2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

# Efficient Re-synthesis of Control Barrier Function via Safe Exploration

Ashkan Zehfroosh,[1] Zhaodan Kong,[1] and Stavros G. Vougioukas[2]

***Abstract*— This paper presents an efficient approach to incremental learning and updating of a valid control barrier function (CBF) so that it renders new explored safe area as its safe zone. For that purpose, we assume having access to sensor information (e.g. LiDAR sensor) that enables us to predict if a given location is immediately unsafe (e.g. close to an obstacle) or not. Using the sensor information, a set of predicted explorations over the potential safe regions is generated. The exploration data is then used to learn a valid CBF with enlarged safe zone. Toward this goal, we propose two methods: the first one is less conservative, as it possibly ends up with a larger safe zone, but it relies on a nonlinear optimization problem. A more computationally efficient alternative only requires Linear Programming at the cost of being more conservative.**

## I. INTRODUCTION

Consider a motion planning scenario using a safe reinforcement learning (RL) approach in which an agent tries to safely explore an unknown environment in order to learn an optimal policy. If a small safe region is provided to the agent initially (e.g. through expert demonstrations), the goal would be to safely expand the given initial safe set as much as possible so that the agent can explore more and more area and hence learn a better motion planning strategy. This paper addresses the problem of safely expanding a given initial safe set through a control barrier function (CBF) synthesis approach.

Control barrier function (CBF) is a powerful tool for systems that admit control inputs with the goal of determining consistent inputs that render a specific set forward invariant or asymptotically stable with respect to the system's dynamic [1]. CBFs have wide range of applications in safety-critical systems [2] among which one of the interesting ones is serving as a safety layer coupled with a learning-based control process so that the safety is guaranteed during the learning stage [3]–[5]. The main challenge however, is how to translate safety constraints into a valid CBF, or in other words, how to synthesise a valid CBF for various safety-critical systems [6].

Learning to synthesise CBF has recently attracted a lot of attention. A support vector machine approach is presented in [7] to learn a CBF from LiDAR sensor data, while the validity of the learned CBF has remained unchecked. As an alternative approach, a linear CBF is incrementally learned from expert demonstrations [8]; however, the correctness of the resulting CBF is not supported by a formal proof. An optimization based approach is proposed in [6] to synthesise a provably valid CBF from expert demonstrations. Robust and hybrid versions of the approach have also been developed [9]–[11]. However, the approach requires optimization over a function space. In addition, the final safe set will be restricted only to the area that is explored by the expert, which itself might be conservative. Building a Neural CBF is also done using labeled state-action pairs [12], but this approach can also lead to undesired or overly conservative behavior in practice [13].

This paper studies the possibility of efficient CBF update using safe exploration data in order to expand its safe set. The motivation of the work is in the context of analogy-based safe RL [14]. This branch of safe RL requires two types of information: an analogy function that can predict (probabilistically in general) the next state of an action, and the ability to recognize when a given state is immediately unsafe. These two types of information are usually available for motion planning purposes. The dynamic equation can play the role of the analogy function, and sensor information (e.g. LiDAR sensor) can be used to determine states that are immediately unsafe. The analogy-based RL process starts with an initial safe set. For each exploration, first the outcome of a possible action is predicted with the analogy function. If the predicted outcome is not immediately unsafe, then it is checked whether there exists a policy to bring the system back into the known safe set from that new state. If so, the intended exploration is allowed and the known safe set is expanded to cover those new states [14]. In the motion planning context, one way to implement the described step of returnability check to the safe set is via CBF-based approach. However, the existing CBF synthesis methods [6], [12] are usually designed for one-time construction of a valid CBF using the available data, that is often done through solving a constrained optimization problem. In other words, the existing solutions do not consider easy updates of the CBF without the need to resolve an optimization problem over the whole data set, each time new data is obtained. Therefore, the existing works are not compatible with the analogy-based safe RL scenario described above.

The general idea of expanding a (possibly) conservative safe set of a CBF has recently received attention [13], [15]. The work in [15] focuses on disturbed systems whose disturbance can be estimated by a Gaussian Process (GP). The structure of the workspace and the positions of all obstacles are assumed to be known, and the confidence interval of the GP is improved through safe exploration data; the improved model is used to expand the safe set. The assumptions related to the known

[1]Ashkan Zehfroosh, and Zhaodan Kong are with the Department of Mechanical and Aerospace Engineering, University of California, Davis, Davis, CA, USA. {azehfroosh, zdkong}@ucdavis.edu

[2]Stavros G. Vougioukas is with the Department of Biological and Agricultural Engineering University of California, Davis, Davis, USA. svougioukas@ucdavis.edu

This work has been supported by the USDA/NIFA AI Institute for Next Generation Food Systems (AIFS), USDA/NIFA award number 2020-67021-32855.

workspace hinder the application of this work to RL in which the workspace is typically considered unknown. The work in [13] uses Hamilton-Jacobi reachability analysis to expand a conservative safe set of a CBF; Again, the workspace and all obstacle positions are assumed to be known, and the method is reported to be able to handle systems that have maximum dimensions of 6. It is also reported that the method cannot locally push the boundary of the safe set, a limitation which is also mentioned as one of their plans for future work.

This paper tries to address the limitations associated with the above works. The assumptions of known workspace structure and obstacle positions are lifted; instead, the expansion of the safe set is done based on sensed local information, and the boundaries of the safe set are gradually and locally expanded. These features make the method appropriate for safe RL applications. Additionally, our work does not suffer from the curse of dimensionality and can handle high-dimensional systems. More specifically, this paper develops an incremental CBF synthesis approach to be used after each safe exploration. The process requires updating the CBF in a way that its safe set covers newly explored safe locations. Two methods are proposed for this purpose. The first one relies on a nonlinear optimization problem, while the second one emerges as a more computationally efficient but less flexible and more conservative alternative that only needs linear programming.

## II. Preliminaries

This section is dedicated to the necessary mathematical background needed for the subsequent technical discussions. The section briefly reviews the control system dynamics, CBFs, and some known results that will be utilized in following sections.

### A. Control System Dynamics

The exposition in this section starts with some minimal technical terminology and a description of the particular system dynamics that the proposed method applies to. To that end, let at time $t \geq 0$, $x(t) \in \mathbb{R}^n$ be the state of the system, $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ be its control input, and let the system's dynamics be expressed in the form

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

where functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz continuous.

Set $\mathcal{C}$ is said to be *forward invariant* for a given control law $u$ with respect to (1) if there exists a unique solution $x$ for (1) such that $x(0) \in \mathcal{C} \implies x(t) \in \mathcal{C}$, $\forall t \in [0, \infty]$.

### B. Control Barrier Functions

A CBF enables controller synthesis for dynamic systems such that if the system starts in a safe set, it will never leave the safe set, rendering the set forward invariant with respect to the dynamics of the system. A CBF can characterize the set of allowable control inputs that guarantee forward invariance of certain regions for a dynamical system at hand. The required control input is picked from a set defined in terms of the CBF, for example by solving an optimization problem in a sampled-data fashion [1].

For a domain $\mathcal{D} \subseteq \mathbb{R}^n$ and a set of interest $\mathcal{C} \subset \mathcal{D}$, a CBF appears in the form of a scalar differentiable function $\mathfrak{b} : \mathbb{R}^n \to \mathbb{R}$ for which

$$\mathcal{C} = \{x \in \mathcal{D} \mid \mathfrak{b}(x) \geq 0\} \ .$$

Mirroring Lyapunov stability analysis, one considers *candidate* barrier functions, which satisfy some basic structural requirements, and *valid* barrier functions, which have been shown to work as intended relative to the dynamics of the system at hand:

**Definition 1** ([2]). *A differentiable scalar function $\mathfrak{b} : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D} \subseteq \mathbb{R}^n$ is an open set, is a* candidate *control barrier function if the set $\mathcal{C} = \{x \in \mathcal{D} \mid \mathfrak{b}(x) \geq 0\}$ is nonempty.*

**Definition 2** ([6]). *A candidate control barrier function $\mathfrak{b} : \mathcal{D} \to \mathbb{R}$ is a* valid *control barrier function for (1) if there exists a locally Lipschitz extended class-$\mathcal{K}$ function $\alpha : \mathbb{R} \to \mathbb{R}$ (that is a monotonically increasing function with $\alpha(0) = 0$) such that for all $x \in \mathcal{C}$ it holds that*

$$\sup_{u \in \mathcal{U}} \Big\{ \nabla \mathfrak{b}(x) \cdot \big(f(x) + g(x)u\big) \Big\} \geq -\alpha\big(\mathfrak{b}(x)\big) \ , \tag{2}$$

*where $\cdot$ denotes the dot product of two vectors.*

Following Definition 2, a nonempty set of CBF consistent inputs for a valid CBF $\mathfrak{b}(x)$ at any $x \in \mathcal{C}$ is defined as:

$$K_{CBF}(x) = \\ \Big\{ u \in \mathcal{U} \mid \nabla \mathfrak{b}(x) \cdot \big(f(x) + g(x)u\big) + \alpha\big(\mathfrak{b}(x)\big) \geq 0 \Big\} \ . \tag{3}$$

Next comes the useful result of the CBF methodology:

**Lemma 1** ([6]). *Let $\mathfrak{b}(x)$ be a valid CBF for a compact set $\mathcal{C} \subset \mathcal{D}$ where $\mathcal{D}$ is an open set. Then starting from an $x(0) \in \mathcal{C}$ and inserting a locally Lipschitz control input $u(x) \in K_{CBF}(x)$ makes $\mathcal{C}$ forward invariant with respect to the control system (1), i.e. $x(t) \in \mathcal{C}$, $\forall t > 0$.*
*Additionally, starting from an $x(0) \in \mathcal{C}^c \cap \mathcal{D}$ (with $\mathcal{C}^c$ denoting the complement of $\mathcal{C}$) and inserting a locally Lipschitz $u(x) \in K_{CBF}(x)$ (in case that it exists) makes $\mathcal{C}$ asymptotically stable, i.e. $x(t)$ approaches $\mathcal{C}$ as $t \to \infty$.*

## III. Problem Statement

Assume that a valid CBF $\mathfrak{b}(x)$ is given for the control system (1) on the open set $\mathcal{D}$ that renders a compact set $\mathcal{C} \subset \mathcal{D}$ as forward invariant. We consider that $\mathfrak{b}(x)$ is constructed in a conservative manner, as its positive level set $\mathcal{C}$ only covers a relatively small portion of the actual safe region of $\mathcal{D}$ (see Fig. 1).

Our goal is to efficiently update $\mathfrak{b}(x)$ to come up with a new CBF that is less conservative by rendering a possibly larger portion of the actual safe region as its safe zone. Toward this purpose, we assume having access to sensor information (e.g. LiDAR sensor) that enables us to predict if a given location $x'$ is immediately unsafe (e.g. close to an obstacle) or not. For that, we set a safety threshold parameter $r$ such that we evaluate $x'$ as immediately unsafe if its closest distance to an obstacle is less than $r$.

Let $\gamma_e$, $e$ and $\epsilon$ (with $e \leq \epsilon$) be positive constants. Assume that the following set of predictions are made using the dynamic equation (1) and the CBF $\mathfrak{b}(x)$:

(i) Choose a not immediately unsafe $x_1 \notin \mathcal{C}$ with distance $e$ to the boundary of $\mathcal{C}$.

(ii) Starting from $x_1$, insert a control input $u(x_1)$ such that

$$\nabla\mathfrak{b}(x_1) \cdot \big(f(x_1) + g(x_1)u(x_1)\big) \geq -\alpha\big(\mathfrak{b}(x_1)\big) + \gamma_e \ .$$

(iii) At each timestep, repeat the above step and record the set of predicted locations $x_i$ (provided that they are not immediately unsafe) and inserted control inputs $u_i$, until the system returns back to the safe set at $x_{k+1} \in \mathcal{C}$ (this is guaranteed to happen according to Lemma 1). So for all $i = 2, \cdots, k$ we have

$$\nabla\mathfrak{b}(x_i) \cdot \big(f(x_i) + g(x_i)u(x_i)\big) \geq -\alpha\big(\mathfrak{b}(x_i)\big) + \gamma_e \ .$$

In addition, the timestep duration is such that $\|x_i - x_{i+1}\| \leq \epsilon$ forl all $i = 1, 2, \cdots, k$.

Then, our goal is to address the following problem:

**Problem 1.** *Consider a twice continuously differentiable* CBF $\mathfrak{b}(x)$ *on a domain* $\mathcal{D}$*, that renders a set* $\mathcal{C} \subset \mathcal{D}$ *forward invariant with respect to the control system* (1)*:*

$$\mathcal{C} = \{x \in \mathcal{D} \mid \mathfrak{b}(x) \geq 0\} \ ,$$
$$K_{CBF}(x) \neq \varnothing \qquad \forall x \in \mathcal{C} \ .$$

*where* $\mathcal{D}$ *is open and* $\mathcal{C}$ *is compact. Using a set of predictions* $\{(x_1, u_1), (x_2, u_2), \cdots, (x_k, u_k)\}$ *that is generated according to (i)–(iii), obtain a new valid* CBF $\mathfrak{b}_{new}(x)$ *that renders not only the previous safe set* $\mathcal{C}$ *but also the new predicted safe locations* $x_i$ *as its safe set* $\mathcal{C}_{new}$*.*

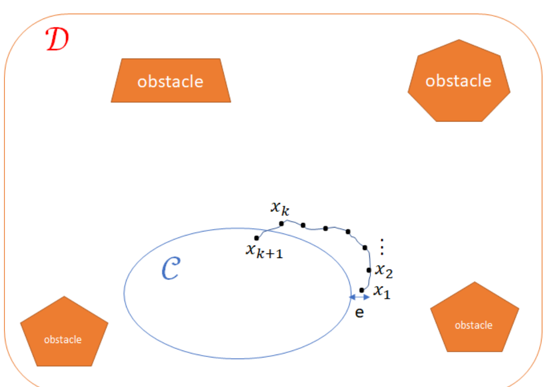


Fig. 1: The safe zone $\mathcal{C}$ of the existing valid CBF $\mathfrak{b}(x)$ covers only a small portion of the actual safe set inside $\mathcal{D}$. Our goal is to update the existing CBF to enlarge its safe zone so that it covers new predicted safe locations.

## IV. Technical Approach

In this section, we propose two solutions to Problem 1 described in Section III.

We set the new CBF as follows:

$$\mathfrak{b}_{new}(x) = \mathfrak{b}(x) + \sum_{i=1}^{k} z_i(x) \ , \tag{4}$$

where each $z_i(x)$ has the following form:

$$z_i(x) := \begin{cases} \frac{\gamma_{s_i}}{r_{s_i}^4}\big(r_{s_i}^2 - \|x - x_i\|^2\big)^2 & \|x - x_i\| \leq r_{s_i} \\ 0 & \|x - x_i\| > r_{s_i} \end{cases} ,$$

where parameters $\gamma_{s_i}$ and $r_{s_i}$ will be set subsequently in a way to guarantee the validity of the new CBF $\mathfrak{b}_{new}(x)$. But, before that, it is worthwhile to mention some features of the function $z_i(x)$ that will be used later. First of all, it is a continuously differentiable function and thus Lipschitz continuous. Since $z_i(x)$ is a scalar-valued function, any bound on its gradient norm can be used as its Lipschitz constant. Here, the smallest of such bound (denotes as $L_{z_i}$) can be parametrically found by computing the maximum gradient norm of $z_i(x)$:

$$L_{z_i} := \sup_{y_1, y_2 \in \mathcal{B}_{r_{s_i}}(x_i)} \frac{|z_i(y_1) - z_i(y_2)|}{\|y_1 - y_2\|} = \frac{8\gamma_{s_i}}{3\sqrt{3}r_{s_i}} \ , \tag{5}$$

where $\mathcal{B}_{r_{s_i}}(x_i)$ denotes the $r_{s_i}$-ball around $x_i$, i.e. $\mathcal{B}_{r_{s_i}}(x_i) = \{x \in \mathcal{D} \mid \|x - x_i\| \leq r_{s_i}\}$. Its gradient $\nabla z_i(x)$ is also Lipschitz continuous.

For $\mathfrak{b}_{new}(x)$ to be a valid CBF, a number of conditions on parameters $\gamma_{s_i}$ and $r_{s_i}$ are required which will be expressed through Theorem 1. Here, a number of notations are introduced that will be used in Theorem 1 and the subsequent analysis.

For $i = 1, 2, \cdots, k$, let $L_{\mathfrak{b}}(x_i)$ be the local Lipschitz bound of $\mathfrak{b}(x)$ within the $\epsilon$-ball around $x_i$ (i.e. $\mathcal{B}_\epsilon(x_i)$) that is defined similar to (5), and,

$$\begin{aligned} \Delta\alpha_i &:= \alpha\big(\mathfrak{b}_{new}(x_i)\big) - \alpha\big(\mathfrak{b}(x_i)\big) \ , \\ \gamma_{d_i} &:= \gamma_e + \Delta\alpha_i + \sum_{j=1}^{k} \nabla z_j(x_i) \cdot \big(f(x_i) + g(x_i)u(x_i)\big) \ , \\ &\qquad \text{for } i = 1, 2, \cdots, k \ . \end{aligned} \tag{6}$$

In addition, the following functions are defined with fixed $u(x_i)$:

$$q_i(x) := \nabla\mathfrak{b}_{new}(x) \cdot \big(f(x) + g(x)u(x_i)\big) + \alpha\big(\mathfrak{b}_{new}(x)\big) \ . \tag{7}$$

Note that $q_i(x)$ are Lipschitz continuous functions. Let also the local Lipschitz bound for each $q_i(x)$ be $L_{q_i}(x_i)$ within $\mathcal{B}_{r_{s_i}}(x_i)$.

Next comes the theorem that guarantees $\mathfrak{b}_{new}(x)$ to be a valid CBF under some conditions.

**Theorem 1.** *If the following conditions holds for* $i = 1, 2, \cdots, k$*:*

$$2\epsilon \leq r_{s_i} \leq r \ , \tag{8}$$

$$\frac{\gamma_{s_i}}{2} \geq \epsilon L_{\mathfrak{b}}(x_i) - \mathfrak{b}(x_i) \ , \tag{9}$$

$$\gamma_{d_i} \geq r_{s_i} L_{q_i}(x_i) \ , \tag{10}$$

*then, the function* $\mathfrak{b}_{new}(x)$ *as* (4) *is a valid* CBF *on* $\mathcal{C}_{new}$*, where*

$$\mathcal{C} \cup \Big(\bigcup_{i=1}^{k} \mathcal{B}_{\epsilon}(x_i)\Big) \ \subseteq \mathcal{C}_{new} \subseteq \ \mathcal{C} \cup \Big(\bigcup_{i=1}^{k} \mathcal{B}_{r_{s_i}}(x_i)\Big) \ . \tag{11}$$

*Proof:* We start by showing that $\mathfrak{b}_{new}(x) \geq 0$ for all $x \in \mathcal{C} \cup \big(\bigcup_{i=1}^{k} \mathcal{B}_{\epsilon}(x_i)\big)$. For $x \in \mathcal{C}$, the fact is trivial since $\mathfrak{b}(x) \geq 0$ and $z_i(x) \geq 0$ for all $i$. For $x \in \mathcal{B}_{\epsilon}(x_j)$, first note that, due to (8), we have that $z_j(x) \geq \frac{\gamma_{s_j}}{2}$, then we can write:

$$\begin{aligned}
\mathfrak{b}_{new}(x) &= \mathfrak{b}(x) + \sum_{i=1}^{k} z_i(x) \\
&\geq \mathfrak{b}(x) + z_j(x) \\
&= \mathfrak{b}(x) - \mathfrak{b}(x_j) + \mathfrak{b}(x_j) + z_j(x) \\
&\geq -L_{\mathfrak{b}}(x_j)\|x - x_j\| + \mathfrak{b}(x_j) + \frac{\gamma_{s_i}}{2} \\
&\geq -\epsilon L_{\mathfrak{b}}(x_j) + \mathfrak{b}(x_j) + \frac{\gamma_{s_i}}{2} \\
&\overset{(9)}{\geq} 0 \ .
\end{aligned}$$

In addition, we need to show that the positive level set of $\mathfrak{b}_{new}(x)$ does not contain any $x \notin \mathcal{C} \cup \big(\bigcup_{i=1}^{k} \mathcal{B}_{r_{s_i}}(x_i)\big)$ that could be potentially unsafe. For such $x$, we have that $z_i(x) = 0$ for all $i = 1, 2, \cdots, k$, thus $\mathfrak{b}_{new}(x) = \mathfrak{b}(x) < 0$.

The next step is to show that $\mathfrak{b}_{new}(x)$ is a valid CBF, i.e. showing that for all $x \in \mathcal{C}_{new}$, $K_{CBF}(x)$ (defined in (3)) is a nonempty set for $\mathfrak{b}_{new}(x)$. Toward this goal, first note that for the $q_i(x)$ function defined as (7), we have:

$$\begin{aligned}
q_i(x_i) = \Big(\nabla \mathfrak{b}(x_i) + \sum_{j=1}^{k} \nabla z_j(x_i)\Big) \cdot \big(f(x_i) + g(x_i)u(x_i)\big) \\
+ \Delta\alpha_i + \alpha\big(\mathfrak{b}(x_i)\big) \ \overset{(6)}{\geq} \gamma_{d_i} \quad i = 1, \cdots, k.
\end{aligned} \tag{12}$$

For $x \in \mathcal{B}_{r_{s_j}}(x_j)$, we can write:

$$\begin{aligned}
q_j(x) &= q_j(x_j) + q_j(x) - q_j(x_j) \\
&\overset{(12)}{\geq} \gamma_{d_j} - \big|q_j(x) - q_j(x_j)\big| \\
&\geq \gamma_{d_j} - L_{q_j}(x_j)\|x - x_j\| \\
&\geq \gamma_{d_j} - r_{s_j} L_{q_j}(x_j) \\
&\overset{(10)}{\geq} 0 \ .
\end{aligned}$$

The above fact holds for all $j = 1, 2, \cdots, k$, which implies that $\mathfrak{b}_{new}$ is a valid CBF on $\bigcup_{i=1}^{k} \mathcal{B}_{r_{s_i}}(x_i)$. For $x \in \mathcal{C} \setminus \bigcup_{i=1}^{k} \mathcal{B}_{r_{s_i}}(x_i)$, we have that $\mathfrak{b}_{new}(x) = \mathfrak{b}(x)$, which is already known to be a valid CBF. Hence, we showed that $\mathfrak{b}_{new}(x)$ is a valid CBF on $\mathcal{C}_{new} \subseteq \mathcal{C} \cup \big(\bigcup_{i=1}^{k} \mathcal{B}_{r_{s_i}}(x_i)\big)$, which completes the proof. □

As the first solution to Problem 1, we use Theorem 1 to design the following optimization problem for obtaining parameters $\gamma_{s_i}$ and $r_{s_i}$ which are needed to determine $\mathfrak{b}_{new}(x)$:

$$\min \sum_{j=1}^{k} \gamma_{s_j} + \frac{1}{r_{s_j}} \qquad \text{s.t. for all} \quad i = 1, 2, \cdots, k$$

$$2\epsilon \leq r_{s_i} \leq r \ , \tag{13a}$$

$$\frac{\gamma_{s_i}}{2} \geq \epsilon L_{\mathfrak{b}}(x_i) - \mathfrak{b}(x_i) \ , \tag{13b}$$

$$\gamma_{d_i} \geq r_{s_i} L_{q_i}(x_i) \ . \tag{13c}$$

The reason behind selecting the above objective function for minimization is to make $\mathfrak{b}_{new}(x)$ as smooth as possible and to cover more potentially safe area as much as possible. While the optimization problem (13) can be non-convex in general, it is a manageable problem, especially if the bound (13c) is excluded and instead verified using the bootstrapping method described in [6]. There might be cases when the above optimization problem does not have any solution. In such cases, one needs to step back and aim for a more conservative exploration outside $\mathcal{C}$ by selecting smaller $e$ and/or larger $\gamma_e$, and then retry (13) with the new set of predictions.

Alternatively, a more computationally efficient approach can be developed with the cost of having less flexible design and being more conservative. For that purpose, $r_{s_i}$ is dropped from the decision variable set of the optimization problem and fixed to some value consistent with (8). In addition, $\alpha(x)$ is assumed to be a linear function in its argument, i.e. $\alpha(x) = L_\alpha x$. Before stating the alternative solution, the following definition is made for easier notation:

$$d_i := \max_{x \in \mathcal{B}_{r_{s_i}}(x_i)} \ \|f(x) + g(x)u(x_i)\| \qquad i = 1, 2, \cdots, k \ .$$

In addition, similar to the function $q_i(x)$ in (7), assume the following functions with fixed $u(x_i)$ for $i = 1, 2, \cdots, k$:

$$p_i(x) := \nabla \mathfrak{b}(x) \cdot \big(f(x) + g(x)u(x_i)\big) + \alpha\big(\mathfrak{b}(x)\big) \ . \tag{14}$$

Note that since $\mathfrak{b}(x)$ is assumed to be twice continuously differentiable, $p_i(x)$ is a Lipschitz continuous function. Let the local Lipschitz bound for each $p_i(x)$ be $L_{p_i}(x_i)$ within $\mathcal{B}_{r_{s_i}}(x_i)$.

Now, the alternative solution to Problem (1) emerges as the following linear programming for obtaining $\gamma_{s_i}$:

$$\min \sum_{j=1}^{k} \gamma_{s_j} \qquad \text{s.t. for all} \quad i = 1, 2, \cdots, k$$

$$\frac{\gamma_{s_i}}{2} \geq \epsilon L_{\mathfrak{b}}(x_i) - \mathfrak{b}(x_i) \ , \tag{15a}$$

$$\gamma_e + \Delta\alpha_i \geq r_{s_i}\Big(L_{p_i} + L_\alpha \sum_{j=1}^{k} L_{z_j}\Big) + \sum_{j=1}^{k} d_j L_{z_j} \ . \tag{15b}$$

Note that (15b) is a linear inequality in terms of $\gamma_{s_i}$ since $\alpha$ is assumed to be linear function in its argument and $L_{z_j}$ (defined in (5)) is also linear in terms of $\gamma_{s_i}$.

**Proposition 1.** *Solution of the optimization problem* (15) *results in a valid* CBF $\mathfrak{b}_{new}$ *as* (4) *for* $\mathcal{C}_{new}$ *as* (11).

*Proof:* Showing that $\mathfrak{b}_{new}(x) \geq 0$ for all $x \in \mathcal{C}_{new}$ is similar to the first part of the proof of Theorem (1). To show that $\mathfrak{b}_{new}(x)$ is a valid CBF, first note that $\|\nabla z_i(x)\| \leq L_{z_i}$ and $\Delta\alpha_i = L_\alpha \sum_{j=1}^{k} z_j(x_i)$ for all $i = 1, 2, \cdots, k$. Then, For $x \in \mathcal{B}_{r_{s_j}}(x_j)$, we can write:

$$
\begin{aligned}
q_j(x) &\geq p_j(x) + L_\alpha \sum_{i=1}^{k} z_i(x) - \sum_{i=1}^{k} d_i L_{z_i} \\
&\geq p_j(x_j) + L_\alpha \sum_{i=1}^{k} z_i(x_j) + p_j(x) + L_\alpha \sum_{i=1}^{k} z_i(x) \\
&\quad - p_j(x_j) - L_\alpha \sum_{i=1}^{k} z_i(x_j) - \sum_{i=1}^{k} d_i L_{z_i} \\
&= p_j(x_j) + \Delta\alpha_j + p_j(x) + L_\alpha \sum_{i=1}^{k} z_i(x) \\
&\quad - p_j(x_j) - L_\alpha \sum_{i=1}^{k} z_i(x_j) - \sum_{i=1}^{k} d_i L_{z_i} \\
&\geq p_j(x_j) + \Delta\alpha_j - \sum_{i=1}^{k} d_i L_{z_i} \\
&\quad - \left| p_j(x) + L_\alpha \sum_{i=1}^{k} z_i(x) - p_j(x_j) - L_\alpha \sum_{i=1}^{k} z_i(x_j) \right| \\
&\geq \gamma_e + \Delta\alpha_j - \sum_{i=1}^{k} d_i L_{z_i} \\
&\quad - \left( L_{p_j} + L_\alpha \sum_{i=1}^{k} L_{z_i} \right) \|x - x_j\| \\
&\geq \gamma_e + \Delta\alpha_j - \sum_{i=1}^{k} d_i L_{z_i} - r_{s_j} \left( L_{p_j} + L_\alpha \sum_{i=1}^{k} L_{z_i} \right) \\
&\overset{(15b)}{\geq} 0 \ .
\end{aligned}
$$

Hence, similar to the proof of Theorem (1), it can be concluded that $\mathfrak{b}_{new}(x)$ is a valid CBF on $\mathcal{C}_{new}$. □

The above optimization can be solved repeatedly to gradually push the boundaries of the safe set: Each time a complete set of predictions (consisting $k$ points according to (i)–(iii)) is successfully generated and the optimization problem is solved to attain a new CBF. Then, the new CBF is used to generate another possible set of predictions for further expansion of the safe set. The process stops when either no further safe predicted exploration data can be generated (e.g., when the boundaries of the safe zone get too close to obstacles or when the required control input exceeds the allowed bound) or when the constraints of (13) or (15) cannot be satisfied. Re-solving either (13) or (15) to get a new CBF $\mathfrak{b}_{new}(x)$ as in (4) adds a number of $z_i(x)$ functions after each predicted exploration set. The total number of functions could get large after numerous explorations rendering the safety analysis difficult. However, at any location in the workspace, only $z_i(x)$ functions whose peak points are within $2r$ distance are actually needed to be considered for the safety analysis since all other $z_i(x)$ functions play no role in the safety analysis as their values - as well as their gradients - vanish at that location.

## V. Simulation Results

This section showcases the utilization of the CBF synthesis approach described in Section IV. We refer to the CBF synthesis based on the optimization problem (13) as Method 1 and based on (15) as Method 2. Consider a 2D sphere world where the workspace (i.e. set $\mathcal{D}$) is a circle centered at $(0, 0)$ with radius of 3. There are two circular obstacles centered at $(1.5, 1.5)$ and $(-1.5, -1.5)$ respectively with radius of 0.5. The initial CBF $\mathfrak{b}(x)$ is given as follows:

$$
\mathfrak{b}(x) = \frac{1}{100}\left(1 - \|x - (1, -1)\|^2\right) \ . \tag{16}
$$

The dynamic system is assumed to be point robot, i.e. $\dot{x} = u$, where $u$ is the control input.

Fig. 2 depicts the initial safe set $\mathcal{C}$. As it can be seen, the above initial CBF is conservative as it renders only a relatively small portion of the actual safe areas of $\mathcal{D}$ as its safe zone $\mathcal{C}$.

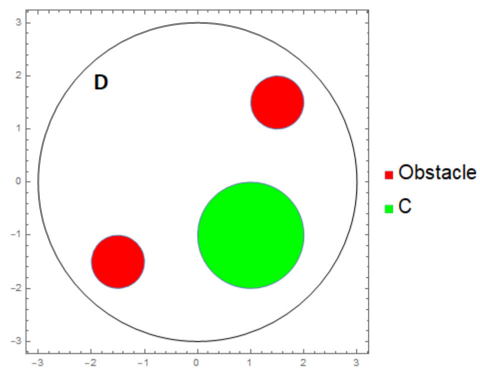


Fig. 2: The initial safe set $\mathcal{C}$ of the CBF $\mathfrak{b}(x)$ (16) covers only a small portion of actually safe regions of the workspace $\mathcal{D}$.

The approach outlined in Section IV is utilized to expand the initial safe set as much as possible. For that purpose, sets of safe predicted exploration data are generated according to (i)–(iii) (see Section III). Then, the updated CBF with enlarged safe zone is synthesised based on (13) or (15). The process is repeated until no further safe set expansion is possible. To do this, an extended class-$\mathcal{K}$ function $\alpha(x) = \frac{1}{100}x$ is selected and the following parameters are set: $r = 0.1$, $\epsilon = 0.01$. In addition, the allowed control input is constrained as $\|u\| \leq 1$, and the minimum allowed value for parameter $e$ is set to 0.005.

We should reiterate that (i)–(iii) are predicted explorations, and so the expansion of the safe set is only done whenever a set of predictions with a reasonably small $k$ can be generated. In fact, a large $k$ for the predictions set at an intended location suggests that the system is either near unsafe regions or the available control resources cannot bring the system back to the current known safe set. So, expansion of the safe set at that location might not be desired or possible. In such cases, the system keeps searching for good sets of predictions on other locations. For this simulation example, we set the maximum allowed $k$ for any predictions set as 5.

Fig. 3 shows the updated safe zone after one iteration of the process described above. For this first iteration, the following parameters are used: $\gamma_e = 0.01$, $e = 0.01$. In addition, a fixed $r_{s_i} = 0.06$ is used for Method 2. As expected, Method 1 expands the safe zone a little larger since it has the flexibility of choosing the maximum compatible $r_{s_i}$ as opposed to Method 1 that drops $r_{s_i}$ from the decision variable set of the optimization problem and uses a fixed value for it instead.

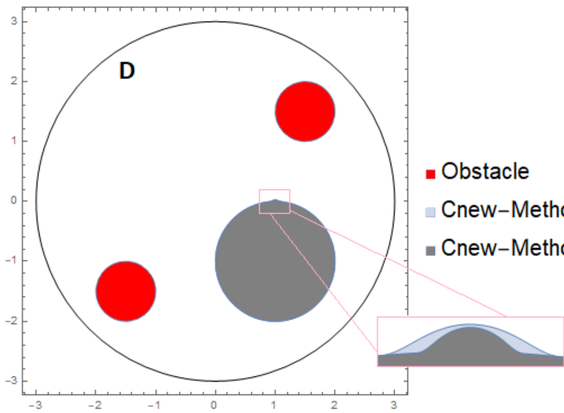


Fig. 3: The expanded safe zone $\mathcal{C}_{new}$ after one iteration of Method 1 and Method 2.

Continuation of the above process results in expanding the safe zone more and more after each iteration. The final expanded safe zones using Method 1 and 2 are shown in Fig. 4. Similar to the results of the first iteration, it can be seen that the final expanded safe zone by Method 1 is a little larger than Method 2 due to its higher flexibility. However, the reduced computational effort of Method 2 (that only needs linear programming in each iteration) can be a fair compromise given that the difference between the two final safe zones is relatively small for this simulation study. In particular, on a system with an Intel Core i7 (2.6GHz) processor and 16GB of memory using Wolfram Mathematica, the first method takes $2045.2$ seconds, while the second method is done in $487.3$ seconds. Hence, the first method takes about $325\%$ longer time while it enlarges the safe set only by about $5\%$ more than the second method.

## VI. Conclusion

It is possible to efficiently re-synthesize a CBF to enlarge its safe zone based on the safe exploration data. For this purpose, two approaches are presented in this paper. The first one provides a more flexible design but relies on a generally non-convex optimization problem, while the second one only requires linear programming at the cost of a more conservative design. The formal proof for the correctness of both methods as well as a supporting numerical example are provided and discussed. The main application that can benefit from this work is reinforcement learning (RL) for motion planning purposes. The utilization of the presented approaches in this context will be the focus of our future work.

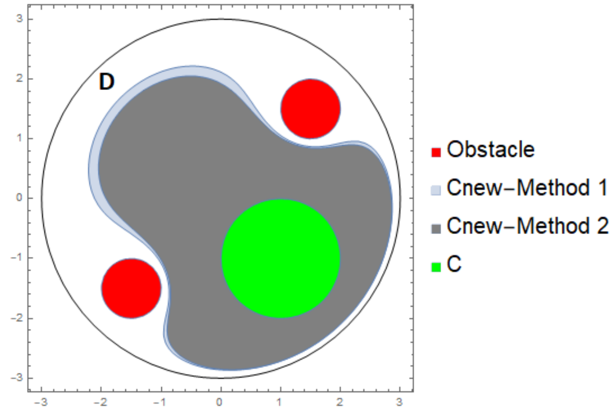


Fig. 4: The final expanded safe zone $\mathcal{C}_{new}$ of the two methods.

## References

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European Control Conference*, 2019, pp. 3420–3431.

[3] Y. Emam, G. Notomista, P. Glotfelter, Z. Kira, and M. Egerstedt, "Safe reinforcement learning using robust control barrier functions," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.

[4] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[5] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[6] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *IEEE Conference on Decision and Control*. IEEE, 2020, pp. 3717–3724.

[7] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7139–7145.

[8] M. Saveriano and D. Lee, "Learning barrier functions for constrained motion planning with dynamical systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 112–119.

[9] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. Dimarogonas, S. Tu, and N. Matni, "Learning hybrid control barrier functions from data," in *Conference on Robot Learning*. PMLR, 2021, pp. 1351–1370.

[10] A. Robey, L. Lindemann, S. Tu, and N. Matni, "Learning robust hybrid control barrier functions for uncertain systems," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 1–6, 2021.

[11] L. Lindemann, A. Robey, L. Jiang, S. Tu, and N. Matni, "Learning robust output control barrier functions from safe expert demonstrations," *arXiv preprint arXiv:2111.09971*, 2021.

[12] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural lyapunov-barrier functions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1724–1735.

[13] S. Tonkens and S. Herbert, "Refining control barrier functions through hamilton-jacobi reachability," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 355–13 362.

[14] M. Roderick, V. Nagarajan, and Z. Kolter, "Provably safe pac-mdp exploration using analogies," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1216–1224.

[15] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2460–2465.