a path is *CLF-CBF compatible* if the CLF-CBF controller (5) successfully connects pairs of consecutive waypoints in the path.

**Definition V.1.** *(CLF-CBF Compatible Path): Let $\mathcal{A} = \{x_i\}_{i=1}^{N_a} \subset \mathcal{F}$ be a sequence of points, with $N_a \in \mathbb{Z}_{>0}$, $x_1 = x_{init}$ and $x_{N_a} \in \mathcal{X}_{goal} := \mathcal{B}(x_{goal}, \delta_{goal})$, where $x_{goal} \in \mathbb{R}^n$ and $\delta_{goal} > 0$. $\mathcal{A}$ is a CLF-CBF compatible path if for each $i \in [N_a - 1]$,*

(i) *there exists a CLF $V_i : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ with respect to $x_{i+1}$ in an open set containing $\Gamma_i := \{x \in \mathbb{R}^n : V_i(x) \leq V_i(x_i)\}$ for system (1);*

(ii) *there exist extended class $\mathcal{K}_\infty$ functions $\{\alpha_{i,l} : \mathbb{R} \to \mathbb{R}\}_{l \in [M]}$ and positive definite functions $W_i : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ with respect to $x_{i+1}$ such that the optimization problem*

$$\min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2 \qquad (24)$$
$$\text{s.t. } L_f h_{j,l}(x) + L_g h_{j,l}(x)u \geq -\alpha_{i,l}(h_{j,l}(x)),$$
$$\forall j \in \mathcal{I}_l(x), l \in [M],$$
$$L_f V_i(x) + L_g V_i(x)u + W_i(x) \leq 0.$$

*is feasible for all $x \in \Gamma_i \cap \mathcal{F}$.*

For each $i \in [N_a - 1]$, let $u_i^* : \Gamma_i \cap \mathcal{F} \to \mathbb{R}^m$ be a function mapping each $x \in \Gamma_i \cap \mathcal{F}$ to the solution of (24). Under the assumption that $u_i^*$ is locally Lipschitz, cf. Remark IV.5, the feasibility of (24) ensures that the solution of the closed-loop system $\dot{x} = f(x) + g(x)u_i^*(x)$ with initial condition $x_i$ (which we denote as $x(\cdot; x_i)$) is collision-free and asymptotically converges to $x_{i+1}$. Indeed,

(i) the satisfaction of the CLF constraint $L_f V_i(x) + L_g V_i(x)u + W_i(x) \leq 0$ at time $t \geq 0$ ensures that $\frac{d}{dt} V(x(t; x_i)) < 0$, and $x(t; x_i)$ asymptotically converges to $x_{i+1}$;

(ii) the satisfaction of the BNCBF constraint $L_f h_{j,l}(x) + L_g h_{j,l}(x)u \geq -\alpha_{i,l}(h_{j,l}(x))$ for all $j \in \mathcal{I}_l(x)$, $l \in [M]$ at time $t \geq 0$ ensures that $\frac{d}{dt} h_{j,l}(x(t; x_i)) \geq -\alpha_l(h_{j,l}(x(t; x_i)))$ for all $j \in \mathcal{I}_l(x)$, $l \in [M]$, and $x(t; x_i)$ is collision-free.

Because $x_i \in \Gamma_i \cap \mathcal{F}$, this ensures that as long as the CLF and BNCBF constraints are satisfied, $x(t; x_i) \in \Gamma_i \cap \mathcal{F}$. In turn, since the definition of CLF-CBF compatible path ensures that (24) is feasible for all $x \in \Gamma_i \cap \mathcal{F}$, this implies that the controller $u_i^*(x(t; x_i))$ is well-defined for all $t \geq 0$, and $x(\cdot; x_i)$ is collision-free and asymptotically converges to $x_{i+1}$. Therefore, CLF-CBF compatible paths guarantee that the controller obtained by solving (24) for each waypoint steers an agent obeying the dynamics (1) towards the next waypoint while remaining collision-free. Even though the convergence to the waypoint $x_{i+1}$ is only achieved in infinite time, one can execute the controller $u_i^*$ until the agent is sufficiently close to $x_{i+1}$ and then switch to the next controller $u_{i+1}^*$. We elaborate more on this point in Section VI, where we identify conditions on the CLF-CBF compatible path under which the controllers $\{u_i^*\}_{i=1}^{N_a - 1}$ can steer the agent from a neighborhood of each waypoint to a neighborhood of the next one, hence ensuring that (24) is feasible at all times if we switch to the next controller $u_{i+1}^*$ when the agent is sufficiently close to $x_{i+1}$.

**Remark V.2.** *(Controllability Requirements for CLF-CBF Compatible Paths):* Definition V.1 requires each of the points in the path $\mathcal{A}$ to be asymptotically stabilizable. This condition imposes some structural properties on the class of systems that admit such paths, which we examine next:

**Same number of inputs and state variables:** In the case when $m = n$ and $g(x)$ is invertible for all $x \in \mathbb{R}^n$, CLF-CBF compatible paths exist because any point $q \in \mathbb{R}^n$ is asymptotically stabilizable. Indeed, in this setting the function $V_q : \mathbb{R}^n \to \mathbb{R}$ defined by $V_q(x) = \frac{1}{2} \|x - q\|^2$ is a CLF with respect to $q$;

**Fewer inputs than state variables:** In the case when $m < n$, the set of stabilizable points is limited. For instance, for linear systems with $f(x) = Ax$ and $g(x) = B$, with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, only the points $q \in \mathbb{R}^n$ such that $Aq \in \text{Im}(B)$ are stabilizable. This is not a major restriction in a lot of cases of interest. For example, for a double-integrator system, where $m = k$ and $n = 2k$, with $k \in \mathbb{Z}_{>0}$, and

$$A = \begin{pmatrix} \mathbb{0}_k & \mathbb{I}_k \\ \mathbb{0}_k & \mathbb{0}_k \end{pmatrix}, \quad B = \begin{pmatrix} \mathbb{0}_k \\ \mathbb{I}_k \end{pmatrix},$$

this condition restricts the set of stabilizable points to those that have a zero velocity, but arbitrary position, as pointed out in Section IV-D. In general, if $m < n$, there often exists a smooth change of coordinates $\psi : \mathbb{R}^n \to \mathbb{R}^m$ that transforms the dynamics into a single integrator in $\mathbb{R}^m$. In [56, Section IV.A] and [57], for instance, this is achieved for unicycle dynamics, by taking the transformation $\psi(x_1, x_2, \theta) = [x_1 + l_0 \cos(\theta), x_2 + l_0 \sin(\theta)]$ (where $l_0 > 0$ is a positive design parameter). Then, for any $q \in \text{Im}(\psi)$, the set $M_q = \{x \in \mathbb{R}^n : \psi(x) = q\}$ can be asymptotically stabilized. Therefore, if $m < n$ but such a transformation $\psi$ exists, Definition V.1 can be adapted so that the points in $\mathcal{A}$ are in sets of the form $M_q$. •

### B. Algorithm Description

In this section we introduce the C-CLF-CBF-RRT algorithm, which builds upon RRT, cf. Section II-B, and generates CLF-CBF compatible paths. Algorithm 2 presents the pseudocode description.

The input for C-CLF-CBF-RRT consists of a compact, convex set $\mathcal{R} \subset \mathbb{R}^n$, an initial configuration $x_{init} \in \mathbb{R}^n$, a goal region $\mathcal{X}_{goal} \subset \mathbb{R}^n$, the number of iterations $k \in \mathbb{Z}_{>0}$ of the algorithm, the number of iterations $\tau \in \mathbb{Z}_{>0}$ for the

**Algorithm 2** C-CLF-CBF-RRT

---

1: **Parameters**: $\mathcal{R}$, $x_{\text{init}}$, $\mathcal{X}_{\text{goal}}$, $k$, $\tau$, $\eta$, $\{h_l, \alpha_l\}_{l=1}^M$
2: $\mathcal{T}.\text{init}(x_{\text{init}})$
3: **for** $i \in [1, \ldots, k]$ **do**
4:      $x_{\text{rand}} \leftarrow$ RANDOM_STATE()
5:      $x_{\text{near}} \leftarrow$ NEAREST_NEIGHBOR($x_{\text{rand}}, \mathcal{T}$)
6:      $x_{\text{new}} \leftarrow$ NEW_STATE($x_{\text{rand}}, x_{\text{near}}, \eta$)
7:      **if** not FREE_SPACE($x_{\text{new}}$) **then**
8:       **skip to next iteration**
9:      **end if**
10:     $V, W \leftarrow$ FIND_CLF($x_{\text{new}}$)
11:     **if** COMPATIBILITY($x_{\text{near}}, x_{\text{new}}, \tau, \{h_l, \alpha_l\}_{l=1}^M, V, W$)
    **then**
12:         $\mathcal{T}.\text{add\_vertex}(x_{\text{new}})$
13:         $\mathcal{T}.\text{add\_edge}(x_{\text{near}}, x_{\text{new}})$
14:         **if** $x_{\text{new}} \in \mathcal{X}_{\text{goal}}$ **then**
15:           **return** $\mathcal{T}$
16:         **end if**
17:     **end if**
18: **end for**
19: **return** $\mathcal{T}$

---

compatibility check, a set of extended class $\mathcal{K}_\infty$ functions $\{\alpha_l\}_{l=1}^M$, the steering parameter $\eta > 0$, and a set of obstacles $\{\mathcal{O}_l\}_{l=1}^M$ defined by functions $h_l : \mathbb{R}^n \to \mathbb{R}$ for $l \in [M]$. At the beginning, a tree $\mathcal{T}$ is initialized with a single node at $x_{\text{init}}$ and no edges.

The C-CLF-CBF-RRT algorithm operates similarly to the GEOM-RRT algorithm described in Section II-B.

At each iteration, steps 4:-6: are the same as in Algorithm 1. In general, RANDOM_STATE samples $\mathcal{R}$ uniformly, but if we know that only a subset of the points in $\mathcal{R}$ is stabilizable, one can choose to sample uniformly only over such points. The functions NEAREST_NEIGHBOR and NEW_STATE operate identically to how they do in GEOM-RRT. We note that, since $\mathcal{R}$ is convex, $x_{\text{new}}$ is guaranteed to belong to it. Next, the function FREE_SPACE checks whether $x_{\text{new}} \in \mathcal{F}$. If $x_{\text{new}} \notin \mathcal{F}$, it skips to the next iteration. Otherwise, FIND_CLF finds a CLF $V$ and associated positive definite function $W$ with respect to $x_{\text{new}}$. Then, the COMPATIBILITY function checks whether there exists a CLF-CBF based controller that steers the system from $x_{\text{near}}$ to $x_{\text{new}}$. If the COMPATIBILITY function returns a value of True, then $x_{\text{new}}$ is added as a vertex to $\mathcal{T}$ and is connected by an edge from $x_{\text{near}}$. If $x_{\text{new}} \in \mathcal{X}_{\text{goal}}$, there exists a single path in $\mathcal{T}$ from $x_{\text{init}}$ to $x_{\text{new}}$.

In Section V-C, we discuss in detail the definition of the function COMPATIBILITY. The function FIND_CLF aims to find a control Lyapunov function, which is a challenging problem for general control systems. Beyond what we noted in Remark V.2, one can use for this a variety of tools, such as sum-of-squares techniques [58], [59], neural networks [60], [61], or the learner-falsifier framework [62].

**Remark V.3.** *(Sampling in Systems with Fewer Inputs than State Variables):* A requirement for step 7: of Algorithm 2 to return a value of True is that $x_{\text{new}}$ is stabilizable. Since this point is obtained through random sampling, in general this might not be the case. However, if we know the set of points that are stabilizable (for instance, an $m$-dimensional manifold $\mathcal{M}$ in the case of systems with $m < n$ controls, cf. Remark V.2), then we can project $x_{\text{new}}$ onto such set.   •

### C. The COMPATIBILITY function

Here we define the operation of the COMPATIBILITY function. Given the CLF $V$ and the positive definite function $W$ with respect to $x_{\text{new}}$ found by FIND_CLF, it checks whether the optimization problem

$$\min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2, \tag{25}$$
$$\text{s.t. } L_f h_{j,l}(x) + L_g h_{j,l}(x)u \geq -\alpha_l(h_{j,l}(x)),$$
$$\forall j \in \mathcal{I}_l(x), l \in [M],$$
$$L_f V(x) + L_g V(x)u + W(x) \leq 0.$$

is feasible for all $x \in \Theta \cap \mathcal{F}$, where $\Theta = \{x \in \mathbb{R}^n \, : \, V(x) \leq V(x_{\text{near}})\}$ and $\alpha_l$ is the class $\mathcal{K}_\infty$ function associated with $h_l$.

*1. Find obstacles that intersect domain of interest:* To check whether (25) is feasible, we first find the obstacles that intersect $\Theta$, i.e., we find $l \in [M]$ such that $\text{Cl}(\mathcal{O}_l) \cap \Theta \neq \emptyset$. This can be done by solving the following optimization problem for every $l \in [M]$:

$$\min_{x \in \mathbb{R}^n} V(x) \tag{26}$$
$$\text{s.t. } h_{i,l}(x) \leq 0, \quad \forall i \in [N_l].$$

Then, $\text{Cl}(\mathcal{O}_l) \cap \Theta \neq \emptyset$ iff the optimal value of (26) is smaller than or equal to $V(x_{\text{near}})$. Problem (26) is tractable under the settings considered in Section IV, where $V$ is quadratic and the constraints are affine (in which case (26) is a quadratic program) or ellipsoidal (in which case (26) is a QCQP).

*2. Reduce number of constraints and check for compatibility:* Next, we check the compatibility of the CLF with each of the CBFs associated with the obstacles in $\mathcal{L} := \{l \in [M] \, : \, \Theta \cap \text{Cl}(\mathcal{O}_l) \neq \emptyset\}$ (Lemma A.1 ensures this step retains consistency). Then, COMPATIBILITY uses Proposition IV.1 for each $l \in \mathcal{L}$. First, for each $l \in \mathcal{L}$, it solves the optimization problem (6) with $\Gamma = \Theta$ and obtains the value $\zeta_{1,l}$. If $\zeta_{1,l} = 0$, it solves (7) with $\Gamma = \Theta$ and obtains the value $\zeta_{2,l}$. If for all $l \in \mathcal{L}$, the obtained values of $\zeta_{1,l}$ and $\zeta_{2,l}$ are such that $\zeta_{1,l} \neq 0$ or $\zeta_{1,l} = 0$ and $\zeta_{2,l} \geq 0$, then $V$ and $h_l$ are compatible in $\Theta \cap \mathcal{F}$ for all $l \in \mathcal{L}$ and COMPATIBILITY returns True.

*3. If unsuccessful, increase feasibility set and recheck:* Otherwise, it updates the set of extended class $\mathcal{K}_\infty$ functions and the function $W$ in a way that increases the feasible set of (25), and performs again the same check about its feasibility. In every subsequent iteration, we use a new $W$ obtained by multiplying the previous one by a constant factor $\sigma \in (0,1)$, and use linear extended class $\mathcal{K}_\infty$ functions $\alpha_l(s) = \alpha_{0,l}s$ with the parameter $\alpha_{0,l}$ being multiplied by a constant factor $\bar{\sigma} > 1$ at every iteration. With this choice, the objective function $\Phi$ of (7) does not decrease at any point, which means that the value of $\zeta_1$ remains the same but the condition $\zeta_2 \geq 0$ becomes

easier to satisfy, which makes it easier for COMPATIBILITY to return a value of True. If after $\tau$ of those updates the function still has not returned a value of True, it returns a value of False. We can also employ other heuristics to make it even easier for COMPATIBILITY to return a value of True. For example, instead of using constant factors $\sigma$, $\bar{\sigma}$, one can increase such factors at every iteration.

**Remark V.4.** *(No Loss of Generality in Assuming Linear Class $\mathcal{K}_\infty$ Function):* Since the set $\Theta$ is compact (because $V$ is proper), for each $l \in [M]$ and $j \in [N_l]$, the function $h_{j,l}$ is bounded in $\Theta$, i.e., there exists $M_{j,l} > 0$ such that $h_{j,l}(x) < M_{j,l}$ for all $x \in \Theta$. Now suppose that $V_q$ and $h_l$ are compatible in $\Theta$, i.e., there exists a controller $u_{\text{com}} : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$L_f h_{j,l}(x) + L_g h_{j,l}(x) u_{\text{com}}(x) + \alpha_l(h_{j,l}(x)) \geq 0, \ \forall j \in \mathcal{I}_l(x),$$
$$L_f V_q(x) + L_g V_q(x) u_{\text{com}}(x) + W(x) \leq 0,$$

for all $x \in \Theta$. Note that there exists $M_{\text{com}} > 0$ sufficiently large such that $M_{\text{com}} z > \alpha_l(z)$ for all $z \in [0, M_{j,l}]$. Using that $h_{j,l}(x) < M_{j,l}$ for all $x \in \Theta$, we deduce

$$L_f h_{j,l}(x) + L_g h_{j,l}(x) u_{\text{com}}(x) + M_{\text{com}} h_{j,l}(x) \geq 0, \ \forall j \in \mathcal{I}_l(x),$$
$$L_f V_q(x) + L_g V_q(x) u_{\text{com}}(x) + W(x) \leq 0,$$

for all $x \in \Theta$. Therefore, $V_q$ and $h_l$ are also compatible in $\Theta$ using a linear class $\mathcal{K}_\infty$ function $\alpha(z) = M_{\text{comp}} z$. Therefore, without loss of generality, we can assume that the class $\mathcal{K}_\infty$ function used in the COMPATIBILITY function is linear. •

## VI. ANALYSIS OF C-CLF-CBF-RRT

In this section we establish the probabilistic completeness of C-CLF-CBF-RRT. We do this by first showing that if C-CLF-CBF-RRT returns a tree with a vertex in $\mathcal{X}_{\text{goal}}$, then this tree contains a CLF-CBF compatible path; and then showing that, under suitable conditions, C-CLF-CBF-RRT in fact returns a tree with a vertex in $\mathcal{X}_{\text{goal}}$ with high probability.

**Proposition VI.1.** *(C-CLF-CBF-RRT and CLF-CBF Compatible Path):* Suppose that C-CLF-CBF-RRT returns a tree $\mathcal{T}$ that contains a vertex $q_{goal} \in \mathcal{X}_{goal}$. Then, the single path in $\mathcal{T}$ from $x_{init}$ to $q_{goal}$ is CLF-CBF compatible.

*Proof.* Let $N_a \in \mathbb{Z}_{>0}$ and $\mathcal{A} = \{x_i\}_{i=1}^{N_a}$ be the path obtained from C-CLF-CBF-RRT, with $x_1 = x_{\text{init}}$ and $x_{N_a} \in \mathcal{X}_{\text{goal}}$. First, FREE_SPACE ensures that $x_i \in \mathcal{F}$ for all $i \in [N_a]$. Moreover, FIND_CLF ensures that, for all $i \in [N_a - 1]$, there exists a CLF $V_i$ with respect to $x_{i+1}$, and COMPATIBILITY ensures that there exists a set of class $\mathcal{K}_\infty$ functions $\{\alpha_{i,l}\}_{l=1}^M$ and a positive definite function $W_i$ with respect to $x_{i+1}$ such that the optimization problem (24) is feasible for all points in the set $\{x \in \mathbb{R}^n : V_i(x) \leq V_i(x_i)\} \cap \mathcal{F}$. This ensures that $\mathcal{A}$ is CLF-CBF compatible. □

We next show that, under some extra assumptions, C-CLF-CBF-RRT returns a tree with a vertex in $\mathcal{X}_{\text{goal}}$ with probability one as the number of iterations $k$ goes to infinity. In doing so, our next result is critical as it provides conditions under which there exist neighborhoods around a CLF-CBF

compatible path for which points of two consecutive neighborhoods can be connected with a CLF-CBF-based controller.

**Lemma VI.2.** *(Compatibility in Neighboring Vertices): Let $\mathcal{A} = \{x_i\}_{i=1}^{N_a}$, $N_a \in \mathbb{Z}_{>0}$, be a CLF-CBF compatible path such that there exists $\delta_{clear} > 0$ with $\mathcal{B}(x_i, \delta_{clear}) \subset \mathcal{F}$ for all $i \in \{2, \ldots, N_a\}$. Let $\mathcal{N}_1 = \{x_{init}\}$. For each $i \in \{2, \ldots, N_a\}$, assume that there exist sets $\mathcal{N}_i$, with $x_i \in \mathcal{N}_i$, and $\hat{\Gamma}_i$, with $\Gamma_i \subset \hat{\Gamma}_i$ (and $\Gamma_i$ defined as in Definition V.1), satisfying the following properties:*

(i) *for each $y \in \mathcal{N}_i$, there exists a CLF $V_y : \hat{\Gamma}_i \to \mathbb{R}$ with respect to $y$ in $\hat{\Gamma}_i$ (with associated positive definite function $W_y$) and a bounded controller $\hat{u}_y : \hat{\Gamma}_i \to \mathbb{R}^m$ satisfying the corresponding CLF condition in $\hat{\Gamma}_i$;*

(ii) *there exists a bounded controller $u_i^* : \hat{\Gamma}_i \cap \mathcal{F} \to \mathbb{R}^m$ that satisfies the constraints in (24) for all points in $\hat{\Gamma}_i$ and, for each $y \in \mathcal{N}_i$,*

$$|(\nabla V_y(x) - \nabla V_i(x))^T (f(x) + g(x) u_i^*(x))|$$
$$< W_i(x), \quad (27)$$

*for all $x \in \mathcal{Z} = \{z \in \mathcal{F} : \exists l \in [M] \text{ s.t. } d(z, \mathcal{O}_l) \leq \frac{\delta_{clear}}{2}\}$;*

(iii) *for each $y_2 \in \mathcal{N}_i$ and $y_1 \in \mathcal{N}_{i-1}$, $\Gamma_{y_1,y_2} := \{x \in \mathbb{R}^n : V_{y_2}(x) \leq V_{y_2}(y_1)\} \subset \hat{\Gamma}_i$;*

(iv) *whenever $x_{new} \in \mathcal{N}_i$, global solutions to the optimization problems (6) and (7) in COMPATIBILITY are found.*

*Then, for each $i \in \{2, \ldots, N_a\}$, $y_2 \in \mathcal{N}_i$, and $y_1 \in \mathcal{N}_{i-1}$, there exists a set of extended class $\mathcal{K}_\infty$ functions $\{\bar{\alpha}_{i,l}\}_{l=1}^M$ and $\bar{\sigma} > 0$ (both dependent on $y_1$, $y_2$) such that, by taking $W_{y_2}^{\bar{\sigma}}(x) = \bar{\sigma} W_{y_2}(x)$, it holds that COMPATIBILITY$(y_1, y_2, 1, \{h_l, \bar{\alpha}_{i,l}\}_{l=1}^M, V_{y_2}, W_{y_2}^{\bar{\sigma}}) = $ True.*

*Proof.* Given $i \in \{2, \ldots, N_a\}$, $y_2 \in \mathcal{N}_i$, and $y_1 \in \mathcal{N}_{i-1}$, our goal is to show that there exists a set of extended class $\mathcal{K}_\infty$ functions $\{\bar{\alpha}_{i,l}\}_{l=1}^M$ and a sufficiently small $\bar{\sigma} > 0$ such that

$$\min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2, \quad (28)$$
$$\text{s.t. } L_f h_{j,l}(x) + L_g h_{j,l}(x) u \geq -\bar{\alpha}_{i,l}(h_{j,l}(x)),$$
$$\forall j \in \mathcal{I}_l(x), l \in [M],$$
$$\nabla V_{y_2}(x)^T (f(x) + g(x) u) + \bar{\sigma} W_{y_2}(x) \leq 0,$$

is feasible for all $x \in \Gamma_{y_1,y_2} \cap \mathcal{F}$. Figure 1 provides a visual aid for the argument that follows. The set $\Gamma_{y_1,y_2}$ is depicted in red, the sets $\mathcal{N}_i$ in blue, $\mathcal{Z}$ in light purple, and the obstacles $\{\mathcal{O}_l\}_{l=1}^M$ in green. For convenience, we let $T_{y_1,y_2} = \Gamma_{y_1,y_2} \cap \mathcal{Z}$ (depicted in dark purple).

**Feasibility on** $(\Gamma_{y_1,y_2} \setminus T_{y_1,y_2}) \cap \mathcal{F}$: Since $T_{y_1,y_2}$ contains all points that are closer than $\frac{\delta_{\text{clear}}}{2}$ from the boundary, there exists $h_0 > 0$ such that $h_{j,l}(x) > h_0$ for all $x \in (\Gamma_{y_1,y_2} \setminus T_{y_1,y_2}) \cap \mathcal{F}$, $l \in [M]$ and $j \in \mathcal{I}_l(x)$. Therefore, by taking $\alpha_{i,l}^* > 0$, with

$$\alpha_{i,l}^* > \frac{\sup\limits_{\substack{x \in (\Gamma_{y_1,y_2} \setminus T_{y_1,y_2}) \cap \mathcal{F}, \\ j \in \mathcal{I}_l(x)}} |L_f h_{j,l}(x) + L_g h_{j,l}(x) \hat{u}_{y_2}(x)|}{h_0},$$

for each $l \in [M]$ (which exists because $\hat{u}_{y_2}$ is bounded on $\hat{\Gamma}_i$ by (i)), it holds that

$$L_f h_{j,l}(x) + L_g h_{j,l}(x) \hat{u}_{y_2}(x) + \alpha_{i,l}^* h_{j,l}(x) \geq 0,$$

$$\forall j \in \mathcal{I}_l(x), l \in [M],$$
$$\nabla V_{y_2}(x)^T(f(x) + g(x)\hat{u}_{y_2}(x)) + \sigma W_{y_2}(x) \leq 0,$$

for all $x \in (\Gamma_{y_1,y_2} \backslash T_{y_1,y_2}) \cap \mathcal{F}$ and $\sigma \in (0,1)$, where we have used that $\hat{u}_{y_2}$ satisfies the CLF condition for $V_{y_2}$ by (i).

**Feasibility on** $T_{y_1,y_2}$: From (ii), there exists a bounded controller $u_i^*$ satisfying the constraints in (24) for all $x \in \hat{\Gamma}_i$. Since $\Gamma_{y_1,y_2} \subset \hat{\Gamma}_i$, cf. (iii), $u_i^*$ satisfies the constraints in (24) for all $x \in \Gamma_{y_1,y_2}$. Moreover, since (27) holds for all $x \in \mathcal{Z}$ (note that this is only possible because $\mathcal{B}(x_i, \delta_{\text{clear}}) \subset \mathcal{F}$ and therefore $x_i \notin \mathcal{Z}$, which means that the right-hand side of (27) is strictly positive), by (ii) it follows that

$$\nabla V_{y_2}(x)^T(f(x) + g(x)u_i^*(x)) < 0,$$

for all $x \in T_{y_1,y_2}$. Since $\mathcal{Z}$ is compact, this implies that there exists $\bar{\sigma} \in (0,1)$ sufficiently small such that

$$L_f h_{j,l}(x) + L_g h_{j,l}(x) u_i^*(x) + \alpha_{i,l}(h_{j,l}(x)) \geq 0,$$
$$\forall j \in \mathcal{I}_l(x), l \in [M],$$
$$\nabla V_{y_2}(x)^T(f(x) + g(x)u_i^*(x)) + \bar{\sigma} W_{y_2}(x) \leq 0.$$

for all $x \in T_{y_1,y_2}$.

Hence, by taking $\bar{\alpha}_{i,l}$ as an extended class $\mathcal{K}_\infty$ function such that $\bar{\alpha}_{i,l}(s) > \max\{\alpha_{i,l}(s), \alpha_{i,l}^* s\}$ for all $s \geq 0$, and $\bar{\sigma} \in (0,1)$ sufficiently small as described above, (28) is feasible for all $x \in \Gamma_{y_1,y_2} \cap \mathcal{F}$. Since COMPATIBILITY finds the global solutions of the optimization problems (6) and (7), cf. (iv), it follows that COMPATIBILITY$(y_1, y_2, 1, \{h_l, \bar{\alpha}_{i,l}\}_{l=1}^M, V_{y_2}, W_{y_2}^{\bar{\sigma}})$ = True (note that since (28) includes CBF constraints for $l \in [M]$, this argument is valid independently of the set $\mathcal{L}$ found by solving (26)). $\square$



Fig. 1: Visual aid for the arguments described in the proof of Lemma VI.2.

**Remark VI.3.** *(Verification of Assumptions of Lemma VI.2 for Specific Classes of Systems):* For systems with the same number of inputs as state variables, the set $\mathcal{N}_i$ in Lemma VI.2 can be taken as a ball centered at the waypoint $x_i$. As mentioned in Remark V.2, for such systems, $V_y(x) = \frac{1}{2}\|x - y\|^2$

is a CLF for any $y \in \mathbb{R}^n$. Moreover, we can take $W_y(x) = \|x - y\|^2$ and the controller $\hat{u} : \mathbb{R}^n \to \mathbb{R}^n$ defined as $\hat{u}(x) = -\frac{(x-y_2)^T f(x) + \|x - y_2\|^2}{\|g(x)^T(x-y_2)\|^2} g(x)^T(x - y_2)$ is such that $(x - y_2)^T(f(x) + g(x)\hat{u}(x)) + \|x - y_2\|^2 \leq 0$ for all $x \in \Gamma_{y_1,y_2}$ and is bounded, since

$$\|\hat{u}(x)\| \leq \frac{\|x - y_2\|(\|f(x)\| + \|x - y_2\|)}{\|g(x)^T(x - y_2)\|}$$
$$\frac{\|g(x)^{-1}g(x)(x - y_2)\|(\|f(x)\| + \|x - y_2\|)}{\|g(x)^T(x - y_2)\|}$$
$$\leq \|g(x)^{-1}\|\|x - y_2\|.$$

*Given that an explicit expression for the CLF is available, the conditions (ii), (iii) in Lemma VI.2 can be verified directly and one can choose the radius of the balls defining $\mathcal{N}_i$ to satisfy them. Furthermore, Propositions IV.7 and IV.9 provide two settings where condition (iv) holds.*

*A similar argument can be made for the double integrator in dimension $2k \in \mathbb{Z}_{>0}$. As mentioned in Remark V.2, in that case only the points of the form $(x_f, \mathbf{0}_k) \in \mathbb{R}^{2k}$ are stabilizable. Hence, the sets $\mathcal{N}_i$ in Lemma VI.2 can be taken in the form $\mathcal{N}_i := \{(x, \mathbf{0}_k) \in \mathbb{R}^{2k} : \|x - x_f\| < \nu_i\}$ for some $\nu_i > 0$. Furthermore, one can use the explicit expression of the CLF provided in Section IV-D and choose the parameters $\nu_i$ in order to verify the rest of the assumptions in Lemma VI.2.* •

In general, if the neighborhood $\mathcal{N}_i$ around $x_i$ in Lemma VI.2 is sufficiently small and $\nabla V_y$ is continuous in $y$ (with the assumption that $V_{x_i} = V_i$), the left-hand side of (27) can be made sufficiently small so that the inequality holds. Note that Assumptions (i), (iii), and (iv) are not restrictive and hold in several cases of interest, as outlined in Remark VI.3. Overall, the assumptions in Lemma VI.2 ensure that there exist neighborhoods around every waypoint of a CLF-CBF compatible path such that the controller obtained as the solution of (24) can connect a point from each neighborhood to any point in the neighborhood of the next waypoint. We next leverage this property to show the probabilistic completeness of C-CLF-CBF-RRT.

**Proposition VI.4.** *(Probabilistic Completeness of C-CLF-CBF-RRT):* Suppose that there exists a CLF-CBF compatible path $\mathcal{A} = \{x_i\}_{i=1}^{N_a}$, $N_a \in \mathbb{Z}_{>0}$, and suppose that all the assumptions in Lemma VI.2 regarding $\mathcal{A}$ hold. Further suppose that

(i) *there exists a positive probabiliy $p_i$ that RANDOM_STATE returns a point from $\mathcal{N}_i$;*

(ii) *for each $y \in \mathcal{N}_i$, FIND_CLF returns $V_y$ and $W_y$ (as defined in item (i) of Lemma VI.2);*

(iii) *the extended class $\mathcal{K}_\infty$ functions $\{\alpha_{i,l}\}_{i \in [N_a], l \in [M]}$ in (24) are upper bounded by linear extended class $\mathcal{K}_\infty$ functions, i.e., there exist $\hat{\alpha}_{i,l} > 0$ for $i \in [N_a]$ and $l \in [M]$ such that $\alpha_{i,l}(s) \leq \hat{\alpha}_{i,l} s$ for all $s \geq 0$;*

(iv) *the steering parameter $\eta$ in NEW_STATE is such that*
$$\eta > \max_{i \in [N_a-1]} \max_{y_2 \in \mathcal{N}_{i+1}, y_1 \in \mathcal{N}_i} \|y_2 - y_1\|.$$

*Then, there exists $\tau^* \in \mathbb{Z}_{>0}$ such that if $\tau > \tau^*$, the probability of C-CLF-CBF-RRT (executed with parameters $\tau, \eta,$ and any set of extended class $\mathcal{K}_\infty$ functions $\{\alpha_l\}_{l \in [M]}$)*

*returning a tree without a vertex in $\mathcal{X}_{goal}$ tends to zero as the number of iterations $k$ goes to infinity.*

*Proof.* The proof follows a similar reasoning to [48, Theorem 1] that proves probabilistic completeness for `GEOM-RRT`. Let $i \in [N_a - 1]$. First, we show that if $\mathcal{N}_i$ contains a vertex $x_{\text{near}}$ from the tree $\mathcal{T}$ in `C-CLF-CBF-RRT`, then with probability $p_i > 0$ in the next iteration a vertex will be added from $\mathcal{N}_{i+1}$. To see this, note that by assumption there exists a probability $p_i > 0$ that the function `RANDOM_STATE` returns a point $x_{\text{rand}}$ from $\mathcal{N}_{i+1}$. Given (iv), the distance between $x_{\text{near}} \in \mathcal{N}_i$ and $x_{\text{rand}} \in \mathcal{N}_{i+1}$ is less than $\eta$, and therefore $x_{\text{new}} = x_{\text{rand}}$. Now, Lemma VI.2 ensures that there exists a set of extended class $\mathcal{K}_\infty$ functions $\{\bar{\alpha}_{i,l}\}_{l=1}^M$, a CLF $V_{x_{\text{rand}}}$ with respect to $x_{\text{rand}}$ and a positive definite function $W_{x_{\text{rand}}}^{\bar{\sigma}}$ with respect to $x_{\text{rand}}$ such that `COMPATIBILITY`$(x_{\text{near}}, x_{\text{rand}}, \tau, \{h_l, \bar{\alpha}_{i,l}\}_{l=1}^M, V_{x_{\text{rand}}}, W_{x_{\text{rand}}}^{\bar{\sigma}})$ returns `True`. Moreover, since the functions $\{\alpha_{i,l}\}_{l=1}^M$ are upper bounded by linear extended class $\mathcal{K}_\infty$ functions with slopes $\{\hat{\alpha}_{i,l}\}_{l=1}^M$, by performing the updates in the extended class $\mathcal{K}_\infty$ functions described in Section V-C.3, it follows that there exists $\tau^*$ sufficiently large such that if $\tau > \tau^*$, the updated linear extended class $\mathcal{K}_\infty$ functions used in `COMPATIBILITY` have slopes larger than $\{\hat{\alpha}_{i,l}\}_{l=1}^M$ respectively and the coefficient multiplying $W_{x_{\text{rand}}}$ is smaller than $\bar{\sigma}$, which makes the `COMPATIBILITY` function return `True`. This means that $x_{\text{rand}}$ is added to $\mathcal{T}$ with the corresponding edge from $x_{\text{near}}$ to $x_{\text{rand}}$, as stated.

Next, in order for `C-CLF-CBF-RRT` to reach $\mathcal{X}_{\text{goal}}$ from $x_{\text{init}}$, the algorithm needs to successively select points from $\mathcal{N}_{i+1}$ as described previously for $i \in [N_a - 1]$. For $k$ iterations of `C-CLF-CBF-RRT`, this stochastic process can be described as $k$ Bernouilli trials [63, Definition 2.5] with success probabilities $\{p_i\}_{i=1}^{N_a-1}$. The algorithm reaches $\mathcal{X}_{\text{goal}}$ from $x_{\text{init}}$ after $N_a - 1$ successful outcomes. Let $p := \min_{i \in [N_a-1]} p_i$. Using the same argument as in [48, Theorem 1], the probability that this stochastic process does not have $N_a - 1$ successful outcomes after $k$ iterations is smaller than $\frac{(N_a-1)!}{(N_a-2)!} k^{N_a-1} e^{-pk}$. This means that the probability of `C-CLF-CBF-RRT` returning a tree without a vertex in $\mathcal{X}_{\text{goal}}$ tends to zero as the number of iterations $k$ goes to infinity. $\square$

**Remark VI.5.** *(Verification of Assumptions of Proposition VI.4):* As mentioned in Remark VI.3, for systems with the same number of inputs as state variables, *the set $\mathcal{N}_i$ in Lemma VI.2 can be taken as a ball centered at the waypoint $x_i$. If `RANDOM_STATE` samples $\mathcal{R}$ uniformly, it returns a point in such ball with probability equal to its relative volume in $\mathcal{R}$. Furthermore, in this case `FIND_CLF` can simply return $V_y(x) = \frac{1}{2} \|x - y\|^2$ and $W_y(x) = \|x - y\|^2$ for any $y \in \mathcal{N}_i$. For the* double integrator *in dimension $2k \in \mathbb{Z}_{>0}$, as mentioned in Remark VI.3, the sets $\mathcal{N}_i$ in Lemma VI.2 can be taken in the form $\mathcal{N}_i := \{(x, \boldsymbol{0}_k) \in \mathbb{R}^{2k} : \|x - x_f\| < \nu_i\}$ for some $\nu_i > 0$ and if `RANDOM_STATE` samples uniformly points of the form $(x_f, \boldsymbol{0}_k) \in \mathbb{R}^{2k}$, then (i) in Proposition VI.4 holds. Furthermore, `FIND_CLF` can return the explicit expression of the CLF provided in [55, Section V.A]. We note also that Assumption (iii) is not restrictive, and Assumption (iv) holds by taking the parameter $\eta$ sufficiently large.* •

**Remark VI.6.** *(Computational Complexity of `C-CLF-CBF-RRT`):* The computational complexity of `C-CLF-CBF-RRT` is the same as `GEOM-RRT` except for the added complexity of the `COMPATIBILITY` function. In general, the optimization problems (6), (7), and (26) required by `COMPATIBILITY` can be non-convex, which makes them not computationally tractable. However, in the setting considered in Proposition IV.7, the worst-case complexity of `COMPATIBILITY` is that of solving $\tau$ QCQPs, for which efficient heuristics exist [52]. In the setting considered in Proposition IV.9, (6), (7), and (26) can be solved in closed form, which means that `C-CLF-CBF-RRT` has the same computational complexity as `GEOM-RRT`. •

**Remark VI.7.** *(`C-CLF-CBF-RRT` for Differentially Flat Systems):* Here we explain how `C-CLF-CBF-RRT` is applicable to differentially flat systems. Differentially flat systems [64] are control systems for which the states and inputs can be written as algebraic functions of carefully selected *flat outputs* and their derivatives. Many robotic systems of interest, such as the unicycle [65] or the quadrotor [66] are differentially flat. This property facilitates the generation of smooth trajectories. Differentially flat systems are equivalent to dynamic feedback linearizable systems [67] (i.e., systems that can be feedback linearized after adding an appropriate number of dynamic inputs). This means that differentially flat systems can be transformed into linear systems after an appropriate change of coordinates and control inputs (the same also applies to static feedback linearizable systems, for which no dynamic inputs need to be added). Furthermore, by constructing an outer approximation of the obstacles using polytopes, and expressing it as a union of convex polytopes, the results in Proposition IV.7 apply, and the optimization problems (6) and (7) are easier to solve, cf. Section IV-B. •

**Remark VI.8.** *(Controller Execution):* Given a CLF-CBF compatible path $\mathcal{A}$, executing the controller (24) has the agent converge from one waypoint to the next asymptotically. However, under the assumptions of Proposition VI.4, there exist neighborhoods around the waypoints of $\mathcal{A}$ such that any two points of two consecutive neighborhoods can be connected with a CLF-CBF controller (possibly, with adjusted CLF, and extended class $\mathcal{K}_\infty$ functions, cf. Lemma VI.2). Therefore, by executing the controller (24) for a sufficiently large but finite time, the agent can visit these different neighborhoods and trace a path whose waypoints are close to those of $\mathcal{A}$. •

**Remark VI.9.** *(`C-CLF-CBF-RRT` for Higher-Relative Degree Systems):* `C-CLF-CBF-RRT` can be adapted to the setting where $h$ is a HOCBF, cf. Section IV-D, with the following modifications:

(i) $x_{\text{init}}$ and $\mathcal{X}_{\text{goal}}$ lie in $\mathcal{C} \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_{\bar{m}}$;
(ii) `RANDOM_STATE` returns states from $\mathcal{C} \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_{\bar{m}}$ (or a subset of it consisting of stabilizable points);
(iii) `COMPATIBILITY` employs the conditions described in Proposition IV.11 instead of those in Proposition IV.1 to check the compatibility of CLFs and HOCBFs. •

## VII. Simulation and Experimental Validation

Here we illustrate the performance of `C-CLF-CBF-RRT` in simulation and hardware experiments. Throughout the section, we deal with a differential-drive robot following the unicycle dynamics:

$$\dot{x} = v\cos(\theta), \tag{29a}$$
$$\dot{y} = v\sin(\theta), \tag{29b}$$
$$\dot{\theta} = \omega, \tag{29c}$$

where $s = [x, y] \in \mathbb{R}^2$ is the position of the robot, $\theta$ its heading, and $v$ and $\omega$ are its linear and angular velocity control inputs, respectively. Following [56, Section IV], we set

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad p = \begin{bmatrix} x \\ y \end{bmatrix} + l_0 R(\theta) e_1,$$

where $e_1 = [1, 0]^T$ and $l_0 > 0$ is a design parameter. This defines $p$ as a point orthogonal to the wheel axis of the robot. Moreover, let

$$L = \begin{bmatrix} 1 & 0 \\ 0 & 1/l_0 \end{bmatrix}.$$

Even though the dynamics (29) are nonlinear, it follows that $\dot{p} = R(\theta)L^{-1}u$, where $u = [v, w]^T$. By defining the new control input $\tilde{u} = R(\theta)L^{-1}u$, the state $p$ follows single integrator dynamics. The original angular and linear velocity inputs can be easily obtained from $\tilde{u}$ as $u = LR(\theta)^{-1}\tilde{u}$. Since $p$ can be made arbitrarily close to $[x, y]$ by taking $l_0$ sufficiently small, in what follows we consider $p$ as our state variable. Throughout the experiments, we use $\alpha_l(s) = 5s$ for all $l \in [M]$ and $\eta = 2m$. We also use $\tau = 5$ and constants $\sigma = 0.5$, $\bar{\sigma} = 2$ as defined in Section V. The results we present in this section have been obtained without the need to resort to increase the value of $\sigma$ or $\bar{\sigma}$ at every iteration, or perform other similar heuristics. Once the robot is within $0.5m$ of a given waypoint, we switch the controller so that it steers the robot towards the next waypoint.

### A. Computer Simulations

We have tested `C-CLF-CBF-RRT` in different simulation environments in a high-fidelity Unity simulator on an Ubuntu PC with Intel Core i9-13900K 3 GHz 24-Core processor. We utilize the function `minimize` from the library `SCIPY` [68] to solve the optimization problems in the `COMPATIBILITY` function. The robots used in the simulation are Clearpath Husky[1] robots, which have the same LIDAR and sensor capabilities as the real ones, and these are used to run a SLAM system that allows each robot to localize itself in the environment and obtain its current state, which is needed to implement the controller from (24). The first simulation environment consists of a series of red obstacles whose projection on the navigation plane is either a circle or a polytope. The second simulation consists of an environment with different rooms. The different walls are modeled as obstacles using nonsmooth CBFs, given that their projection on the navigation plane are

[1]Spec. sheets for the Husky and Jackal robots can be found at https://clearpathrobotics.com

quadrilaterals. To ensure that the whole physical body of the robot remains safe, we add a slack term to the CBF that takes into account the robot dimensions. For example, for a circular obstacle with center at $x_c \in \mathbb{R}^2$ and radius $r > 0$, and a circular robot with radius $r_0 > 0$, the CBF can be taken as $h(x) = \|x - x_c\|^2 - (r + r_0)^2$. Both simulation environments have dimensions $20m \times 50m$, and in each of them the projection of the obstacles in the navigation plane is either a circle or a polytope, so the `COMPATIBILITY` function runs efficiently (cf. Section IV). Figure 2 shows the tree generated by `C-CLF-CBF-RRT` in both simulation experiments, as well as the corresponding trajectory executed by the robot using the controller obtained as the solution of (24), which successfully reaches the end goal while remaining collision-free.
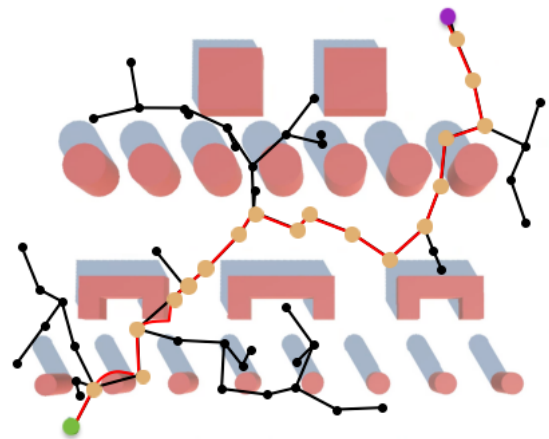


(a) Environment with obstacles
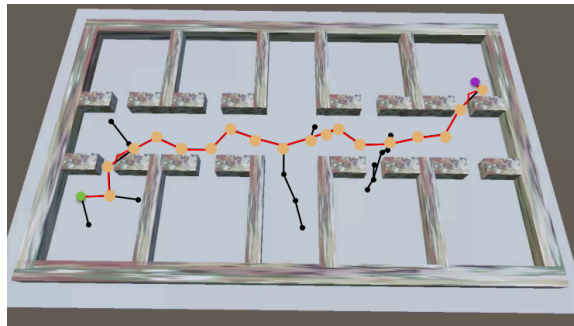


(b) Environment with rooms

Fig. 2: (a) First and (b) second simulation environment experiments. Tree generated by `C-CLF-CBF-RRT` (black), waypoints of the returned path (dark yellow) and trajectory followed by the robot using the controller from (24) (red). The starting point is the green dot and the end goal is the purple dot. In each environment, the robot successfully visits the waypoints while avoiding collisions with obstacles.
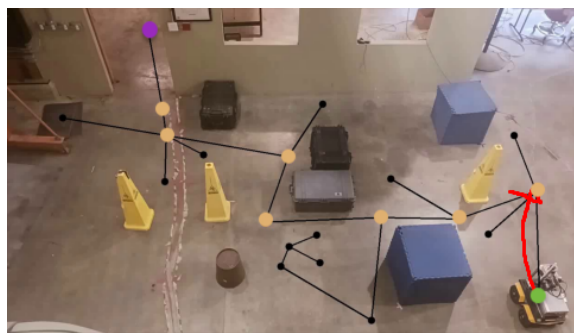
### B. Hardware Experiments

We have also tested `C-CLF-CBF-RRT` in a physical environment using a Clearpath Jackal robot. The robot is equipped with GPS, IMU and LIDAR sensors, which are used to run a SLAM system to localize its position in the environment and execute the controller from (24). The environment, with dimensions $4m \times 9m$, consists of different obstacles whose

projection on the navigation plane is either a circle or a polytope. We ensure the whole physical body of the robot remains safe using a slack term in the CBF formulation, as described in Section VII-A. Figure 3(a) shows the tree generated by C-CLF-CBF-RRT as well as the trajectory executed by the robot, successfully reaching its goal. We use $\alpha_l(s) = 5s$ for all $l \in [M]$ and choose $\eta = 2m$. Once the robot is within $0.5m$ of a given waypoint, we switch the controller so that it steers the robot towards the next waypoint.
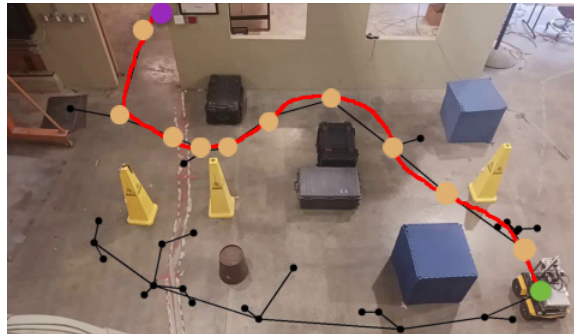


(a) C-CLF-CBF-RRT



(b) GEOM-RRT

Fig. 3: Execution of (a) C-CLF-CBF-RRT and (b) GEOM-RRT in the hardware experiment. In both plots, tree generated by the corresponding algorithm (black), waypoints of the returned path (dark yellow), and trajectory followed by the robot (red) using the controller from (24) (red). The starting point is the green dot and the end goal is the purple dot. The trajectory executed by the robot under C-CLF-CBF-RRT reaches its goal safely, whereas it fails under GEOM-RRT because it quickly encounters a point where the optimization problem (24) is infeasible.

### C. Comparison with GEOM-RRT

Here we compare the performance of C-CLF-CBF-RRT with GEOM-RRT in both the simulation and hardware environments. Figure 3(b) shows the tree generated by GEOM-RRT as well as the trajectory executed by the robot in the hardware environment using the controller obtained from (24). One can observe that the trajectory generated by the robot is unable to reach the end goal and stops rather early, at a point where the optimization problem (24) becomes infeasible. This occurs because GEOM-RRT does not take into account the dynamic feasibility of the path it generates.

We should point out that the steering parameter $\eta$ critically affects the performance of GEOM-RRT. To show this, we run various executions of GEOM-RRT in the simulation environment with obstacles depicted in Figure 2(a). Table I shows that

smaller values of $\eta$ yield a higher percentage of feasible paths but with a higher average execution time. For comparison, the average execution time of C-CLF-CBF-RRT, whose paths are always dynamically feasible, for the same simulation environment and with $\eta = 4m$, is 8.72 seconds. To match the dynamic feasibility of the produced paths, GEOM-RRT has to be run with $\eta = 1m$, at a significantly higher computational cost.

| $\eta$ (meters) | Percentage of feasible paths | Average execution time (seconds) |
|---|---|---|
| 1 | 100% | 154.36 |
| 2 | 90% | 140.62 |
| 4 | 50% | 130.62 |
| 8 | 30% | 4.83 |
| 16 | 5% | 1.84 |

TABLE I: Comparison of the percentage of feasible paths (i.e., paths for which the controller in (5) steers the robot from the initial point to the end goal by following the waypoints generated by the path) and the average execution time of GEOM-RRT (over 20 executions). The paths are generated for the simulation environment with obstacles in Figure 2(a).

**Remark VII.1.** *(Convergence to waypoints):* Since the robot asymptotically converges to each waypoint, we observe in the experiments that it tends to slow down when reaching a waypoint and speed up when switching to the next one. Here we describe ways in which this behavior can be alleviated:

(i) **Modifying the objective function**: The minimum-norm controller in (24) naturally seeks the smallest control action, which can lead to the observed *slowing down* effect near waypoints. Alternatively, given a nominal controller $u_{\text{nom}} : \mathbb{R}^n \to \mathbb{R}^m$ with a desired behavior (towards a waypoint or the end goal), one can modify the objective function in (24) by $\frac{1}{2} \|u - u_{\text{nom}}(x)\|^2$ and implement the resulting controller.

(ii) **Finite-time CLFs**: Fixed-time Control Lyapunov Function [69] can be used to design controllers that guarantee convergence to a desired waypoint within a specified time horizon. By extending the notion of compatibility to consider BNCBFs and finite-time CLFs, the optimization problems (6) and (7) can be reformulated using finite-time CLFs. Then, these optimization problems can be utilized to define a version of C-CLF-CBF-RRT that accounts for finite-time CLFs. We leave the study of the properties of such an algorithm for future work.

(iii) **CLF convergence rate**: given a closed-loop system satisfying the CLF condition (2), the function $W$ dictates the rate of decrease of trajectories to the origin. For example, by taking $W(x) = \gamma V(x)$, with $\gamma > 0$, trajectories of the closed-loop system converge to the origin at a rate $\gamma$. Therefore, by increasing $\gamma$, the rate of convergence can be increased. However, it should be pointed out that an increased rate of convergence might compromise the compatibility of $V$ with a CBF. ●

### D. Comparison with CBF-RRT and LQR-CBF-RRT⋆

Here we compare C-CLF-CBF-RRT with other related algorithms in the literature leveraging CBFs. First, we compare it with CBF-RRT, a sampling-based motion planning algorithm proposed in [42] that also employs control barrier functions.

Initially, `CBF-RRT` starts with a tree consisting of a single node in $x_{\text{init}}$. Then, each iteration of `CBF-RRT` operates as follows. First, it randomly samples a vertex $x_0$ from the current tree. Next, it generates a reference input, e.g., one steering the robot from $x_0$ to the goal set $\mathcal{X}_{\text{goal}}$ (cf. [42, Section 5] for more details). Finally, for a fixed period of simulated time $T_0$, at every state it executes the controller closest to the reference input that satisfies the CBF conditions associated to all obstacles. This quadratic optimization program is solved using the convex optimization library `CVXOPT` [70]. The state $x_{\text{new}}$ reached by the robot after this period of time $T_0$ gets added to the tree. To generate the trajectory, we numerically integrate the closed-loop system using the `odeint` method from the Python library `SCIPY` [68] and use a time discretization step of 0.005 seconds. We have ran multiple times `C-CLF-CBF-RRT` and `CBF-RRT` in the simulation environment with obstacles of Figure 2(a). Note that `CBF-RRT` is more computationally costly, as it requires running a trajectory for every new node added to the tree. Furthermore, this trajectory is generated by a controller that is obtained as the solution of an optimization problem at every point. In contrast, `C-CLF-CBF-RRT` only requires solving a single optimization problem (and, in the cases discussed in Section IV-C, not even that, since an algebraic check is enough) for every new node added to the tree. For example, if $T_0$ is small (e.g., $T_0 = 5$), the average execution time of `CBF-RRT` exceeds one minute. For $T_0 = 15$, the average execution time of `CBF-RRT` (over 10 different runs) is 384.58 seconds. The average execution time is similar for $T_0 = 10$, $T_0 = 20$. These numbers seem to indicate that smaller values of $T_0$ find a feasible path more rapidly, but such paths contain a larger number of waypoints. In contrast, larger values of $T_0$ lead to paths with a smaller number of waypoints but require more time to be found. In comparison, the average execution time of `C-CLF-CBF-RRT` with the same initial point and end goal (and with $\alpha_l(s) = 5s$ for all $l \in [M]$ and $\eta = 4m$) is 8.72 seconds, almost two orders of magnitude faster. We should also point out that there exists a trade-off between the computational complexity of CBF-RRT and the underlying safety guarantees. Indeed, since the CBF-QP controller cannot be solved continuously, CBF-RRT [42] solves the CBF-QP optimization problem periodically along the generated trajectory with sampling time $T_0$. As a consequence, in-between the times when the CBF-QP is solved, safety violations may occur. One way to remedy this is to solve the CBF-QP at a higher frequency. However, this increases the computational complexity of CBF-RRT, since the overall number of optimization problems to be solved is higher.

Finally, we compare `C-CLF-CBF-RRT` with `LQR-CBF-RRT*`. This is a sampling-based algorithm proposed in [44] which generates reference trajectories using LQR-based controllers of linearized dynamics around a new added node to the RRT, and checks the CBF condition at a finite set of points along this reference trajectory. The resolution with which such CBF condition is checked affects the safety of the overall trajectory (theoretically, it is safe only if every point satisfies the CBF condition). Table II compares different resolutions with which the CBF condition

checks are made, along with the corresponding average execution times (over 20 runs) and safety violations (which, to have a fair comparison with `C-CLF-CBF-RRT`, has been implemented without the adaptive sampling procedure described in [44, Section V.C]). Smaller resolutions naturally lead to larger execution times and a smaller percentage of safety violations. We note that a resolution of $0.01m$ leads to no safety violations and only has a slightly higher execution time compared to `C-CLF-CBF-RRT`. However, this lack of safety violations is not theoretically guaranteed in general and it is not known a priori what resolution results in no safety violations.

| Resolution (m) | Average execution time (s) | Safety violations |
|---|---|---|
| 0.5 | 0.26 | 60 % |
| 0.1 | 1.09 | 40 % |
| 0.05 | 1.7 | 5 % |
| 0.01 | 11.31 | 0 % |

TABLE II: Comparison of the resolution with which the CBF checks are made in `LQR-CBR-RRT*` and the corresponding average execution time (over 20 executions). The paths are generated for the simulation environment with obstacles in Figure 2(a).

## VIII. Conclusions

We have introduced `C-CLF-CBF-RRT`, a sampling-based motion planning algorithm that generates dynamically feasible collision-free paths from an initial point to an end goal. The algorithm creates a sequence of waypoints and results in a well-defined CLF-CBF-based controller that generates trajectories guaranteed to be safe and to sequentially visit the waypoints. These guarantees are based on a result of independent interest that shows that the problem of verifying whether a CLF and a BNCBF are compatible in a set of interest can be solved by finding the optimal value of an optimization problem. For systems with linear dynamics, quadratic CLFs, and CBFs of polytopic or ellipsoidal obstacles, this optimization problem is a QCQP, and for CBFs of circular obstacles, it can be solved in closed form. In these scenarios, this ensures that `C-CLF-CBF-RRT` can be executed efficiently. Finally, we have shown that `C-CLF-CBR-RRT` is probabilistically complete and can be generalized to systems where safety constraints have a high relative degree. Simulations and hardware experiments illustrate the performance and computational benefits of `C-CLF-CBR-RRT`. Future work will explore the extension of the results to other sampling-based algorithms (e.g., RRT*, bidirectional RRT), construct asymptotically optimal versions of `C-CLF-CBR-RRT`, and integrate available computational tools to find CLFs and verify the compatibility of CLF-CBF pairs. We also plan to explore techniques to alleviate the observed *slowing down* effect near waypoints, consider systems under uncertainty, both in the robot dynamics and the obstacles in the environment, and extend hardware implementations of `C-CLF-CBF-RRT` to more complex systems exploiting the notion of differential flatness.

## REFERENCES

[1] M. Diehl, H. G. Bock, H. Diegam, and P. B. Wieber, *Fast Motions in Biomechanics and Robotics*. New York: Springer, 2006.

[2] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Algarve, Portugal, 2012, pp. 1917–1922.

[3] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-Based Collision Avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2021.

[4] J. Tordesillas and J. P. How, "MADER: Trajectory planner in multiagent and dynamic environments," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 463–476, 2021.

[5] ——, "PANTHER: Perception-aware trajectory planner in dynamic environments," *IEEE Access*, vol. 10, pp. 22 662–22 677, 2022.

[6] W. Ding, W. Gao, K. Wang, and S. Shen, "An efficient B-spline-based kinodynamic replanning framework for quadrotors," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1287–1306, 2019.

[7] C. Richter, A. Bry, and N. Roy, *Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Environments*. New York: Springer, 2016.

[8] N. Csomay-Shanklin, W. D. Compton, and A. D. Ames, "Dynamically feasible path planning in cluttered environments via reachable bezier polytopes," *arXiv preprint arXiv:2411.13507*, 2023.

[9] T. Marcucci, M. Petersen, D. V. Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science Robotics*, vol. 8, no. 84, p. eadf7843, 2023.

[10] T. Marcucci, J. Umemberger, P. Parrillo, and R. Tedrake, "Shortest Paths in Graphs of Convex Sets," *SIAM Journal on Optimization*, vol. 34, no. 1, pp. 507–532, 2024.

[11] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional space," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[12] S. M. LaValle, "Rapidly-exploring random trees : a new tool for path planning," *The Annual Research Report*, 1998.

[13] J. Kuffner and S. LaValle, "RRT-connect: an efficient approach to single-query path planning," in *IEEE Int. Conf. on Robotics and Automation*, San Francisco, USA, 2000, pp. 995–1001.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[15] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu.

[16] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 5054–5061.

[17] L. Yi, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.

[18] E. Glassman and R. Tedrake, "A quadratic regulator-based heuristic for rapidly exploring state space," in *IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, 2010, pp. 5021–5028.

[19] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "Optimal sampling-based motion planning with automatically derived extension heuristics," in *IEEE Int. Conf. on Robotics and Automation*, St. Paul, USA, 2012, pp. 2537–2542.

[20] A. J. LaValle, B. Sakcak, and S. M. LaValle, "Bang-bang boosting of RRTs," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, Detroit, USA, 2023, pp. 2869–2876.

[21] L. Palmieri and K. O. Arras, "Distance metric learning for RRT-based motion planning with constant-time inference," in *IEEE Int. Conf. on Robotics and Automation*, Seattle, USA, 2015, pp. 637–643.

[22] Y. Li and K. E. Bekris, "Learning approximate cost-to-go metrics to improve sampling-based motion planning," in *IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011, pp. 4196–4201.

[23] W. J. Wolfslag, M. Bharatheesha, T. M. Moerland, and M. Wisse, "RRT-CoLearn: Towards kinodynamic planning without numerical trajectory optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1655–1662, 2018.

[24] H. L. Chiang, J. Hsu, M. Fiser, L. Rapia, and A. Faust, "RL-RRT: kinodynamic motion planning via learning reachability estimators from RL policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.

[25] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[26] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[27] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis*, vol. 7, no. 11, pp. 1163–1173, 1983.

[28] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: theory and applications," in *European Control Conference*, Naples, Italy, 2019, pp. 3420–3431.

[29] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[30] 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, 2017.

[31] L. Wang, A. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[32] P. Mestres and J. Cortés, "Optimization-based safe stabilizing feedback with guaranteed region of attraction," *IEEE Control Systems Letters*, vol. 7, pp. 367–372, 2023.

[33] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier functions based quadratic programming with application to adaptive cruise control," in *IEEE Conf. on Decision and Control*, 2014, pp. 6271–6278.

[34] S. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with applications to bipedal robot walking," in *American Control Conference*, Chicago, USA, July 2015.

[35] M. F. Reis, A. P. Aguilar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 731–736, 2021.

[36] X. Tan and D. V. Dimarogonas, "On the undesired equilibria induced by control barrier function based quadratic programs," *Automatica*, vol. 159, p. 111359, 2024.

[37] Y. Chen, P. Mestres, E. Dall'Anese, and J. Cortés, "Characterization of the dynamical properties of safety filters for linear planar systems," in *IEEE Conf. on Decision and Control*, Milan, Italy, 2024, pp. 2397–2402.

[38] W. S. Cortez and D. V. Dimarogonas, "On compatibility and region of attraction for safe, stabilizing control laws," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 7706–7712, 2022.

[39] P. Ong and J. Cortés, "Universal formula for smooth safe stabilization," in *IEEE Conf. on Decision and Control*, Nice, France, Dec. 2019, pp. 2373–2378.

[40] A. Ahmad, C. Belta, and R. Tron, "Adaptive sampling-based motion planning with control barrier functions," in *IEEE Conf. on Decision and Control*, Cancun, Mexico, Dec. 2022, pp. 4513–4518.

[41] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe navigation in human occupied environments using sampling and control barrier functions," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, Prague, Czech Republic, 2021, pp. 5794–5800.

[42] G. Yan, B. Vang, Z. Serlin, C. Belta, and R. Tron, "Sampling-based motion planning using control barrier functions," in *International Conference on Automation, Control and Robots*, Prague, Czech Republic, 2019, pp. 22–29.

[43] A. Manjunath and Q. Nguyen, "Safe and Robust Motion Planning for Dynamic Robotics via Control Barrier Functions," in *IEEE Conf. on Decision and Control*, Austin, USA, 2021, pp. 2122–2128.

[44] G. Yang, M. Cai, A. Ahmad, A. Prorok, R. Tron, and C. Belta, "LQR-CBF-RRT*: Safe and Optimal Motion Planning," *arXiv preprint arXiv:2304.00790*, 2023.

[45] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, 2nd ed., ser. TAM. Springer, 1998, vol. 6.

[46] R. A. Freeman and P. V. Kototovic, *Robust Nonlinear Control Design: State-space and Lyapunov Techniques*. Cambridge, MA, USA: Birkhauser Boston Inc., 1996.

[47] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth approach to controller synthesis for Boolean specifications," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5160–5174, 2021.

[48] M. Kleinbort, K. Solovey, Z. Littlefield, K. Bekris, and D. Halperin, "Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. i–vii, 2019.

[49] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[50] P. Mestres, A. Allibhoy, and J. Cortés, "Regularity properties of optimization-based controllers," *European Journal of Control*, vol. 81, p. 101098, 2025.

[51] H. Tuy, *Nonconvex Quadratic Programming*. New York: Springer, 1998.

[52] J. Park and S. Boyd, "General heuristics for nonconvex constrained quadratic programming," *arXiv preprint arXiv:1703.07870v2*, 2017.

[53] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *IEEE Conf. on Decision and Control*, Nice, France, Dec. 2019, pp. 474–479.

[54] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[55] G. Yang, C. Belta, and R. Tron, "Self-triggered control for safety critical systems using control barrier functions," in *American Control Conference*, Philadelphia, USA, Jul. 2019, pp. 4454–4459.

[56] 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.

[57] 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.

[58] W. Tan, "Nonlinear control analysis and synthesis using sum-of-squares programming," Ph.D. dissertation, University of California, Berkeley, 2006.

[59] H. Dai, C. Jian, H. Zhang, and A. Clark, "Verification and Synthesis of Compatible Control Lyapunov and Control Barrier Functions," in *IEEE Conf. on Decision and Control*, Milan, Italy, 2024, pp. 8178–8185.

[60] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural Lyapunov-barrier functions," London, UK, 2021.

[61] Y.-C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," in *Conference on Neural Information Processing Systems*, vol. 32, Vancouver, Canada, Dec. 2019, pp. 3240–3249.

[62] H.Ravanbakhsh and S. Sankaranarayanan, "Learning control Lyapunov functions from counterexamples and demonstrations," *Autonomous Robots*, vol. 43, pp. 275–307, 2019.

[63] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. John Wiley and Sons, 2004.

[64] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "On differentially flat nonlinear system," in *IFAC Symposium on Nonlinear Control Systems*, Bourdeaux, France, 1992, pp. 408–412.

[65] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011, pp. 2520–2525.

[66] L. E. Beaver and A. A. Malikopoulos, "Optimal control of differentially flat systems is surprisingly easy," *Automatica*, vol. 159, p. 111404, 2024.

[67] J. Lévine, "On the equivalence between differential flatness and dynamic feedback linearizability," *IFAC Proceedings Volumes*, vol. 40, no. 20, pp. 338–343, 2007.

[68] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[69] K. Garg, E. Arabi, and D. Panagou, "Fixed-time control under spatiotemporal and input constraints: A quadratic programming based approach," *Automatica*, vol. 141, p. 110314, 2022.

[70] M. S. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A python package for convex optimization, version 1.1.6." *Available at cvxopt.org, 2013*, 2013.

## APPENDIX

The following result shows that the problem of checking whether the optimization problem (25) is feasible can be simplified by only checking the pairwise feasibility of the CLF constraint and the CBF constraints associated with the obstacles that intersect with $\Theta$ (as defined in Section V-C).

**Lemma A.1.** (*Checking pairwise compatibility of a reduced set of CBFs*): *Let $x_{near} \in \mathbb{R}^n$, $x_{new} \in \mathbb{R}^n$, and $V : \mathbb{R}^n \to \mathbb{R}$ be*

a CLF with respect to $x_{new}$. Define $\Theta = \{x \in \mathbb{R}^n : V(x) \leq V(x_{near})\}$. Let $\mathcal{L} := \{l \in [M] : \Theta \cap \text{Cl}(\mathcal{O}_l) = \emptyset\}$. *Suppose that there exists a set of extended class $\mathcal{K}_\infty$ functions $\{\alpha_l\}_{l \in \mathcal{L}}$ such that for each $l \in \mathcal{L}$, the problem*

$$\min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2 \tag{30}$$
$$s.t. \ L_f h_{j,l}(x) + L_g h_{j,l}(x)u \geq -\alpha_l(h_{j,l}(x)), \ j \in \mathcal{I}_l(x),$$
$$L_f V(x) + L_g V(x)u + W(x) \leq 0,$$

*is feasible for all $x \in \Theta \cap \mathcal{F}$ and there exists a set of disjoint open sets $\{\mathcal{Y}_l\}_{l \in \mathcal{L}}$ (with $\mathcal{Y}_l$ being a neighborhood of $\partial \mathcal{O}_l$ satisfying $\mathcal{Y}_l \cap \mathcal{O}_{l'} = \emptyset$ for all $l' \neq l$) and a bounded controller $\hat{u}$ satisfying the constraints in (30) for each $x \in \mathcal{Y}_l \cap \Theta \cap \mathcal{F}$ and $l \in \mathcal{L}$. Then, there exists a set of extended class $\mathcal{K}_\infty$ functions $\{\bar{\alpha}_l\}_{l \in [M]}$ such that*

$$\min_{u \in \mathbb{R}^m} \frac{1}{2} \|u\|^2 \tag{31}$$
$$s.t. \ L_f h_{j,l}(x) + L_g h_{j,l}(x)u \geq -\bar{\alpha}_l(h_{j,l}(x)),$$
$$\forall j \in \mathcal{I}_l(x), l \in \mathcal{L},$$
$$L_f V(x) + L_g V(x)u + W(x) \leq 0.$$

*is feasible for all $x \in \Theta \cap \mathcal{F}$.*

*Proof.* Let $l \in \mathcal{L}$. Note that since $\mathcal{Y}_l \cap \mathcal{O}_{l'} = \emptyset$ for all $l' \in [M] \backslash \{l\}$, there exists $d_l > 0$ such that $h_{l'}(x) \geq d_l$ for all $l' \in [M] \backslash \{l\}$ and $x \in \mathcal{Y}_l \cap \Theta \cap \mathcal{F}$. Now, take $\hat{\alpha}_l > 0$ such that

$$\hat{\alpha}_l > \frac{\sup\limits_{x \in \mathcal{Y}_l \cap \Theta \cap \mathcal{F}} |L_f h_{j,l'}(x) + L_g h_{j,l'}(x)\hat{u}(x)|}{d_l}$$

for all $l' \in [M] \backslash \{l\}$ and $j \in \mathcal{I}_{l'}(x)$. Note that such $\hat{\alpha}_l$ exists because $\hat{u}$ is bounded and $\Theta$ is compact. Further let $\hat{\alpha} > \hat{\alpha}_l$ for all $l \in \mathcal{L}$, and take $\bar{\alpha}_l$ so that $\bar{\alpha}_l(s) > \max\{\alpha_l(s), \hat{\alpha}s\}$ for all $s \geq 0$. Now, $\hat{u}(x)$ is feasible for (31) for any $x \in \left(\bigcup_{l \in \mathcal{L}} \mathcal{Y}_l\right) \cap \Theta \cap \mathcal{F}$. On the other hand, there exists $d_{-1} > 0$ such that $h_l(x) > d_{-1}$ for all $l \in [M]$ and $x \in \Theta \cap \mathcal{F} \backslash \left(\bigcup_{l \in \mathcal{L}} \mathcal{Y}_l\right)$. Now, take $\hat{\alpha}_{-1} > 0$ such that

$$\hat{\alpha}_{-1} > \frac{\sup\limits_{x \in \Theta \cap \mathcal{F} \backslash (\bigcup_{l \in \mathcal{L}} \mathcal{Y}_l)} |L_f h_{j,l}(x) + L_g h_{j,l}(x)\hat{u}(x)|}{d_{-1}},$$

for all $l \in [M]$ and $j \in \mathcal{I}_l(x)$. Again, such $\hat{\alpha}_{-1}$ exists because $\hat{u}$ is bounded and $\Theta$ is compact. Further let $\hat{\alpha}_* > \max\{\hat{\alpha}, \hat{\alpha}_{-1}\}$ and take $\bar{\alpha}_l$ so that $\bar{\alpha}_l(s) > \max\{\alpha_l(s), \hat{\alpha}_* s\}$ for all $s \geq 0$. Hence, $\hat{u}(x)$ is feasible for (31) for any $x \in \Theta \cap \mathcal{F}$. $\square$

**Pol Mestres** received the Bachelor's degree in mathematics and the Bachelor's degree in engineering physics from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2020, and the Master's degree in mechanical engineering in 2021 from the University of California, San Diego, La Jolla, CA, USA, where he is currently a Ph.D. candidate. His research interests include safety-critical control, optimization-based controllers, distributed optimization and motion planning.

| Symbol | Meaning |
|---|---|
| **Dynamics** | |
| $n$ | state dimension |
| $m$ | input dimension |
| $f, g$ | vector fields defining dynamics |
| **CBF definitions** | |
| $M$ | number of obstacles |
| $\mathcal{O}_l$ | $l$-th obstacle |
| $h_l$ | BNCBF associated with $l$-th obstacle |
| $\mathcal{I}_l(x)$ | set of active indices for $h_l$ at $x \in \mathbb{R}^n$ |
| $\mathcal{F}$ | safe space |
| $N_l$ | number of functions defining $h_l$ |
| $h_{i,l}$ | $i$-th function defining $h_l$ |
| $\bar{m}$ | relative degree of a HOCBF |
| $\mathcal{Z}_{l,\mathcal{J}}$ | set of points where active constraints for $\mathcal{O}_l$ have indices in $\mathcal{J}$ |
| $\{\beta_i\}_{i \in \mathcal{J}}$ | optimization variables in 6, 7 |
| **CLF-CBF compatible path** | |
| $\mathcal{A} = \{x_i\}_{i=1}^{N_a}$ | CLF-CBF compatible path |
| $N_a$ | number of waypoints in $\mathcal{A}$ |
| $x_{\text{init}}$ | starting point of $\mathcal{A}$ |
| $\mathcal{X}_{\text{goal}}$ | goal set |
| $V_i$ | CLF with respect to waypoints $x_{i+1}$ |
| $\Gamma_i$ | $\{x \in \mathbb{R}^n : V_i(x) \le V_i(x_{i+1})\}$ |
| $W_i$ | positive definite associated with $V_i$ |
| $\alpha_{i,l}$ | extended class $\mathcal{K}_\infty$ associated to waypoint $x_{i+1}$ and $\mathcal{O}_l$ |
| **C$-$CLF$-$CBF$-$RRT** | |
| $\mathcal{T}$ | tree constructed in C$-$CLF$-$CBF$-$RRT |
| $x_{\text{rand}}$ | randomly sampled node |
| $x_{\text{new}}$ | new node to be added to $\mathcal{T}$ |
| $x_{\text{near}}$ | nearest node in $\mathcal{T}$ to $x_{\text{new}}$ |
| $k$ | number of iterations of C$-$CLF$-$CBF$-$RRT |
| $\eta$ | steering parameter in NEW_STATE |
| $\tau$ | number of updates of $\{\alpha_l\}_{l \in [M]}$ and $W$ in COMPATIBILITY |
| $V, W$ | CLF and associated positive definite function returned by FIND_CLF |
| **COMPATIBILITY** | |
| $\Theta$ | $\{x \in \mathbb{R}^n : V(x) \le V(x_{\text{near}})\}$ |
| $\mathcal{L}$ | $\{l \in [M] : \text{Cl}(\mathcal{O}_l) \cap \Theta \ne \emptyset\}$ |
| $\zeta_{1,l}$ | solution of (6) for $l \in \mathcal{L}$ |
| $\zeta_{2,l}$ | solution of (7) for $l \in \mathcal{L}$ |
| **Notation in Proposition VI.4** | |
| $\delta_{\text{clear}}$ | positive number such that $\mathcal{B}(x_i, \delta_{\text{clear}}) \subset \mathcal{F}$ |
| $\mathcal{N}_i$ | neighborhood of waypoint $x_i$ in Lemma VI.2 |
| $\hat{\Gamma}_i$ | superset of $\Gamma_i$ in Lemma VI.2 |
| $\Gamma_{y_1, y_2}$ | $\{x \in \mathbb{R}^n : V_{y_2}(x) \le V_{y_2}(y_1)\}$ |
| $\mathcal{Z}$ | $\mathcal{Z} = \{z \in \mathcal{F} : \exists l \in [M] \text{ s.t. } d(z, \mathcal{O}_l) \le \frac{\delta_{\text{clear}}}{2}\}$ |
| **Simulations** | |
| $R(\theta)$ | rotation matrix of angle $\theta$ |
| $v, \omega$ | linear and angular velocities in unicycle model |

TABLE III: Summary of symbols used in the paper.

computer vision, localization and mapping, autonomous navigation, and control in complex environments. He is a recipient of the 2022 Transactions on Robotics King-Sun Fu Memorial Best Paper Award.

**Jorge Cortés** (M'02, SM'06, F'14) received the Licenciatura degree in mathematics from Universidad de Zaragoza, Zaragoza, Spain, in 1997, and the Ph.D. degree in engineering mathematics from Universidad Carlos III de Madrid, Madrid, Spain, in 2001. He held postdoctoral positions with the University of Twente, Twente, The Netherlands, and the University of Illinois at Urbana-Champaign, Urbana, IL, USA. He was an Assistant Professor with the Department of Applied Mathematics and Statistics, University of California, Santa Cruz, CA, USA, from 2004 to 2007. He is a Professor and Cymer Corporation Endowed Chair in High Performance Dynamic Systems Modeling and Control at the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA. He is a Fellow of IEEE, SIAM, and IFAC. His research interests include distributed control and optimization, network science, nonsmooth analysis, reasoning and decision making under uncertainty, network neuroscience, and multi-agent coordination in robotic, power, and transportation networks.

**Carlos Nieto-Granda** is a Research Scientist in the Science of Intelligent Systems Division at the U.S. Army Research Laboratory (DEVCOM/ARL). He has obtained a B.S. degree in Electronics Systems from Tecnológico de Monterrey, Campus Estado de Mexico, Mexico; an M.S. degree in Computer Science from Georgia Institute of Technology; and a Ph.D. degree in Intelligent Systems, Robotics, and Control from University of California San Diego. His research interests include autonomous exploration, coordination, and decision-making for heterogeneous multi-robot teams focused on state estimation, sensor fusion,