

Generalizations of Backup Control Barrier Functions: Expansion and Adaptation for Input-Bounded Safety-Critical Control

David E. J. van Wijk¹, Dohyun Lee¹, Ersin Daş², Tamas G. Molnar³, Aaron D. Ames¹, Joel W. Burdick¹

Abstract—Guaranteeing the safety of nonlinear systems with bounded inputs remains a key hurdle in safe autonomy. Backup control barrier functions (bCBFs) provide a powerful mechanism for constructing controlled invariant sets by propagating trajectories under a pre-verified backup controller to a forward invariant backup set. While effective, standard bCBFs use the same controller for both set expansion and safety certification, which can restrict the expanded safe set and lead to conservative behavior. In this study, we generalize bCBFs by separating the set-expanding controller from the verified backup controller, thereby enabling a broader class of expansion strategies while preserving formal safety guarantees. We establish sufficient conditions for forward invariance of the resulting implicit safe set and show how the generalized construction recovers existing bCBF methods as special cases. Further, we extend the proposed framework to parameterized controller families, enabling online adaptation of the expansion controller while maintaining safety guarantees in the presence of input bounds.

I. INTRODUCTION

Ensuring the safety of modern autonomous systems is crucial to their real-world deployment. *Control barrier functions* (CBFs) have emerged as a popular approach to guaranteeing the safety of such systems by enforcing the forward invariance of safe sets [1]. While CBFs have been successfully deployed across many applications, from robotics to aerospace systems [2]–[4], many important challenges remain. Importantly, one must verify that for all states in the safe set, there exists a *feasible* safe controller, which is especially difficult for complex systems with input bounds.

Recent works have proposed reachability and prediction-based methods for constructing safety certificates under input constraints. Specifically, [5] introduces robust control barrier-value functions to unify Hamilton–Jacobi reachability and CBFs, yielding safe sets that recover the viability kernel while remaining useful for CBF-based filtering. Similarly, [6] uses dynamic-programming-based reachability to produce progressively less conservative barrier certificates. Along a related predictive direction, [7] shows that a time-invariant CBF can be constructed from finite-horizon predictions.

Perhaps one of the most promising approaches to address input constraints is *backup control barrier functions* (bCBFs) [8], [9]. Using the idea of set expansion, bCBFs create controlled invariant safe sets by examining the evolution of the system under a pre-verified *backup controller*, and ensuring

the finite-time backward reachability of a *backup set*. This construction yields forward invariance conditions which can be enforced via a quadratic program (QP). This approach scales to high-dimensional systems and has showcased strong performance across complex robotic platforms [9]–[12].

A key advantage of the bCBF method is that expanding the backup set with a pre-verified backup controller yields a tractable optimization problem and desirable properties on the expanded set. However, by requiring that the expansion be done with the backup controller, the possible set of controllers is restricted. This may yield a conservative expanded set, potentially resulting in reduced performance. In this work, we propose a method to separate the expanding controller and the backup controller, providing a more general technique for set expansion. A related approach is proposed in [13], where a neural backup policy learns to construct an implicit safe set from a backup set. This approach specializes to a learned backup policy and does not account for online adaptive expansion. Our contributions are as follows.

- We generalize the backup set method by decoupling the set-expanding controller from the backup controller that certifies the invariance of the backup set, and show that the resulting implicit safe set is controlled invariant.
- We derive forward invariance conditions for the generalized implicit set and formulate the corresponding QP that preserves safety guarantees with bounded inputs.
- We extend the framework to parameterized controller families, enabling online adaptation of the expansion controller while preserving invariance guarantees.

II. PRELIMINARIES

A. Control Barrier Functions

Consider the control affine dynamical system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n, \quad \mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m, \quad (1)$$

where $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^n$ and $\mathbf{g} : \mathcal{X} \rightarrow \mathbb{R}^{n \times m}$ are continuously differentiable, and the input constraint set $\mathcal{U}$ is a convex polytope. We are interested in designing a controller $\mathbf{k} : \mathcal{X} \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$, that guarantees the safety of system (1). We frame safety through forward invariance¹ and state membership to a set $\mathcal{C}_S \subseteq \mathcal{X}$ defined by a continuously differentiable safety function $h : \mathcal{X} \rightarrow \mathbb{R}$ as $\mathcal{C}_S \triangleq \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) \geq 0\}$. *Control barrier functions* (CBFs) provide forward invariance conditions and enable the synthesis of safe controllers. We call h a CBF² for (1) if $\sup_{\mathbf{u} \in \mathcal{U}} \dot{h}(\mathbf{x}, \mathbf{u}) > -\alpha(h(\mathbf{x}))$ holds for all $\mathbf{x} \in \mathcal{C}_S$ with an extended class- $\mathcal{K}$ function α .

¹A set $\mathcal{C}$ is *forward invariant* along the closed-loop system associated with (1) if $\mathbf{x}_0 \triangleq \mathbf{x}(0) \in \mathcal{C} \implies \mathbf{x}(t) \in \mathcal{C}, \forall t \geq 0$.

²The strict inequality of the CBF facilitates Lipschitz-continuous controllers and ensures that zero is a regular value of h .

*This work was supported in part by DARPA under the LINC program and by the Technology Innovation Institute (TII).

¹Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125, USA, {vanwijk, dohyun, ames, jburdick}@caltech.edu.

²Mechanical, Materials, and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL 60616, USA, edas2@illinoistech.edu.

³Mechanical Engineering, Wichita State University, Wichita, KS 67260, USA, tamas.molnar@wichita.edu.

Theorem 1 ([1]). *If h is a CBF for (1) on $\mathcal{C}_S$, then any locally Lipschitz controller $\mathbf{k} : \mathcal{X} \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$ satisfying*

$$\dot{h}(\mathbf{x}, \mathbf{u}) \geq -\alpha(h(\mathbf{x})), \quad (2)$$

for all $\mathbf{x} \in \mathcal{C}_S$ renders the set $\mathcal{C}_S$ forward invariant.

Given a primary controller $\mathbf{k}_p : \mathcal{X} \rightarrow \mathcal{U}$, safety can be enforced via a quadratic program (QP) subject to (2). CBFs are a powerful tool for generating safe controllers, but verifying that h is a CBF is notoriously challenging, especially for nonlinear systems with tight input bounds in $\mathcal{U}$ [8].

B. Backup Control Barrier Functions

Although finding a CBF is challenging with input bounds, it is often possible by linearizing (1) about a stabilizable equilibrium and considering a sublevel set of a Lyapunov function³. Yet, these techniques often result in a very conservative CBF (with a very small safe set), which can significantly inhibit the performance of CBF-based controllers. However, as shown in [9], such conservative sets can be *expanded* through a backup controller, resulting in a less conservative definition of safety. Therefore, bCBFs are more of a tool for *set expansion* rather than *CBF construction*. We now review this technique, and formalize the notion of set “expansion”. For this, we require the following definition⁴.

Definition 1 (Backup Set and Controller). A continuously differentiable controller $\mathbf{k}_b : \mathcal{X} \rightarrow \mathcal{U}$ is called a *backup controller* if it renders a set $\mathcal{C}_B \triangleq \{\mathbf{x} \in \mathcal{X} : h_b(\mathbf{x}) \geq 0\} \subseteq \mathcal{C}_S$, called a *backup set*, forward invariant for (1). The continuously differentiable function $h_b : \mathcal{X} \rightarrow \mathbb{R}$ is called the *backup function*.

We will henceforth not assume that h is a CBF, but h merely encodes a safety constraint. Because h_b is a CBF if $\mathcal{C}_B$ is compact, this could be used directly as in Section II-A to enforce safety, given that $\mathcal{C}_B \subseteq \mathcal{C}_S$. However, as previously mentioned, techniques to construct CBFs often result in conservative sets, yielding overly restrictive safe controllers. The backup set method expands $\mathcal{C}_B$ by considering the evolution of the system under the backup controller. Consider the *flow* of (1) under $\mathbf{k}_b$, written as $\phi_b(\tau, \mathbf{x})$, over $\tau \in [0, T]$ with a constant time horizon $T > 0$. We can obtain this flow by forward integrating the following differential equation:

$$\frac{\partial}{\partial \tau} \phi_b(\tau, \mathbf{x}) = \mathbf{f}_b(\phi_b(\tau, \mathbf{x})), \quad \phi_b(0, \mathbf{x}) = \mathbf{x}, \quad (3)$$

with the short-hand notation $\mathbf{f}_b(\mathbf{x}) \triangleq \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}_b(\mathbf{x})$.

Now consider the expanded set defined using this flow:

$$\mathcal{C}_I \triangleq \left\{ \mathbf{x} \in \mathcal{X} \mid \begin{array}{l} h(\phi_b(\tau, \mathbf{x})) \geq 0, \forall \tau \in [0, T], \\ h_b(\phi_b(T, \mathbf{x})) \geq 0 \end{array} \right\} \subseteq \mathcal{C}_S. \quad (4)$$

The set $\mathcal{C}_I$ contains the states from which the system (1) can *safely* reach the backup set, $\mathcal{C}_B$, in a finite time T , when using the backup controller $\mathbf{k}_b$. This set is implicit as it is defined in terms of the flow $\phi_b(\tau, \mathbf{x})$, which can seldom be computed analytically. This enables online safety enforcement without expensive offline computations because $\mathcal{C}_I$ is not needed in an explicit form. Instead, only finite-time trajectories are needed for safety enforcement, which has proven a useful strategy in

many challenging scenarios [10], [15], and has been shown to scale well with increasing state dimension [9]. We emphasize that $\mathcal{C}_I$ encodes that the flow is contained within $\mathcal{C}_S$ over the entirety of the time horizon, $\tau \in [0, T]$. This ensures that $\mathcal{C}_I \subseteq \mathcal{C}_S$, and thus membership of $\mathbf{x}$ to $\mathcal{C}_I$ implies membership of $\mathbf{x}$ to $\mathcal{C}_S$. Further, we note that because $\mathcal{C}_I \equiv \mathcal{C}_B$ for $T = 0$, and $\mathcal{C}_I$ grows for $T > 0$, $\mathcal{C}_I$ is indeed an *expansion* of $\mathcal{C}_B$. We now review the main benefit of expanding the backup set in this manner: $\mathcal{C}_I$ is controlled invariant⁵ by construction.

Lemma 1 ([9] [16, Lem. 1]). *The set $\mathcal{C}_I$ in (4) is controlled invariant, and the backup controller $\mathbf{k}_b : \mathcal{X} \rightarrow \mathcal{U}$ renders $\mathcal{C}_I$ forward invariant along (1).*

Theorem 2 ([9]). *There exist class- $\mathcal{K}_\infty$ functions α , α_b and a locally Lipschitz controller $\mathbf{k} : \mathcal{X} \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$ satisfying*

$$\dot{h}(\phi_b(\tau, \mathbf{x}), \mathbf{u}) \geq -\alpha(h(\phi_b(\tau, \mathbf{x}))), \quad \forall \tau \in [0, T], \quad (5a)$$

$$\dot{h}_b(\phi_b(T, \mathbf{x}), \mathbf{u}) \geq -\alpha_b(h_b(\phi_b(T, \mathbf{x}))), \quad (5b)$$

$\forall \mathbf{x} \in \mathcal{C}_I$. This controller renders $\mathcal{C}_I \subseteq \mathcal{C}_S$ forward invariant.

As in Section II-A, it is possible to solve a QP to obtain the safe control signal using the less conservative set, $\mathcal{C}_I$:

$$\begin{aligned} \mathbf{k}_{\text{safe}}(\mathbf{x}) = \underset{\mathbf{u} \in \mathcal{U}}{\text{argmin}} \quad & \|\mathbf{u} - \mathbf{k}_p(\mathbf{x})\|^2 & (\text{bCBF-QP}) \\ \text{s.t.} \quad & (5a), (5b). \end{aligned}$$

As written, (5a) represents an infinite number of constraints as τ varies continuously over the interval $[0, T]$. Therefore, in practice, this horizon is discretized and the flow is evaluated at a finite number of points along the horizon such that $\tau \in \{0, \Delta, \dots, T\}$ for a discretization step $\Delta > 0$ with $T/\Delta \in \mathbb{N}$. We assume that for sufficiently fine discretization, satisfaction of the discretized version of (5a) implies satisfaction of its continuous counterpart.

Though bCBFs have been used successfully across a wide range of applications, the size of the expanded, implicit safe set $\mathcal{C}_I$ is inherently tied to the choice of the backup controller. In this work, we challenge this paradigm, and offer a generalization of the standard backup set method.

III. GENERALIZED BACKUP SET METHOD

Our main idea is that the controller which expands the backup set $\mathcal{C}_B$ can be *different* from the one which renders $\mathcal{C}_B$ invariant. Specifically, the backup controller does not have to be the one that expands $\mathcal{C}_B$ to obtain an implicit safe set.

Assume that there exists a continuously differentiable controller $\mathbf{k}_e : \mathcal{X} \rightarrow \mathcal{U}$ which represents the *set-expanding* controller. Though $\mathbf{k}_e$ may be a good set expander⁶, it does not necessarily render $\mathcal{C}_B$ forward invariant. Because of this, one cannot directly use this controller to construct an implicit safe set as done in Section II-B for $\mathcal{C}_I$ (c.f. (4)). Instead, we can still leverage the properties of the backup controller from Def. 1 by considering a switching-based controller,

$$\mathbf{k}_s(\mathbf{x}) = (1 - \eta(\mathbf{x}))\mathbf{k}_e(\mathbf{x}) + \eta(\mathbf{x})\mathbf{k}_b(\mathbf{x}), \quad (6)$$

with a switching function $\eta : \mathcal{X} \rightarrow [0, 1]$. Notice that by construction, $\mathbf{k}_s(\mathbf{x}) \in \mathcal{U}$ for all $\mathbf{x} \in \mathcal{X}$ if $\mathbf{k}_b(\mathbf{x}) \in \mathcal{U}$, $\mathbf{k}_e(\mathbf{x}) \in \mathcal{U}$,

³This approach is successfully used to generate a backup controller and a backup set for a nonlinear quadrotor in Case Study 2.

⁴See [14] for methods to obtain backup controllers and set pairs.

⁵A set $\mathcal{C}$ is *controlled invariant* if there exists a controller $\mathbf{k} : \mathcal{X} \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$ rendering $\mathcal{C}$ forward invariant for (1).

⁶The notion of a “good” set expander will be further elaborated upon.

and the input constraint set $\mathcal{U}$ is convex. We now restrict the choice of η such that $\mathbf{k}_s$ has desirable properties.

Assumption 1. *The switching function $\eta : \mathcal{X} \rightarrow [0, 1]$ is continuously differentiable, and satisfies $\eta(\mathbf{x}) = 1, \forall \mathbf{x} \in \mathcal{C}_B$.*

An example of many possible switching functions satisfying Assumption 1 is $\eta(\mathbf{x}) = \Gamma(h_b(\mathbf{x}))$ where $\Gamma : \mathbb{R} \rightarrow [0, 1]$ is a continuously differentiable step function given by

$$\Gamma(z) \triangleq \begin{cases} 0 & \text{if } z \leq -\varepsilon, \\ 3(\frac{z+\varepsilon}{\varepsilon})^2 - 2(\frac{z+\varepsilon}{\varepsilon})^3 & \text{if } -\varepsilon < z < 0, \\ 1 & \text{if } z \geq 0, \end{cases} \quad (7)$$

and $\varepsilon > 0$ governs the switching region.

Lemma 2. *The switched controller $\mathbf{k}_s$ in (6) renders the backup set $\mathcal{C}_B$ satisfying Definition 1 forward invariant along (1) for any switching function η satisfying Assumption 1.*

Proof. By Assumption 1, for all $\mathbf{x} \in \mathcal{C}_B$ we have that $\mathbf{k}_s(\mathbf{x}) = \mathbf{k}_b(\mathbf{x})$ as per (6), and by Definition 1, the backup controller $\mathbf{k}_b$ renders $\mathcal{C}_B$ forward invariant. ■

Remark 1. *For any η satisfying Assumption 1, the switched controller $\mathbf{k}_s$ itself is a valid backup controller for $\mathcal{C}_B$.*

Using (6), the switched dynamical system is

$$\dot{\mathbf{x}} = \mathbf{f}_s(\mathbf{x}) \triangleq \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}_s(\mathbf{x}). \quad (8)$$

With the above definitions, the expanded safe set is

$$\mathcal{C}_1^* \triangleq \left\{ \mathbf{x} \in \mathcal{X} \mid \begin{array}{l} h(\phi_s(\tau, \mathbf{x})) \geq 0, \forall \tau \in [0, T], \\ h_b(\phi_s(T, \mathbf{x})) \geq 0 \end{array} \right\}. \quad (9)$$

Here, the switched flow $\phi_s(\tau, \mathbf{x})$ is the evolution of (8) over $\tau \in [0, T]$ for a fixed, finite horizon $T > 0$ starting at state $\mathbf{x}$:

$$\frac{\partial}{\partial \tau} \phi_s(\tau, \mathbf{x}) = \mathbf{f}_s(\phi_s(\tau, \mathbf{x})), \quad \phi_s(0, \mathbf{x}) = \mathbf{x}. \quad (10)$$

Remark 2. *A possible switching function that would render $\mathcal{C}_B$ forward invariant could be $\eta(\mathbf{x}) = 1$ if $h_b(\mathbf{x}) \geq 0$ and $\eta(\mathbf{x}) = 0$ otherwise. However, the discontinuity of this switch would make the controller (6) and the switched system (8) non-smooth. Restricting η to be continuously differentiable admits global solutions $\phi_s(\tau, \mathbf{x})$ and well-behaved gradients.*

We now state a powerful result for the expanded set $\mathcal{C}_1^*$.

Lemma 3. *The set $\mathcal{C}_1^*$ in (9) is controlled invariant, and the switched controller $\mathbf{k}_s$ renders $\mathcal{C}_1^*$ forward invariant along (8), such that $\mathbf{x} \in \mathcal{C}_1^* \implies \phi_s(\tau, \mathbf{x}) \in \mathcal{C}_1^* \subseteq \mathcal{C}_S, \forall \tau \geq 0$.*

Proof. From the definition of $\mathcal{C}_1^*$, we have $\mathbf{x} \in \mathcal{C}_1^* \implies \phi_s(T, \mathbf{x}) \in \mathcal{C}_B$, and with Lemma 2,

$$\mathbf{x} \in \mathcal{C}_1^* \implies \phi_s(\tau, \mathbf{x}) \in \mathcal{C}_B \subseteq \mathcal{C}_S, \forall \tau \geq T. \quad (11)$$

The flow satisfies the group property: for any $\mathbf{x} \in \mathbb{R}^n$ and $\tau, \vartheta \geq 0$, $\phi_s(\tau + \vartheta, \mathbf{x}) = \phi_s(\tau, \phi_s(\vartheta, \mathbf{x}))$. Using this and (11),

$$\mathbf{x} \in \mathcal{C}_1^* \implies \phi_s(T, \phi_s(\vartheta, \mathbf{x})) \in \mathcal{C}_B, \forall \vartheta \geq 0. \quad (12)$$

From (11) and (9), $\mathbf{x} \in \mathcal{C}_1^* \implies \phi_s(\tau, \mathbf{x}) \in \mathcal{C}_S, \forall \tau \geq 0$. Using the group property of the flow once more

$$\mathbf{x} \in \mathcal{C}_1^* \implies \phi_s(\tau, \phi_s(\vartheta, \mathbf{x})) \in \mathcal{C}_S, \forall \tau \in [0, T], \forall \vartheta \geq 0. \quad (13)$$

Definition (9) with (12) and (13) completes the proof. ■

Now that $\mathcal{C}_1^*$ has been shown to be controlled invariant, we derive its forward invariance conditions, which may be used to solve for a safe control signal. We require that for some class- $\mathcal{K}_\infty$ functions α, α_b

$$\begin{aligned} \dot{h}(\phi_s(\tau, \mathbf{x}), \mathbf{u}) &\geq -\alpha(h(\phi_s(\tau, \mathbf{x}))), \quad \forall \tau \in [0, T], \\ \dot{h}_b(\phi_s(T, \mathbf{x}), \mathbf{u}) &\geq -\alpha_b(h_b(\phi_s(T, \mathbf{x}))) \end{aligned} \quad (14)$$

holds. From the chain rule we have

$$\dot{h}(\phi_s(\tau, \mathbf{x}), \mathbf{u}) = \nabla h(\phi_s(\tau, \mathbf{x}))\Phi(\tau, \mathbf{x})\dot{\mathbf{x}}, \quad (15a)$$

$$\dot{h}_b(\phi_s(T, \mathbf{x}), \mathbf{u}) = \nabla h_b(\phi_s(T, \mathbf{x}))\Phi(T, \mathbf{x})\dot{\mathbf{x}}, \quad (15b)$$

with $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$. Here, $\Phi(\tau, \mathbf{x}) \triangleq \frac{\partial \phi_s(\tau, \mathbf{x})}{\partial \mathbf{x}}$ is the sensitivity of the switched flow to $\mathbf{x}$, the initial condition for (10). The sensitivity matrix is the solution to

$$\frac{\partial}{\partial \tau} \Phi(\tau, \mathbf{x}) = \mathbf{F}_s(\phi_s(\tau, \mathbf{x}))\Phi(\tau, \mathbf{x}), \quad \Phi(0, \mathbf{x}) = \mathbf{I}_n, \quad (16)$$

where $\mathbf{F}_s(\mathbf{x}) \triangleq \frac{\partial \mathbf{f}_s}{\partial \mathbf{x}}(\mathbf{x})$ is the closed-loop Jacobian of (8).

We now present the main result.

Theorem 3. *Any locally Lipschitz controller $\mathbf{k} : \mathcal{X} \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$ satisfying*

$$\nabla h(\phi_s(\tau, \mathbf{x}))\Phi(\tau, \mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) \geq -\alpha(h(\phi_s(\tau, \mathbf{x}))), \quad \forall \tau \in [0, T], \quad (17a)$$

$$\nabla h_b(\phi_s(T, \mathbf{x}))\Phi(T, \mathbf{x})(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) \geq -\alpha_b(h_b(\phi_s(T, \mathbf{x}))), \quad (17b)$$

for all $\mathbf{x} \in \mathcal{C}_1^*$, renders $\mathcal{C}_1^* \subseteq \mathcal{C}_S$ forward invariant for (1).

Proof. By applying Theorem 1 to system (1), we have that the conditions in (17) imply that for any $\mathbf{x}(0) \in \mathcal{C}_1^*$, $\mathbf{x}(t) \in \mathcal{C}_1^*$ for all $t \geq 0$. By definition, $\mathcal{C}_1^* \subseteq \mathcal{C}_S$. ■

We use Theorem 3 to obtain a point-wise optimal safe controller via the proposed *Generalized Backup CBF* method:

$$\begin{aligned} \mathbf{k}_{\text{safe}}(\mathbf{x}) &= \underset{\mathbf{u} \in \mathcal{U}}{\operatorname{argmin}} \quad \|\mathbf{u} - \mathbf{k}_p(\mathbf{x})\|^2 \\ &\text{s.t. (17a), (17b).} \end{aligned} \quad (\text{GB-QP})$$

Further, by Lemma 3, there exists at least one controller (namely $\mathbf{k}_s$) which renders $\mathcal{C}_1^*$ forward invariant, and satisfies the input constraints imposed by $\mathcal{U}$.

Remark 3. *If $\mathbf{k}_e(\mathbf{x}) = \mathbf{k}_b(\mathbf{x})$, then $\mathcal{C}_1^* \equiv \mathcal{C}_1$ and the (GB-QP) recovers the (bCBF-QP) as a special case.*

Having established the theoretical basis for generalizing the backup set method, we direct our attention to an example to answer the question: “Why should the set-expanding controller be different from the backup controller?”

Case Study 1 (Double Integrator). Consider the system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (18)$$

with state $\mathbf{x} = [x_1 \ x_2]^\top \in \mathbb{R}^2$, input $\mathbf{u} \in \mathcal{U} = [-1, 1]$, and constraint set $\mathcal{C}_S \triangleq \{\mathbf{x} \in \mathbb{R}^2 : h(\mathbf{x}) = x_{\max}^2 - x_1^2 \geq 0\}$. Let the backup set be $\mathcal{C}_B \triangleq \{\mathbf{x} \in \mathbb{R}^2 : h_b(\mathbf{x}) = \rho - \mathbf{x}^\top \mathbf{P} \mathbf{x} \geq 0\}$, for a positive definite matrix $\mathbf{P}$ and a $\rho > 0$ such that $\mathcal{C}_B \subseteq \mathcal{C}_S$. Consider the backup controller with gain $\mathbf{K}$ as

$$\mathbf{k}_b(\mathbf{x}) = \operatorname{sat}(-\mathbf{K}\mathbf{x}), \quad (19)$$

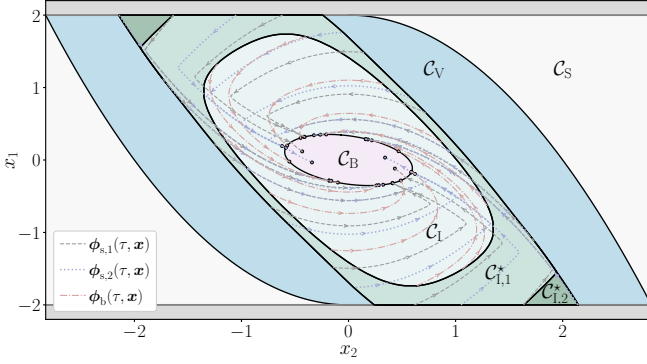


Fig. 1. Controlled invariant sets for (18) with three different expansion strategies. C_I is constructed using the closed-loop flow for the backup controller k_b , whilst $C_{I,1}^*$ and $C_{I,2}^*$ use the switching controller k_s in (6), which allows for a more flexible set expansion policy. The flows denoted by ϕ_b , $\phi_{s,1}$, and $\phi_{s,2}$ represent the evolution of (18) under each of the controllers. The viability kernel, denoted by C_V , is plotted for comparison.

where sat denotes any smooth saturation function. To ensure that k_b does not saturate inside C_B , we require a condition⁷ on K so that $k_b(x) = -Kx$ for all $x \in C_B$. Then, this controller renders C_B forward invariant if K satisfies $A_{cl}^\top P + PA_{cl} = -Q$ where $A_{cl} \triangleq A - BK$ with a positive definite matrix Q . Thus, to render C_B forward invariant while satisfying input limits for k_b , the choice of the gain K is restricted. This restriction may negatively impact the size of the set C_I when the same controller k_b is used to expand C_B .

Now, consider the expanding controller

$$k_{e,1}(x) = \text{sat}(-K_e x), \quad (20)$$

with a gain K_e . Because we do not have to verify that this controller renders C_B forward invariant, and because (20) is saturated to satisfy input constraints, the gain K_e can be made arbitrarily large. Choosing a larger gain may facilitate a larger expansion C_I^* of the backup set C_B . Alternatively, to further improve expansion, we may also consider a smooth approximation of the time-optimal controller described in [18, Ch. 3.9] which reaches the origin in minimum time:

$$k_{e,2}(x) = -\text{sat}((x_2^2 \text{sat}(x_2/\beta) + 2x_1)/\beta), \quad (21)$$

where $\beta > 0$ is a smoothing parameter. Figure 1 compares⁸ the size of C_I defined in (4) with $C_{I,1}^*$ and $C_{I,2}^*$ defined in (9) using k_s in (6) with the expanding controllers $k_{e,1}$ and $k_{e,2}$ respectively, and the switching function in (7). Clearly, $C_I \subseteq C_{I,1}^*, C_{I,2}^*$, illustrating that the generalized backup CBF method yields significantly larger safe sets in this example.

A. Expansion Controller Design

By decoupling the backup and expanding controllers, one may construct a larger controlled invariant set (see Figure 1), leading to less conservative safe controllers. Further, by establishing the necessary machinery for this decoupling, we allow for a very broad class of expanding controllers. Note, however, that not all controllers may be *useful* expanders⁹.

As evidenced in Case Study 1, a useful expander is often of the same form as k_b , but with gains that can be made

⁷By [17, Eq. (B4)] we require $\max_{x^\top P x \leq \rho} \|Kx\| = \sqrt{\rho} \|KP^{-1/2}\| \leq 1$.

⁸The simulation uses the constants $K = [2, 1.6]$, $K_e = 30K$, $\rho = 0.15$, $Q = I_2$, $\beta = 1 \times 10^{-9}$, $\varepsilon = 1 \times 10^{-3}$, $T = 2$, $\Delta = 0.025$.

⁹For an ill-informed choice of k_e , we have $C_I^* \equiv C_B$ in the worst case.

much more aggressive. This is because the expander need not satisfy the strict requirements of Def. 1, but still exhibits convergence to C_B . However, we are not restricted to only expanders of the same form as k_b : we may even consider learned controllers [13], or *optimal* expanding controllers.

Remark 4 (Optimal Expanding Controller). We recall the optimal control problem¹⁰ from [8]:

$$k_e^*(x) \triangleq \arg \max_{u \in \mathcal{U}_c} h_b(\phi(T, x)) \quad (22)$$

$$\begin{aligned} \text{s.t. } & \frac{\partial}{\partial \tau} \phi(\tau, x) = f(\phi(\tau, x)) + g(\phi(\tau, x))u(\tau), \forall \tau \in [0, T], \\ & u(\tau) \in \mathcal{U}, \forall \tau \in [0, T], \\ & \phi(0, x) = x. \end{aligned}$$

The controller k_e^* that solves (22) is an optimal expanding control policy ensuring that the terminal state, $\phi(T, x)$, lies as far as possible inside the backup set. Though solving (22) is challenging for nonlinear systems and arbitrary h_b , minimum-time-to-origin optimal control problems may often be useful surrogates to (22) (see e.g., (21)).

Next, we discuss how the cost function in (22) may inform the adaptation of parameterized expansion controllers.

IV. ADAPTIVE SET EXPANSION

In this section, we consider yet another layer of generalization: we explore expanders that are not fixed, but change over time through adaptation. Suppose now that the controller k_e is parameterized¹¹ by $\theta \in \mathbb{R}^p$ and written as $k_e(x, \theta)$. Considering the parameters θ as auxiliary states, the full augmented state is written as $\hat{x} = [x^\top \theta^\top]^\top \in \hat{\mathcal{X}} \triangleq \mathcal{X} \times \mathbb{R}^p$, and thus the augmented dynamics for (1) can be written as

$$\dot{\hat{x}} = \underbrace{\begin{bmatrix} f(x) \\ 0_{p \times 1} \end{bmatrix}}_{\triangleq \hat{f}(\hat{x})} + \underbrace{\begin{bmatrix} g(x) & 0_{n \times p} \\ 0_{p \times m} & I_p \end{bmatrix}}_{\triangleq \hat{g}(\hat{x})} \hat{u}, \quad (23)$$

where $\hat{u} \triangleq [u^\top u_\theta^\top]^\top \in \hat{\mathcal{U}} = \mathcal{U} \times \mathbb{R}^p$ and $u_\theta \in \mathbb{R}^p$ is an auxiliary input which modifies the parameter θ . We now define the *augmented* backup controller and expanding controller:

$$\hat{k}_b(\hat{x}) \triangleq [k_b(x)^\top \ 0_{1 \times p}]^\top, \quad \hat{k}_e(\hat{x}) \triangleq [k_e(x, \theta)^\top \ 0_{1 \times p}]^\top.$$

Notice that each controller keeps θ fixed. Analogously to Section III, we write the switched augmented dynamics as

$$\dot{\hat{x}} = \hat{f}_s(\hat{x}) \triangleq \hat{f}(\hat{x}) + \hat{g}(\hat{x})\hat{k}_s(\hat{x}), \quad (24)$$

for the switched controller

$$\hat{k}_s(\hat{x}) = [(1 - \eta(x))k_e(x, \theta)^\top + \eta(x)k_b(x)^\top \ 0_{1 \times p}]^\top, \quad (25)$$

with $\eta : \mathcal{X} \rightarrow [0, 1]$. Then, the augmented switched flow is

$$\frac{\partial}{\partial \tau} \hat{\phi}_s(\tau, \hat{x}) = \hat{f}_s(\hat{\phi}_s(\tau, \hat{x})), \quad \hat{\phi}_s(0, \hat{x}) = \hat{x}, \quad (26)$$

over the interval $\tau \in [0, T]$ for a horizon $T > 0$. We redefine the safe and backup sets in terms of the augmented state $\hat{x}$:

$$\hat{C}_s \triangleq \{\hat{x} \in \hat{\mathcal{X}} : \hat{h}(\hat{x}) \geq 0\}, \quad \hat{C}_b \triangleq \{\hat{x} \in \hat{\mathcal{X}} : \hat{h}_b(\hat{x}) \geq 0\},$$

where $\hat{h}(\hat{x}) \triangleq h(x)$ and $\hat{h}_b(\hat{x}) \triangleq h_b(x)$. We now present a result analogous to Lemma 2 for the augmented system.

¹⁰ $\mathcal{U}_c$ is the set of piecewise-continuous control signals.

¹¹For example, one could write k_e in (20) as $k_e(x, \theta) = \text{sat}(-\theta^\top x)$.

Lemma 4. The controller $\hat{k}_s$ in (25) renders $\hat{C}_B$ forward invariant along (23) for any η satisfying Assumption 1.

Proof. Observe that in (23), the dynamics of x are decoupled from those of θ , and that $\hat{x} \in \hat{C}_B$ if and only if $x \in C_B$, for any $\theta \in \mathbb{R}^p$. Therefore, $\hat{k}_s$ rendering $\hat{C}_B$ invariant along (23) is equivalent to k_s rendering C_B invariant along (1). Thus, by Lemma 2, $\hat{C}_B$ is invariant under $\hat{k}_s$. ■

We define the new implicit safe set $\hat{C}_I^* \subseteq \hat{C}_S$ based on $\hat{x}$:

$$\hat{C}_I^* \triangleq \left\{ \hat{x} \in \hat{\mathcal{X}} \mid \begin{aligned} &\hat{h}(\hat{\phi}_s(\tau, \hat{x})) \geq 0, \forall \tau \in [0, T], \\ &\hat{h}_b(\hat{\phi}_s(T, \hat{x})) \geq 0 \end{aligned} \right\}. \quad (27)$$

This set encodes all states $\hat{x}$ for which the x components of $\hat{\phi}_s(\tau, \hat{x})$ stay within C_S and reach C_B when $\tau = T$. We now show that this new implicit safe set is controlled invariant.

Lemma 5. The set $\hat{C}_I^*$ is controlled invariant and rendered forward invariant along (23) by $\hat{k}_s$.

Proof. The proof follows that of Lemma 3. Due to $\hat{x} \in \hat{C}_I^* \implies \hat{\phi}_s(T, \hat{x}) \in \hat{C}_B$, Lemma 4, and the flow group property,

$$\hat{x} \in \hat{C}_I^* \implies \hat{\phi}_s(T, \hat{\phi}_s(\vartheta, \hat{x})) \in \hat{C}_B, \forall \vartheta \geq 0. \quad (28)$$

Based on (27), the flow group property, and since $\hat{C}_B \subseteq \hat{C}_S$,

$$\hat{x} \in \hat{C}_I^* \implies \hat{\phi}_s(\tau, \hat{\phi}_s(\vartheta, \hat{x})) \in \hat{C}_S, \forall \tau \in [0, T], \forall \vartheta \geq 0. \quad (29)$$

Thus (28), (29) with definition (27) complete the proof. ■

Now that the controlled invariance of $\hat{C}_I^*$ has been established, we are in a position to derive the forward invariance conditions for this set. From definition (27), this requires

$$\nabla \hat{h}(\hat{\phi}_s(\tau, \hat{x})) \hat{\Phi}(\tau, \hat{x}) \dot{\hat{x}} \geq -\alpha(\hat{h}(\hat{\phi}_s(\tau, \hat{x}))), \quad (30a)$$

$$\nabla \hat{h}_b(\hat{\phi}_s(T, \hat{x})) \hat{\Phi}(T, \hat{x}) \dot{\hat{x}} \geq -\alpha_b(\hat{h}_b(\hat{\phi}_s(T, \hat{x}))), \quad (30b)$$

for all $\tau \in [0, T]$ and some class- $\mathcal{K}_\infty$ functions α, α_b . The term $\hat{\Phi}(\tau, \hat{x}) \triangleq \frac{\partial \hat{\phi}_s(\tau, \hat{x})}{\partial \hat{x}}$ is the sensitivity of the flow with respect to changes in $\hat{x}$, and is calculated similarly to (16).

Theorem 4. Any locally Lipschitz controller $\hat{k} : \hat{\mathcal{X}} \rightarrow \hat{\mathcal{U}}$, $\hat{u} = \hat{k}(\hat{x})$ satisfying

$$\begin{aligned} \nabla \hat{h}(\hat{\phi}_s(\tau, \hat{x})) \hat{\Phi}(\tau, \hat{x}) (\hat{f}(\hat{x}) + \hat{g}(\hat{x}) \hat{u}) &\geq \\ &-\alpha(\hat{h}(\hat{\phi}_s(\tau, \hat{x}))), \quad \forall \tau \in [0, T], \end{aligned} \quad (31a)$$

$$\begin{aligned} \nabla \hat{h}_b(\hat{\phi}_s(T, \hat{x})) \hat{\Phi}(T, \hat{x}) (\hat{f}(\hat{x}) + \hat{g}(\hat{x}) \hat{u}) &\geq \\ &-\alpha_b(\hat{h}_b(\hat{\phi}_s(T, \hat{x}))), \end{aligned} \quad (31b)$$

for all $\hat{x} \in \hat{C}_I^*$, renders $\hat{C}_I^*$ forward invariant for (23), and thus ensures that $\hat{x}(0) \in \hat{C}_I^* \implies x(t) \in C_S, \forall t \geq 0$.

Proof. By applying Theorem 1 to system (23), the conditions in (31) imply that $\hat{x}(t) \in \hat{C}_I^*$ for all $t \geq 0$ if $\hat{x}(0) \in \hat{C}_I^*$. Since $\hat{C}_I^* \subseteq \hat{C}_S$, and because $\hat{x} \in \hat{C}_S$ if and only if $x \in C_S$, for any $\theta \in \mathbb{R}^p$, we have that $x(t) \in C_S, \forall t \geq 0$. ■

Using Theorem 4, we design an *Adaptive Generalized Backup CBF* controller which ensures safety and is feasible:

$$\begin{aligned} \hat{k}_{\text{safe}}(\hat{x}) = \underset{\hat{u} \in \hat{\mathcal{U}}}{\operatorname{argmin}} \quad &\|\hat{u} - \hat{k}_p(\hat{x})\|_W^2 \\ \text{s.t.} \quad &(31a), (31b), \end{aligned} \quad (\text{aGB-QP})$$

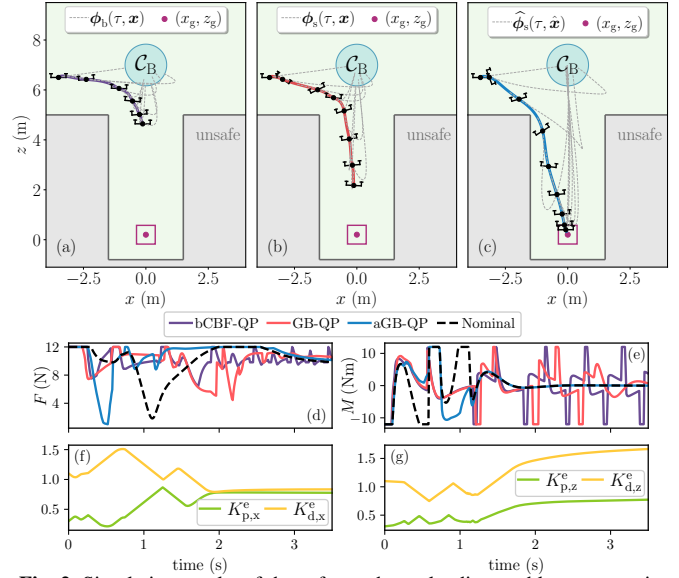


Fig. 2. Simulation results of the safe quadrotor landing problem, comparing the standard bCBF (purple), the generalized bCBF (red), and the adaptive bCBF (blue). All three approaches maintain safety (a,b,c) and obey the input constraints (d,e), yet differ in task performance. The bCBF is unable to reach the goal due to the terminal constraint, as the flows (dashed gray) must reach C_B (a). The (GB-QP) also does not reach the goal, but achieves a much larger expansion of C_B than the (bCBF-QP) (b). Using the adaptation law in Corollary 1, the (aGB-QP) reaches the goal via superior expansion of C_B (c) by modifying the expansion controller gains (f,g).

with matrix $W \succ 0$ weighting the inputs related to x and θ . The augmented primary controller is

$$\hat{k}_p(\hat{x}) \triangleq [\hat{k}_p(x)^\top \quad \hat{k}_\theta(x, \theta)^\top]^\top. \quad (32)$$

While $\hat{k}_p$ is prescribed, the desired auxiliary control given by $\hat{k}_\theta$ is in fact a design variable. Therefore, to encourage set expansion, we propose an approach to adapt θ that is based on the gradient of the cost function in the optimal control problem (22). We select the control direction $\hat{k}_\theta$ for the parameter θ which increases $\hat{h}_b(\hat{\phi}_s(T, \hat{x}))$ the most, pushing the terminal point of the flow deeper inside $\hat{C}_B$. This allows $\hat{x}$ to evolve farther from $\hat{C}_B$, improving the expansion.

Corollary 1 (Set Expansion Adaptation). *The controller*

$$\hat{k}_\theta(x, \theta) = \gamma \cdot \Pi(\nabla \hat{h}_b(\hat{\phi}_s(T, \hat{x})) \hat{\Phi}(T, \hat{x}) \hat{g}(\hat{x}))^\top, \quad (33)$$

adapts the parameters θ in the direction of steepest ascent of $\hat{h}_b(\hat{\phi}_s(T, \hat{x}))$, for $\Pi \triangleq [0_{p \times m} \quad I_p]$ and $\gamma > 0$, and thus locally increases the cost function of (22).

Since the adaptation law (33) does not necessarily guarantee the forward invariance of $\hat{C}_I^*$, the (aGB-QP) filters this signal and results in a controller that minimizes the deviation from $\hat{k}_\theta$, whilst preserving the forward invariance of $\hat{C}_I^*$.

V. NUMERICAL SIMULATIONS

Case Study 2 (Quadrotor). Consider a planar quadrotor

$$\ddot{x} = F \sin(\theta)/m, \quad \ddot{z} = F \cos(\theta)/m - g_D, \quad \ddot{\theta} = -M/J, \quad (34)$$

with state vector $x = [x \ z \ \theta \ \dot{x} \ \dot{z} \ \dot{\theta}]^\top \in \mathcal{X} \subset \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3$. The states x and z denote the horizontal and vertical positions, and θ is the pitch angle. The bounded inputs are the thrust $F \in [0, F_{\max}]$ and moment $M \in [-M_{\max}, M_{\max}]$. Here, g_D

is gravitational acceleration, m is the mass of the quadrotor, and J is the principal moment of inertia about the y -axis.

We consider a safe landing scenario in which the quadrotor must reach a goal location (x_g, z_g) whilst avoiding the walls and floor of a narrow landing zone (see Figure 2). The safety constraint defines the unsafe walls and floor as a polytopic region $C_S \triangleq \{x \in \mathcal{X} : h(x) = \max\{\min\{h_1, h_2, h_3\}, h_4\} \geq 0\}$, where $h_1(x) = x - x_L$, $h_2(x) = x_R - x$, $h_3(x) = z - z_L$, $h_4(x) = z - z_T$ with $x_R > x_L$ and $z_T > z_L$. The safe set is then smoothly approximated as in [19]. The nominal controller k_p is a feedback controller that attempts to reach the goal position but ignores the landing obstructions.

The backup controller is a proportional-derivative controller that drives the system (34) to the equilibrium point $x^* = [x_b \ z_b \ 0 \ 0 \ 0]^T \in C_S$ such that $k_b(x) = [F_b, M_b]^T$:

$$a_x = -K_{p,x}(x - x_b) - K_{d,x}\dot{x}, \quad (35)$$

$$a_z = -K_{p,z}(z - z_b) - K_{d,z}\dot{z} + g_D, \quad (36)$$

$$\theta_d = \text{sat}(\arctan(a_x/a_z)), \quad (37)$$

$$F_b = \text{sat}(m\sqrt{a_x^2 + a_z^2}), \quad (38)$$

$$M_b = \text{sat}(J(K_{p,\theta}(\theta - \theta_d) + K_{d,\theta}\dot{\theta})). \quad (39)$$

Here, $K_{p,x}, K_{d,x}, K_{p,z}, K_{d,z}, K_{p,\theta}, K_{d,\theta} > 0$ are gains. The backup set is a sublevel set of a Lyapunov function centered on x^* , obtained by linearizing (34) under k_b which yields

$$C_B = \{x \in \mathcal{X} : h_b(x) = \rho - (x - x^*)^T P(x - x^*) \geq 0\},$$

for a positive definite matrix P and $\rho > 0$. For C_B and k_b to be a valid backup set and controller pair (Def. 1), the gains and ρ must be carefully chosen to ensure that k_b does not saturate within C_B , that k_b renders C_B forward invariant, and that $C_B \subset C_S$. Satisfaction of these conditions (the details of which are omitted for space constraints¹²) results in a restrictive set of possible gains and level set sizes. Controller k_e is of the same form as (35)-(39), but with much more aggressive gains $K_{p,x}^e, K_{d,x}^e, K_{p,z}^e, K_{d,z}^e, K_{p,\theta}^e, K_{d,\theta}^e > 0$, since the expander is not limited by the requirements in Def. 1.

Figure 2 compares¹³ the standard backup set method (Section II-B) with the proposed generalized and adaptive approaches (Section III, Section IV). While all three approaches guarantee the safety of (34) in the presence of strict input bounds, the three controllers achieve differing levels of set expansion. The (bCBF-QP) is unable to substantially expand C_B due to the restrictions imposed on the gains. By separating the expanding controller and the backup controller, the (GB-QP) is able to move much farther away from C_B , and thus gets much closer to the goal. The (aGB-QP) uses initial parameters $\theta(0) = [K_{p,x}^e \ K_{d,x}^e \ K_{p,z}^e \ K_{d,z}^e \ K_{p,\theta}^e \ K_{d,\theta}^e]^T$ and adapts them according to Corollary 1. This method enables the quadrotor to safely reach the goal via even better expansion of C_B . Note in panels (f) and (g) that when the quadrotor achieves a horizontal position near x_b , the gains governing x remain constant, while the gains governing z increase to

¹²Note that sufficient conditions on the gains and ρ can be obtained by using [20, Sec. 3.7.3] in conjunction with [21, Thm. 4.7].

¹³The simulation uses $g_D = 9.81 \text{ m/s}^2$, $J = 0.25 \text{ kgm}^2$, $m = 1 \text{ kg}$, $F_{\max} = 12 \text{ N}$, $M_{\max} = 12 \text{ Nm}$, $\rho = 0.22$, $\epsilon = 0.01$, $T = 6.75 \text{ s}$, $\gamma = 10$, $\alpha(r) = 2r + r^3$, $\alpha_b(r) = 10r + r^3$, $W = \text{diag}(300, 300, \mathbf{1}_{6 \times 1})$, $\Delta = 0.02 \text{ s}$, $K_{p,x} = K_{p,z} = 0.16$, $K_{d,x} = K_{d,z} = 0.8$, $K_{p,\theta} = 64$, $K_{d,\theta} = 16$, $K_{p,x}^e = K_{p,z}^e = 0.303$, $K_{d,x}^e = K_{d,z}^e = 1.1$, $K_{p,\theta}^e = 64$, $K_{d,\theta}^e = 16$.

allow the quadrotor to move farther down towards the goal.

VI. CONCLUSION

We presented a novel approach for guaranteeing the safety of input-bounded nonlinear systems by enhancing the backup set method. We proved that separating the expanding controller from the backup controller preserves the controlled invariance of the expanded set, resulting in substantially larger safe sets. Further, our framework allows for online adaptation of the expanding controller, which was demonstrated to be highly beneficial in a complex quadrotor landing problem.

REFERENCES

- [1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [2] K. Long, Y. Yi, Z. Dai, S. Herbert, J. Cortés, and N. Atanasov, "Sensor-based distributionally robust control for safe robot navigation in dynamic environments," *Int. J. Robot. Res.*, vol. 45, no. 2, pp. 328–351, 2026.
- [3] A. K. Dickson, J. C. P. Garcia, M. L. Anderson, R. Jing, S. Alizadeh-Shabdz, A. X. Wang, C. DeLorey, Z. J. Patterson, and A. P. Sabelhaus, "Safe autonomous environmental contact for soft robots using control barrier functions," *IEEE Robot. Autom. Lett.*, vol. 10, no. 11, pp. 11283–11290, 2025.
- [4] D. R. Agrawal and D. Panagou, "Online safety under multiple constraints and input bounds using gatekeeper: Theory and applications," *IEEE Control Syst. Lett.*, vol. 9, pp. 2309–2314, 2025.
- [5] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier-value functions for safety-critical control," in *60th IEEE Conf. Decis. Control*, pp. 6814–6821, 2021.
- [6] S. Tonkens and S. Herbert, "Refining control barrier functions through Hamilton-Jacobi reachability," in *IEEE Int. Conf. Intell. Robots Syst.*, pp. 13355–13362, 2022.
- [7] A. Wiltz, X. Tan, and D. V. Dimarogonas, "Construction of control barrier functions using predictions with finite horizon," in *62nd IEEE Conf. Decis. Control*, pp. 2743–2749, 2023.
- [8] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An Online Approach to Active Set Invariance," in *57th IEEE Conf. Decis. Control*, pp. 3592–3599, 2018.
- [9] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187249–187275, 2020.
- [10] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2897–2904, 2022.
- [11] D. Ko and W. K. Chung, "A backup control barrier function approach for safety-critical control of mechanical systems under multiple constraints," *IEEE/ASME Trans. Mechatron.*, vol. 30, no. 6, pp. 4460–4471, 2025.
- [12] N. C. Janwani, E. Daş, T. Touma, S. X. Wei, T. G. Molnar, and J. W. Burdick, "A learning-based framework for safe human-robot collaboration with multiple backup control barrier functions," in *IEEE Int. Conf. Robot. Autom.*, pp. 11676–11682, 2024.
- [13] P. Rabiee and A. Safari, "Safe exploration in reinforcement learning: Training backup control barrier functions with zero training-time safety violations," in *Learn. Dyn. Control Conf.*, pp. 1326–1337, PMLR, 2025.
- [14] L. Gacsi, A. K. Kiss, and T. G. Molnar, "Braking within barriers: Constructive safety-critical control for input-constrained vehicles via the backup set method," *arXiv preprint arXiv:2510.15797*, 2025.
- [15] A. W. Singletary, M. H. Cohen, T. G. Molnar, and A. D. Ames, "Safety Guardrails in the Sky: Realizing Control Barrier Functions on the VISTA F-16 Jet," *arXiv preprint arXiv:2603.27912*, 2026.
- [16] T. G. Molnar and A. D. Ames, "Safety-critical control with bounded inputs via reduced order models," in *Amer. Control Conf.*, pp. 1414–1421, 2023.
- [17] H. K. Khalil, *Nonlinear control*. Boston: Pearson, 1 ed., 2015.
- [18] A. E. Bryson and Y. Ho, *Applied Optimal Control*. Hemisphere Publishing Corporation, 1975.
- [19] T. G. Molnar, "Navigating polytopes with safety: A control barrier function approach," in *IEEE Conf. Control Technol. Appl.*, pp. 179–184, 2025.
- [20] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [21] H. K. Khalil, *Nonlinear Systems*. Pearson Education, Prentice Hall, 3 ed., 2002.