

Enforcing Mixed State-Input Constraints with Multiple Backup Control Barrier Functions: A Projection-based Approach

Laszlo Gacsi, Adam K. Kiss, Ersin Daş, and Tamas G. Molnar

Abstract—Ensuring the safety of control systems often requires the satisfaction of constraints on states (such as position or velocity), control inputs (such as force), and a mixture of states and inputs (such as power that depends on both velocity and force). This paper presents a safety-critical control framework for enforcing *mixed state-input constraints* through a generalization of backup control barrier functions (backup CBFs). First, we extend the backup CBF approach to maintain multiple *decoupled* state and input constraints using a single backup set-backup controller pair. Second, we address *mixed* state-input constraints by converting them into state constraints using a projection from the state-input space to the state space along the backup controller. In the special case of decoupled state and input constraints, the proposed method simplifies the synthesis of backup CBFs by eliminating the need for saturating backup control laws. Finally, we demonstrate the efficacy of the proposed method on an inverted pendulum example, where constraints on the angle (state), torque (input), and power (mixture of state and input) are satisfied simultaneously.

I. INTRODUCTION

Safe behavior is an essential requirement for controlling modern autonomous systems. Formally, safety is often defined as a state constraint – keeping the system’s state within a safe set in the state space – which can be enforced, for example, using *control barrier functions* (CBFs) [1]. Yet, this formulation often does not take into account the limitations of control inputs explicitly. Synthesizing safe controllers for dynamical systems subject to both state and input constraints is a common challenge in real-world control problems.

In many applications, such as robotic systems and vehicles, state and input constraints are coupled. For example, acceleration-based safety constraints for rigid-body robotic systems are both state and input-dependent [2]. For autonomous ground vehicles, the admissible stability region can vary with steering angle and braking force, making the corresponding safety constraint depend on both state and input [3]. Similarly, tipover avoidance constraints for legged and wheeled robots explicitly depend not only on states but also on inputs [2], [4]. Input-dependent constraints also arise in passivity-based teleoperation of multi-robot systems [5], and the trajectory space forward invariance problem [6].

L. Gacsi and T. G. Molnar are with the Department of Mechanical Engineering, Wichita State University, Wichita, KS 67260, USA, lxgacsi@shockers.wichita.edu, tamas.molnar@wichita.edu.

Adam K. Kiss is with the Department of Applied Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Budapest, Hungary, kiss_a@mm.bme.hu. Adam K. Kiss was supported by NKFIH (NKKP STARTING 153210) and the Janos Bolyai Research Scholarship of the Hungarian Academy of Sciences.

Ersin Daş is with the Department of Mechanical, Materials, and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL 60616, USA, edas2@illinoisistech.edu.

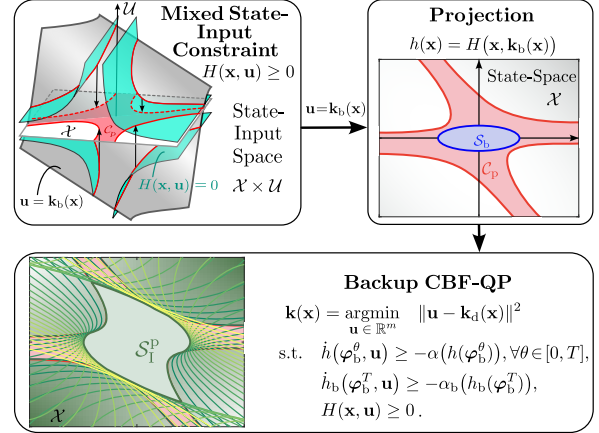


Fig. 1. Overview of the proposed projection-based backup CBF approach. A mixed state-input constraint $H(\mathbf{x}, \mathbf{u}) \geq 0$ is projected from the state-input space to the state space along the backup controller $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$. Then, a subset $\mathcal{S}_b^p$ of the resulting projected constraint set $\mathcal{C}_p$ is rendered forward invariant using the backup CBF-QP with the backup set $\mathcal{S}_b$ and flow φ_b .

To address *decoupled* state and input limits, integral CBF methods [7], [8] augment the dynamics by treating inputs as new states, whereas input constrained CBFs [9] rely on the meticulous selection of class- $\mathcal{K}$ functions. While these approaches are effective, it remains challenging to verify controller feasibility. Recently, zero-order CBFs were introduced [4] to enforce input-dependent safety via a one-step sampled-data condition; however, the resulting constraint is generally nonconvex. An alternative approach is the *backup CBF method* [10], [11] which offers both formal safety guarantees and feasibility under input bounds. Backup CBFs have been applied to mobile robots [12], spacecraft [13], and manipulators [14], with other works exploring non-smooth constraints [15], timed reach-avoid specifications [16], and dynamic safety margin [17]. Yet, backup CBFs have not been established for *mixed* (coupled) state-input constraints.

In this paper, we develop a safe control approach to enforce mixed state-input constraints using backup CBFs while preserving recursive feasibility guarantees. First, we extend the backup CBF method to handle *multiple decoupled* state and input constraints using a single backup set-backup controller pair. Then, we address *mixed* constraints. The key idea of our approach, shown in Fig. 1, is a projection from the state-input space to the state space, to transform mixed constraints into pure state constraints. If the mixed constraints are affine in the input, the proposed backup CBF-based controller synthesis problem becomes an easy-to-solve quadratic program. Moreover, for decoupled state and input constraints, our approach also helps to simplify the design of backup controllers and the calculation of backup CBFs

by obviating the need for saturating backup controllers. We showcase our approach on an inverted pendulum subject to angle (state), torque (input), and power (mixed) constraints.

II. INPUT CONSTRAINED SAFETY

The system of interest is considered in the following form:

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

with state $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$, where $\mathcal{U}$ is assumed to be a convex polytope, and smooth functions $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$. A locally Lipschitz controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$, yields the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}). \quad (2)$$

We revisit controllers that enforce state constraints by maintaining the state inside a constraint set, $\mathbf{x}(t) \in \mathcal{C} \subseteq \mathbb{R}^n$, while respecting *decoupled* input limits, $\mathbf{u}(t) \in \mathcal{U}$, for all time.

A. Classical Control Barrier Function Framework

The state constraint set $\mathcal{C}$ is defined using a smooth constraint function $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : \psi(\mathbf{x}) \geq 0\}. \quad (3)$$

The safety of system (2) is then synonymous with the forward invariance of $\mathcal{C}$: if $\mathbf{x}_0 \in \mathcal{C}$, then $\mathbf{x}(t) \in \mathcal{C}$ for all $t \geq 0$. This can be ensured using control barrier functions (CBFs).

Definition 1 ([1]). Function ψ is a *control barrier function* for (1) on $\mathcal{C}$ if there exists¹ $\alpha \in \mathcal{K}$ such that:

$$\sup_{\mathbf{u} \in \mathcal{U}} \dot{\psi}(\mathbf{x}, \mathbf{u}) > -\alpha(\psi(\mathbf{x})) \quad (4)$$

for all $\mathbf{x} \in \mathcal{C}$, with $\dot{\psi}(\mathbf{x}, \mathbf{u}) = \nabla \psi(\mathbf{x}) \cdot (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u})$.

Given a CBF, the set $\mathcal{C}$ can be rendered forward invariant.

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

$$\dot{\psi}(\mathbf{x}, \mathbf{k}(\mathbf{x})) \geq -\alpha(\psi(\mathbf{x})), \quad (5)$$

for all $\mathbf{x} \in \mathcal{C}$, renders $\mathcal{C}$ forward invariant for (2).

This result can be used for safety-critical control synthesis, by solving the quadratic program (QP):

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} \quad & \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|^2 \\ \text{s.t.} \quad & \dot{\psi}(\mathbf{x}, \mathbf{u}) \geq -\alpha(\psi(\mathbf{x})), \end{aligned} \quad (6)$$

where the desired controller $\mathbf{k}_d : \mathbb{R}^n \rightarrow \mathcal{U}$ is minimally modified in a way that $\mathcal{C}$ is rendered forward invariant. The CBF-QP in (6) is feasible if ψ is a CBF as defined in (4).

However, an arbitrary constraint function ψ may not necessarily be a CBF, especially with input constraints ($\mathcal{U} \subset \mathbb{R}^m$) which may impact the forward invariance of $\mathcal{C}$. In this case, finding CBFs can be challenging because the input that could maintain the system's safety might be outside the admissible input set. A constructive approach that addresses both state and input constraints is the *backup CBF method*.

¹A continuous function $\alpha : [0, a) \rightarrow [0, \infty)$, for some $a > 0$, belongs to class- $\mathcal{K}$ ($\alpha \in \mathcal{K}$) if it is strictly increasing and $\alpha(0) = 0$.

B. Backup Control Barrier Functions

The foundation of the backup CBF method [10], [11] is the use of a *backup controller* and an associated *backup set*. The backup set is a subset of the constraint set, defined as:

$$\mathcal{S}_b = \{\mathbf{x} \in \mathbb{R}^n : h_b(\mathbf{x}) \geq 0\}, \quad (7)$$

with the smooth function $h_b : \mathbb{R}^n \rightarrow \mathbb{R}$, such that $\mathcal{S}_b \subseteq \mathcal{C}$. The backup controller is a smooth controller $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathcal{U}$ rendering the backup set forward invariant along the corresponding closed-loop system:

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

The solution of (8) over time $\theta \geq 0$ starting from state $\mathbf{x}$ is the *backup flow* $\varphi_b(\theta, \mathbf{x})$. Note that constructing backup set–backup controller pairs is nontrivial in the presence of input bounds. While addressing this problem is beyond the scope of this paper, existing approaches can be found in [18], [19].

Typically, the backup set is a small subset of the constraint set, representing safe but (very) conservative behavior. Therefore, the backup controller $\mathbf{k}_b$ is not used directly to keep the system inside the backup set $\mathcal{S}_b$, but rather to expand $\mathcal{S}_b$ into a larger set $\mathcal{S}_I \subseteq \mathcal{C}$ that is invariant under $\mathbf{k}_b$:

$$\mathcal{S}_I = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{aligned} & \psi(\varphi_b(\theta, \mathbf{x})) \geq 0, \quad \forall \theta \in [0, T], \\ & h_b(\varphi_b(T, \mathbf{x})) \geq 0 \end{aligned} \right\}. \quad (9)$$

This set encodes the states from which the backup set can be safely reached over a horizon $T > 0$ using the backup controller. This construction enables the synthesis of controllers satisfying decoupled state and input constraints by rendering $\mathcal{S}_I$ forward invariant, as established by the following theorem.

Theorem 2 ([20]). Consider the system (1), the set $\mathcal{C}$ in (3), a backup set $\mathcal{S}_b \subseteq \mathcal{C}$ in (7), a backup controller $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathcal{U}$ that renders $\mathcal{S}_b$ forward invariant along (8), and the set $\mathcal{S}_I$ in (9) with $T > 0$. Then, there exist $\alpha, \alpha_b \in \mathcal{K}$ such that:

$$\begin{aligned} \dot{\psi}(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) & \geq -\alpha(\psi(\varphi_b(\theta, \mathbf{x}))), \quad \forall \theta \in [0, T], \\ h_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) & \geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))) \end{aligned} \quad (10)$$

holds for $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}_I$. Moreover, any locally Lipschitz controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$, that satisfies (10) for all $\mathbf{x} \in \mathcal{S}_I$ renders $\mathcal{S}_I \subseteq \mathcal{C}$ forward invariant along (2).

For control synthesis, the CBF-QP in (6) is augmented with predictive constraints, leading to the *backup CBF-QP*:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} \quad & \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|^2 \\ \text{s.t.} \quad & \dot{\psi}(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) \geq -\alpha(\psi(\varphi_b(\theta, \mathbf{x}))), \quad \forall \theta \in [0, T], \\ & h_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) \geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))). \end{aligned} \quad (11)$$

This QP is feasible even under input constraints with appropriately chosen α, α_b and it enforces state constraints (safety) on the basis of Theorem 2. In practice, the constraints in (11) are discretized into N_c constraints evaluated at discrete θ instants along the backup flow. The choice of N_c affects discretization errors, which can be overcome by constraint tightening [10], and also the complexity of the QP, which can be reduced by combining the constraints into a single

one [21]. As for the backup flow, it can be computed via forward integration of the backup system (8) using an ordinary differential equation (ODE) solver, or, when (8) is linear, it can be solved analytically; see later in Remark 6.

III. MULTIPLE BACKUP CBFs

Before addressing mixed state-input constraints, we extend the backup CBF method to handle multiple state constraints that are decoupled from input constraints. Inspired by [14], we incorporate multiple constraints using a single backup set–backup controller pair. This complements [21] where a single constraint and multiple backup controllers were used.

Hereby, we introduce N constraint functions $\psi_j : \mathbb{R}^n \rightarrow \mathbb{R}$ for $j \in \mathcal{J} = \{1, \dots, N\}$. These define the constraint set:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : \psi_j(\mathbf{x}) \geq 0, \forall j \in \mathcal{J}\}. \quad (12)$$

We consider a single backup set $\mathcal{S}_b \subseteq \mathcal{C}$ that respects all state constraints, and a backup controller $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathcal{U}$ rendering it forward invariant. Then, we redefine the invariant set in (9):

$$\mathcal{S}_I = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{array}{l} \psi_j(\varphi_b(\theta, \mathbf{x})) \geq 0, \forall j \in \mathcal{J}, \forall \theta \in [0, T], \\ h_b(\varphi_b(T, \mathbf{x})) \geq 0 \end{array} \right\}, \quad (13)$$

indicating the set of states from which the backup flow $\varphi_b(\theta, \mathbf{x})$ satisfies all N state constraints for the duration $\theta \in [0, T]$ and terminates in $\mathcal{S}_b$ at $\theta = T$. As a consequence of Theorem 2, we can state the following corollary.

Corollary 1. *Consider the system (1), the set $\mathcal{C}$ in (12), a backup set $\mathcal{S}_b \subseteq \mathcal{C}$ in (7), a backup controller $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathcal{U}$ that renders $\mathcal{S}_b$ forward invariant along (8), and the set $\mathcal{S}_I$ in (13) with $T > 0$. Then, there exist $\alpha_j, \alpha_b \in \mathcal{K}$ such that:*

$$\begin{aligned} \dot{\psi}_j(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) &\geq -\alpha_j(\psi_j(\varphi_b(\theta, \mathbf{x}))), \forall \theta \in [0, T], \forall j \in \mathcal{J}, \\ \dot{h}_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) &\geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))) \end{aligned} \quad (14)$$

holds for $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}_I$. Moreover, any locally Lipschitz controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathcal{U}$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$, that satisfies (14) for all $\mathbf{x} \in \mathcal{S}_I$ renders $\mathcal{S}_I \subseteq \mathcal{C}$ forward invariant along (2).

Proof. For each individual constraint $j \in \mathcal{J}$, let us define the constraint set $\mathcal{C}_j = \{\mathbf{x} \in \mathbb{R}^n : \psi_j(\mathbf{x}) \geq 0\}$ and invariant set:

$$\mathcal{S}_I^j = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{array}{l} \psi_j(\varphi_b(\theta, \mathbf{x})) \geq 0, \forall \theta \in [0, T], \\ h_b(\varphi_b(T, \mathbf{x})) \geq 0 \end{array} \right\}. \quad (15)$$

By definition of $\mathcal{C}$ in (12) and $\mathcal{S}_I$ in (13), $\mathbf{x} \in \mathcal{S}_I$ if and only if $\varphi_b(\theta, \mathbf{x}) \in \mathcal{C}_j, \forall \theta \in [0, T], \forall j \in \mathcal{J}$, while $\varphi_b(T, \mathbf{x}) \in \mathcal{S}_b$. This means $\mathbf{x} \in \mathcal{S}_I \iff \mathbf{x} \in \mathcal{S}_I^j, \forall j \in \mathcal{J}$, thus $\mathcal{S}_I = \bigcap_{j=1}^N \mathcal{S}_I^j$.

Based on Theorem 2, for each individual constraint $j \in \mathcal{J}$ the backup controller $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ satisfies (10) with ψ_j for all $\mathbf{x} \in \mathcal{S}_I^j$. Hence the backup controller $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ satisfies the combined constraint (14) when $\mathbf{x} \in \mathcal{S}_I$. Note that this requires that the constraints share the same backup set and backup controller. Furthermore, based on Theorem 2, for each constraint $j \in \mathcal{J}$ a locally Lipschitz controller $\mathbf{k}$ satisfying (10) with ψ_j for all $\mathbf{x} \in \mathcal{S}_I^j$ renders $\mathcal{S}_I^j$ forward invariant. Because the intersection of forward invariant sets is forward invariant under the same control law, $\mathcal{S}_I$ is forward invariant along (2) for controllers satisfying (14). Finally, since $\varphi_b(0, \mathbf{x}) = \mathbf{x}$, $\mathbf{x} \in \mathcal{S}_I$ implies $\mathbf{x} \in \mathcal{C}$. $\square$

IV. MIXED STATE-INPUT CONSTRAINTS

In this section, we present our main result: a projection-based backup CBF framework to handle mixed state-input constraints. We first introduce it for a single constraint and then extend it to the multi-constraint case.

A. Projection with a Single Constraint

We define the mixed state-input constraint set $\mathcal{D}$ via the smooth function $H : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$:

$$\mathcal{D} = \{(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^n \times \mathbb{R}^m : H(\mathbf{x}, \mathbf{u}) \geq 0\}. \quad (16)$$

Our objective is to find a locally Lipschitz controller $\mathbf{k}$ such that the closed-loop system (2) satisfies $(\mathbf{x}(t), \mathbf{k}(\mathbf{x}(t))) \in \mathcal{D}$, or equivalently $H(\mathbf{x}(t), \mathbf{k}(\mathbf{x}(t))) \geq 0$, for all $t \geq 0$ (at least for a certain set of initial conditions to be discussed later). For now, we focus on a single mixed constraint and omit other state, input, or mixed constraints—these will be addressed in the next subsection. Unlike pure state constraints, $H(\mathbf{x}, \mathbf{u})$ has *relative degree zero* because the input $\mathbf{u}$ appears explicitly. While one could be tempted to directly enforce the constraint $H(\mathbf{x}, \mathbf{u}) \geq 0$ in optimization, this may not be recursively feasible. To ensure recursive feasibility for mixed state-input constraints, we extend the backup CBF method.

Our key idea is to project the constraint from the state-input space $\mathbb{R}^n \times \mathbb{R}^m$ to the state space $\mathbb{R}^n$ along the backup controller $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$. We define the *projected constraint set*:

$$\mathcal{C}_p = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x}, \mathbf{k}_b(\mathbf{x})) \in \mathcal{D}\} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\}, \quad (17)$$

with the *projected constraint function* $h : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$h(\mathbf{x}) \triangleq H(\mathbf{x}, \mathbf{k}_b(\mathbf{x})), \quad (18)$$

which is smooth because both H and $\mathbf{k}_b$ are smooth.

We require the following for the backup set and controller.

Assumption 1. The backup set satisfies $\mathcal{S}_b \subseteq \mathcal{C}_p$, while the backup controller $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathbb{R}^m$ renders the backup set $\mathcal{S}_b$ forward invariant along (8).

Remark 1. In the special case of decoupled state and input constraints, $\mathcal{D} = \mathcal{C} \times \mathcal{U}$, Assumption 1 is similar to, but less restrictive than, the requirements of the standard backup CBF approach in Section II-B. The forward invariance of $\mathcal{S}_b$ is assumed in both cases. The key difference is that the standard backup CBF method requires $\mathcal{S}_b \subseteq \mathcal{C}$ and $\mathbf{k}_b(\mathbf{x}) \in \mathcal{U}$ for all $\mathbf{x} \in \mathbb{R}^n$. This often means that the backup controller needs to be saturated to enforce input constraints over the entire state space. In contrast, Assumption 1 requires $\mathcal{S}_b \subseteq \mathcal{C}_p$, meaning $\mathcal{S}_b \subseteq \mathcal{C}$ and $\mathbf{k}_b(\mathbf{x}) \in \mathcal{U}$ for all $\mathbf{x} \in \mathcal{S}_b$. That is, the backup controller only needs to satisfy input constraints inside the (small) backup set, eliminating the need for saturation.

With the backup set–backup controller pair, we consider the backup flow φ_b corresponding to (8), and we define the invariant set:

$$\mathcal{S}_I^p = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{array}{l} h(\varphi_b(\theta, \mathbf{x})) \geq 0, \forall \theta \in [0, T], \\ h_b(\varphi_b(T, \mathbf{x})) \geq 0 \end{array} \right\}. \quad (19)$$

We establish that the mixed state-input constraint is satisfied when using the backup controller inside this set.

Lemma 1. If $\mathbf{x} \in \mathcal{S}_I^P$, then $(\mathbf{x}, \mathbf{k}_b(\mathbf{x})) \in \mathcal{D}$.

Proof. By the definition of $\mathcal{S}_I^P$ in (19), $\mathbf{x} \in \mathcal{S}_I^P$ implies $h(\varphi_b(0, \mathbf{x})) \geq 0$. Since the backup flow at $\theta = 0$ is $\varphi_b(0, \mathbf{x}) = \mathbf{x}$, it follows that $h(\mathbf{x}) \geq 0$. By the definition of $h(\mathbf{x})$ in (18), this is equivalent to $(\mathbf{x}, \mathbf{k}_b(\mathbf{x})) \in \mathcal{D}$. $\square$

Remark 2. It is important to highlight that $\mathbf{x} \in \mathcal{S}_I^P$ does not inherently yield $(\mathbf{x}, \mathbf{k}(\mathbf{x})) \in \mathcal{D}$ for an arbitrary controller $\mathbf{k}$. Thus, keeping $\mathbf{x}$ inside $\mathcal{S}_I^P$ does not guarantee safety on its own. However, $\mathcal{S}_I^P$ helps to establish recursive feasibility because $\mathbf{k}_b$ is a controller that renders $\mathcal{S}_I^P$ forward invariant.

Lemma 2. The backup controller $\mathbf{k}_b$ renders the set $\mathcal{S}_I^P$ in (19) forward invariant along (8).

Proof. Let $\mathbf{x}_0 \in \mathcal{S}_I^P$. To show forward invariance, we establish that $\varphi_b(t, \mathbf{x}_0) \in \mathcal{S}_I^P$ for all $t \geq 0$, which is equivalent to $\varphi_b(t + \theta, \mathbf{x}_0) \in \mathcal{C}_p$ and $\varphi_b(t + T, \mathbf{x}_0) \in \mathcal{S}_b$ for all $t \geq 0$ and $\theta \in [0, T]$. Because $\mathbf{x}_0 \in \mathcal{S}_I^P$, we have $\varphi_b(t + \theta, \mathbf{x}_0) \in \mathcal{C}_p$ for $t + \theta \leq T$. Furthermore, because $\mathbf{x}_0 \in \mathcal{S}_I^P$, we also have that $\varphi_b(T, \mathbf{x}_0) \in \mathcal{S}_b$. Since $\mathcal{S}_b$ is forward invariant under $\mathbf{k}_b$, this means $\varphi_b(t + T, \mathbf{x}_0) \in \mathcal{S}_b$ for all $t \geq 0$, and also $\varphi_b(t + \theta, \mathbf{x}_0) \in \mathcal{S}_b \subseteq \mathcal{C}_p$ for any $t + \theta > T$. $\square$

We now link the forward invariance of $\mathcal{S}_I^P$ to the corresponding conditions satisfied by the backup control input.

Lemma 3. There exist $\alpha, \alpha_b \in \mathcal{K}$ such that $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ satisfies the following constraints for all $\mathbf{x} \in \mathcal{S}_I^P$:

$$\begin{aligned} \dot{h}(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) &\geq -\alpha(h(\varphi_b(\theta, \mathbf{x}))), \quad \forall \theta \in [0, T], \\ \dot{h}_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) &\geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))). \end{aligned} \quad (20)$$

Proof. This is a direct consequence of Lemma 2 and the classical backup CBF result in Theorem 2, by replacing ψ with h and considering $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathbb{R}^m$ instead of $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathcal{U}$. The proof follows that of [20, Lemma 2]. $\square$

These lemmas lead to our main theorem that states conditions for the control input to ensure safety with respect to mixed state-input constraints in a recursively feasible way.

Theorem 3. Consider the system (1), the set $\mathcal{D}$ in (16), a backup set $\mathcal{S}_b$ in (7), a backup controller $\mathbf{k}_b : \mathbb{R}^n \rightarrow \mathbb{R}^m$, the projected constraint set $\mathcal{C}_p$ in (17), and the set $\mathcal{S}_I^P$ in (19) with $T > 0$. Let Assumption 1 hold. Then, there exist $\alpha, \alpha_b \in \mathcal{K}$ such that:

$$\dot{h}(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) \geq -\alpha(h(\varphi_b(\theta, \mathbf{x}))), \quad \forall \theta \in [0, T], \quad (21a)$$

$$\dot{h}_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) \geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))), \quad (21b)$$

$$H(\mathbf{x}, \mathbf{u}) \geq 0 \quad (21c)$$

holds for $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}_I^P$. Moreover, any locally Lipschitz controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{u} = \mathbf{k}(\mathbf{x})$, that satisfies (21) for all $\mathbf{x} \in \mathcal{S}_I^P$ renders $\mathcal{S}_I^P$ forward invariant along (2), implying $(\mathbf{x}(t), \mathbf{k}(\mathbf{x}(t))) \in \mathcal{D}$ for all $t \geq 0$ if $\mathbf{x}_0 \in \mathcal{S}_I^P$.

Proof. By Lemma 3, $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ satisfies (21a) and (21b), and by Lemma 1, $\mathbf{u} = \mathbf{k}_b(\mathbf{x})$ satisfies (21c). For $\mathbf{u} = \mathbf{k}(\mathbf{x})$,

²Observe that the flow satisfies $\varphi_b(\theta, \varphi_b(t, \mathbf{x}_0)) = \varphi_b(t + \theta, \mathbf{x}_0)$ for any $t \geq 0$ and $\theta \in [0, T]$.

a unique closed-loop trajectory exists since $\mathbf{k}$ is locally Lipschitz, and the constraints in (21a) and (21b) ensure the forward invariance of $\mathcal{S}_I^P$ based on Theorem 2. This means that $\mathbf{x}_0 \in \mathcal{S}_I^P$ implies $\mathbf{x}(t) \in \mathcal{S}_I^P$ for all $t \geq 0$. Finally, the constraint in (21c) directly enforces that the mixed state-input safety condition $(\mathbf{x}(t), \mathbf{k}(\mathbf{x}(t))) \in \mathcal{D}$ is satisfied $\forall t \geq 0$. $\square$

Remark 3. The constraints (21a) and (21b) are required for guaranteeing *recursive feasibility*. They enforce $\mathbf{x}(t) \in \mathcal{S}_I^P$, thereby restricting the system to visit only those states where there exists at least one controller (the backup controller) satisfying the mixed constraints. The constraint (21c) is required for *safety*, that is, for satisfying the mixed constraints with the chosen input, such that $(\mathbf{x}(t), \mathbf{k}(\mathbf{x}(t))) \in \mathcal{D}$.

B. Projection with Multiple Constraints

By combining the ideas from Section III and Section IV-A, we extend the projection method to multiple constraints:

$$\mathcal{D} = \{(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^n \times \mathbb{R}^m : H_j(\mathbf{x}, \mathbf{u}) \geq 0, \forall j \in \mathcal{J}\}, \quad (22)$$

with smooth constraint functions $H_j : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$. This involves state constraint ($\mathcal{D} = \mathcal{C} \times \mathbb{R}^m$), input constraint ($\mathcal{D} = \mathbb{R}^n \times \mathcal{U}$), and the combination of decoupled state and input constraints ($\mathcal{D} = \mathcal{C} \times \mathcal{U}$) as special cases, while also capturing general mixed constraints in the state-input space.

We propose to use the projected constraint set:

$$\begin{aligned} \mathcal{C}_p &= \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x}, \mathbf{k}_b(\mathbf{x})) \in \mathcal{D}\} \\ &= \{\mathbf{x} \in \mathbb{R}^n : h_j(\mathbf{x}) \geq 0, \forall j \in \mathcal{J}\}, \end{aligned} \quad (23)$$

with the projected constraint functions $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$h_j(\mathbf{x}) \triangleq H_j(\mathbf{x}, \mathbf{k}_b(\mathbf{x})). \quad (24)$$

Then, with a backup set $\mathcal{S}_b$ and a backup controller $\mathbf{k}_b$ satisfying Assumption 1, we construct the invariant set:

$$\mathcal{S}_I^P = \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{aligned} &h_j(\varphi_b(\theta, \mathbf{x})) \geq 0, \forall \theta \in [0, T], \forall j \in \mathcal{J}, \\ &h_b(\varphi_b(T, \mathbf{x})) \geq 0 \end{aligned} \right\}, \quad (25)$$

see Fig. 1 (where $h_j(\varphi_b(\theta, \mathbf{x})) = 0$ is plotted for various θ).

Based on the results of the previous sections, we enforce the invariance of this set and the satisfaction of the mixed constraints using (21a) for each h_j , (21b), and (21c) for each H_j . Thus, we propose the optimization problem (OP):

$$\begin{aligned} \mathbf{k}(\mathbf{x}) &= \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|^2 \\ \text{s.t.} \quad &\dot{h}_j(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) \geq -\alpha_j(h_j(\varphi_b(\theta, \mathbf{x}))), \\ &\quad \quad \quad \forall \theta \in [0, T], \forall j \in \mathcal{J}, \\ &\dot{h}_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) \geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))), \\ &H_j(\mathbf{x}, \mathbf{u}) \geq 0, \quad \forall j \in \mathcal{J}. \end{aligned} \quad (26)$$

By construction, this OP is recursively feasible because $\mathbf{k}(\mathbf{x}) = \mathbf{k}_b(\mathbf{x})$ is a feasible solution.

Corollary 2. There exist $\alpha_j, \alpha_b \in \mathcal{K}$ such that the backup CBF-OP (26) is feasible for all $\mathbf{x} \in \mathcal{S}_I^P$. Furthermore, the controller (26) ensures for all $\mathbf{x}_0 \in \mathcal{S}_I^P$ that $\mathbf{x}(t) \in \mathcal{S}_I^P$ and $(\mathbf{x}(t), \mathbf{k}(\mathbf{x}(t))) \in \mathcal{D}$ hold for all $t \geq 0$ for system (2).

Remark 4. The backup CBF-OP (26) is a quadratic program if and only if H_j are affine in $\mathbf{u}$:

$$H_j(\mathbf{x}, \mathbf{u}) = b_j(\mathbf{x}) - \mathbf{a}_j(\mathbf{x})\mathbf{u}, \quad \forall j \in \mathcal{J}, \quad (27)$$

where $\mathbf{a}_j : \mathbb{R}^n \rightarrow \mathbb{R}^{1 \times m}$, $b_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are smooth functions.

Remark 5 (Special cases). For state constraints ($\mathbf{a}_j(\mathbf{x}) = \mathbf{0}$), the last constraint with H_j in (26) is inactive and can be omitted. For input constraints ($\mathbf{a}_j$ and b_j are constants), H_j encodes the input limits and the optimization is done over $\mathbf{u} \in \mathbb{R}^m$, rather than $\mathbf{u} \in \mathcal{U}$ as in (11), while the constraint with h_j is included. For a single state and multiple decoupled input constraints, like in Section II-B, we obtain:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|^2 \quad (28)$$

$$\begin{aligned} \text{s.t.} \quad & \dot{\psi}(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) \geq -\alpha(\psi(\varphi_b(\theta, \mathbf{x}))), \quad \forall \theta \in [0, T], \\ & -\mathbf{a}_j \dot{\mathbf{k}}_b(\varphi_b(\theta, \mathbf{x}), \mathbf{u}) \geq -\alpha_j(b_j - \mathbf{a}_j \mathbf{k}_b(\varphi_b(\theta, \mathbf{x}))), \\ & \quad \forall \theta \in [0, T], \quad \forall j \in \mathcal{J}, \\ & \dot{h}_b(\varphi_b(T, \mathbf{x}), \mathbf{u}) \geq -\alpha_b(h_b(\varphi_b(T, \mathbf{x}))), \\ & b_j - \mathbf{a}_j \mathbf{u} \geq 0, \quad \forall j \in \mathcal{J}. \end{aligned}$$

Compared to the standard backup CBF-QP (11), this includes the additional second constraint with the derivative of $\mathbf{k}_b$.

Remark 6. The projection-based approach can also simplify the computation of the backup flow. This is because, based on Remark 1, the backup controller need not be saturated. For example, if (1) is feedback linearizable, the backup controller can be constructed so that the backup system (8) becomes linear, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, and the backup flow can be obtained analytically, $\varphi_b(\theta, \mathbf{x}) = e^{\mathbf{A}\theta}\mathbf{x}$. This eliminates forward integration at each time step, saving significant computational cost for the backup CBF method. The price of these benefits is that the projection reduces the set where the system operates.

V. STATE-INPUT-CONSTRAINED INVERTED PENDULUM

We demonstrate the proposed method on an inverted pendulum with unit-coefficient equations of motion:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \sin(x_1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad (29)$$

where the state $\mathbf{x} = [x_1 \ x_2]^\top$ contains angle and angular velocity, and the input u is a torque, with power $P = x_2 u$. The primary control objective is to keep the angle x_1 within the bounds $|x_1| \leq \varphi_{\max}$ with the desired controller $\mathbf{k}_d(\mathbf{x}) = \mathbf{0}$. Moreover, we require input constraints $u_{\min} \leq u \leq u_{\max}$ and power constraints $P_{\min} \leq x_2 u \leq P_{\max}$. These mixed constraints can be expressed in the affine form (27) with:

$$H_1(\mathbf{x}, \mathbf{u}) = \varphi_{\max} - x_1, \quad H_2(\mathbf{x}, \mathbf{u}) = x_1 + \varphi_{\max}, \quad (30)$$

$$H_3(\mathbf{x}, \mathbf{u}) = u_{\max} - u, \quad H_4(\mathbf{x}, \mathbf{u}) = u - u_{\min}, \quad (31)$$

$$H_5(\mathbf{x}, \mathbf{u}) = P_{\max} - x_2 u, \quad H_6(\mathbf{x}, \mathbf{u}) = x_2 u - P_{\min}. \quad (32)$$

The parameters of this example are listed in Table I.

For the backup controller, we utilize feedback linearization and velocity feedback with gain $K > 0$, as in [14]:

$$\mathbf{k}_b(\mathbf{x}) = -\sin(x_1) - Kx_2. \quad (33)$$

TABLE I
PARAMETERS OF THE INVERTED PENDULUM EXAMPLE

Symbol	Value	Unit	Symbol	Value	Unit	Symbol	Value	Unit
$u_{\min}$	-1.1	Nm	$\varphi_{\max}$	1.75	rad	$\alpha_j(h_j)$	$15h_j$	1/s
$u_{\max}$	1.2	Nm	K	0.7	Nms	$\alpha_b(h_b)$	h_b	1/s
$P_{\min}$	-0.7	W	μ	0.001	s ⁻²	T	8	s
$P_{\max}$	0.2	W	γ	0.7	1/s	N_c	100	1

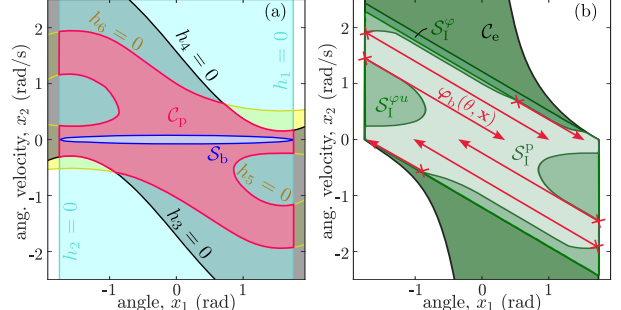


Fig. 2. Set analysis for the inverted pendulum (29) with mixed state-input constraints. Panel (a) depicts the six constraint sets from (30)–(32) that are projected using the backup controller (33) (light blue: state, gray: input, yellow: power constraints). Their intersection, C_p from (23) (red), contains the backup set S_b from (35) (blue). Panel (b) compares four sets: the zero-superlevel set C_e of the HOCBF in (36), the invariant set S_I^{φ} for state constraints (30), the set $S_I^{\varphi u}$ for state and input constraints (30)–(31), and the set S_I^p for all six constraints (30)–(32). The backup flows (34) launched from different initial conditions are also highlighted (red lines).

The six mixed state-input constraints are then projected onto the state space, creating the projected constraint set C_p in (23). Note that input limits are captured by C_p , thus it is not necessary to saturate the backup controller. Under the backup policy (33), the system becomes linear and the backup flow can be computed analytically using Remark 6:

$$\varphi_b(\theta, \mathbf{x}) = \begin{bmatrix} \varphi_1(\theta, \mathbf{x}) \\ \varphi_2(\theta, \mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1 + \frac{1}{K}(1 - e^{-K\theta})x_2 \\ e^{-K\theta}x_2 \end{bmatrix}, \quad (34)$$

thus there is no need to integrate the backup flow online, and backup CBFs become explicit instead of implicit. The backup set is chosen to be a rotated ellipse [22]:

$$h_b(\mathbf{x}) = \varphi_{\max}^2 - x_1^2 - \frac{1}{2\mu}(x_2 + K_b x_1)^2, \quad (35)$$

with $\mu > 0$. Choosing $K_b = (K - \sqrt{K^2 - 8\mu})/2$, the backup set (35) is forward invariant under the backup controller (33), because $\dot{h}_b(\mathbf{x}) = (K - K_b)x_2^2/\mu \geq 0$ for all $\mathbf{x} \in \mathbb{R}^2$.

To highlight the effect of input and power constraints, we will compare our method to a classical, high-order CBF (HOCBF) technique [23] that only enforces state limits using the set $C_e = \{\mathbf{x} \in \mathbb{R}^2 : \psi_e(\mathbf{x}) \geq 0\}$ with:

$$\psi_e(\mathbf{x}) = -2x_1 x_2 + \gamma(\varphi_{\max}^2 - x_1^2), \quad (36)$$

and $\gamma > 0$, where ψ_e is derived from $\psi(\mathbf{x}) = \varphi_{\max}^2 - x_1^2$.

Figure 2 shows the sets C_p , S_b , S_I^p for the proposed method. Panel (a) depicts the projected constraint set C_p (red) as the intersection of the zero-superlevel sets of h_j obtained from (30)–(32) (light blue, gray, and yellow areas show state, input, and power constraints, respectively). The backup set S_b (blue) from (35) lies inside C_p for the chosen parameters, confirming that Assumption 1 is satisfied and the backup controller obeys input bounds inside the backup set. Panel

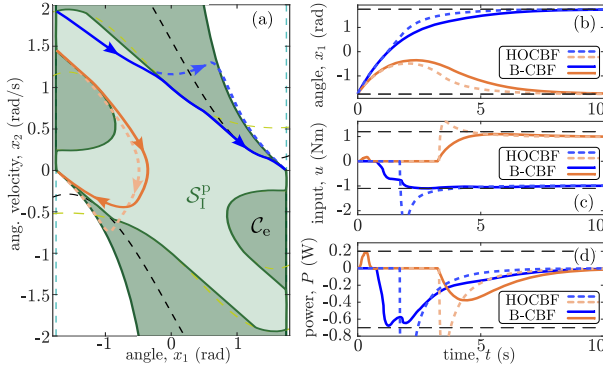


Fig. 3. Simulation of the inverted pendulum (29) using the proposed backup CBF-QP (26) (solid) and, for comparison, the HOCBF (36) (dotted). Panel (a) illustrates two trajectories (blue and orange) initiated from the invariant set $\mathcal{S}_I^P$ (green) where they remain throughout the simulation. The boundaries of the projected constraint set $\mathcal{C}_P$ are indicated by dashed lines (light blue: state, black: input, yellow: power constraint). Panels (b)–(d) display the evolution of angle, input, and power signals, with dashed black lines representing their limits. The proposed method maintains all constraints. Codes are available at <https://github.com/LaszloGacsi/State-Input-Constrained-Inverted-Pendulum>.

(b) depicts the corresponding invariant sets. $\mathcal{C}_e$ belongs to the HOCBF, $\mathcal{S}_I^\varphi$ is calculated by considering only the state constraints (30), $\mathcal{S}_I^{\varphi u}$ is based on state and input constraints (30)–(31), while $\mathcal{S}_I^P$ accounts for all constraints (30)–(32). That is, $\mathcal{S}_I^\varphi$, $\mathcal{S}_I^{\varphi u}$, and $\mathcal{S}_I^P$ are defined by (25) with $\mathcal{J} = \{1, 2\}$, $\mathcal{J} = \{1, 2, 3, 4\}$, and $\mathcal{J} = \{1, \dots, 6\}$, respectively. Notice how the sets are getting smaller. The smaller size of $\mathcal{S}_I^\varphi$ compared to $\mathcal{C}_e$ indicates the effect of using backup CBFs to ensure recursive feasibility. The size of $\mathcal{S}_I^{\varphi u}$ shows the effect of input constraints, while $\mathcal{S}_I^P$ highlights the impact of power constraints. All of these sets were computed analytically, as the backup flow (34) is explicit. The backup flow (plotted in red) exponentially approaches the horizontal axis along oblique lines of slope $-K$ without exiting $\mathcal{S}_I^\varphi$, $\mathcal{S}_I^{\varphi u}$, or $\mathcal{S}_I^P$.

Figure 3 depicts the numerical simulation of (29) using the backup CBF-QP (26) with constraints (30)–(32), backup controller (33), backup set (35), and two initial conditions (solid blue and orange). We compare this to the HOCBF method using (36) in the QP (6) while including input and power constraints directly in the QP. The HOCBF-QP becomes infeasible, hence the trajectories (dotted blue and orange) leave $\mathcal{C}_e$ slightly and violate the input and power limits. Meanwhile, the backup CBF-QP satisfies all mixed state-input constraints, even for this open-loop unstable system. This is thanks to forward prediction, which makes the backup CBF intervene earlier than the HOCBF; see panel (c).

VI. CONCLUSION

In this paper, we proposed a systematic control approach to enforcing mixed state-input constraints. We introduced a projection to convert input constraints and more complex, mixed (e.g., power) constraints to state constraints, and enforce them using backup CBFs in a recursively feasible manner. Our approach also enhances the classical backup CBF framework by eliminating the need for saturated backup controllers and potentially enabling analytical backup flow calculation. We demonstrated our theoretical results on an inverted pendulum with state, input, and power constraints.

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] Y. De Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, “Trajectory optimization for wheeled-legged quadrupedal robots using linearized ZMP constraints,” *IEEE Rob. Autom. Lett.*, vol. 4, no. 2, pp. 1633–1640, 2019.
- [3] Y. Huang, S. Z. Yong, and Y. Chen, “Stability control of autonomous ground vehicles using control-dependent barrier functions,” *IEEE Trans. Intell. Veh.*, vol. 6, no. 4, pp. 699–710, 2021.
- [4] X. Tan, E. Daş, A. D. Ames, and J. W. Burdick, “Zero-order control barrier functions for sampled-data systems with state and input dependent safety constraints,” in *Amer. Control Conf.*, 2025, pp. 283–290.
- [5] G. Notomista, “Stable, safe, and passive teleoperation of multi-robot systems,” in *IEEE Int. Conf. Robot. Autom.*, 2024, pp. 11 540–11 546.
- [6] M. Vahs, R. I. C. Muchacho, F. T. Pokorný, and J. Tumova, “Forward invariance in trajectory spaces for safety-critical control,” in *IEEE Int. Conf. Robot. Autom.*, 2025, pp. 3926–3932.
- [7] A. D. Ames, G. Notomista, Y. Wardi, and M. Egerstedt, “Integral control barrier functions for dynamically defined control laws,” *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 887–892, 2020.
- [8] E. Daş, A. D. Ames, and J. W. Burdick, “Rollover prevention for mobile robots with control barrier functions: Differentiator-based adaptation and projection-to-state safety,” *IEEE Control Syst. Lett.*, vol. 8, pp. 1727–1732, 2024.
- [9] D. R. Agrawal and D. Panagou, “Safe control synthesis via input constrained control barrier functions,” in *IEEE Conf. Decis. Control*, 2021, pp. 6113–6118.
- [10] 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. 187 249–187 275, 2020.
- [11] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, “Backup control barrier functions: Formulation and comparative study,” in *IEEE Conf. Decis. Control*, 2021, pp. 6835–6841.
- [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.*, 2024, pp. 11 676–11 682.
- [13] D. E. van Wijk, E. Daş, T. G. Molnar, A. D. Ames, and J. W. Burdick, “Safety-critical control with bounded inputs: A closed-form solution for backup control barrier functions,” *arXiv preprint arXiv:2510.05436*, 2025.
- [14] 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.
- [15] K. He, A. Alan, S. Shi, T. van den Boom, and B. De Schutter, “Predictive control barrier functions for piecewise affine systems with non-smooth constraints,” *arXiv preprint arXiv:2510.21321*, 2025.
- [16] Y. Zhang, T. Jiang, Z. Cai, and Y. Ji, “Safety-critical control under timed reach-avoid specifications: A backup control barrier function approach,” *IEEE Control Syst. Lett.*, vol. 10, pp. 109–114, 2026.
- [17] V. Freire and M. M. Nicotra, “Designing control barrier functions using a dynamic backup policy,” *arXiv preprint arXiv:2510.09810*, 2025.
- [18] 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.
- [19] L. Wang, D. Han, and M. Egerstedt, “Permissive barrier certificates for safe stabilization using sum-of-squares,” in *Amer. Control Conf.*, 2018, pp. 585–590.
- [20] T. G. Molnar and A. D. Ames, “Safety-critical control with bounded inputs via reduced order models,” in *Amer. Control Conf.*, 2023, pp. 1414–1421.
- [21] P. Rabiee and J. B. Hoagg, “Soft-minimum and soft-maximum barrier functions for safety with actuation constraints,” *Automatica*, vol. 171, p. 111921, 2025.
- [22] L. Gacsi, M. H. Cohen, and T. G. Molnar, “Activated backstepping with control barrier functions for the safe navigation of automated vehicles,” *IEEE Control Syst. Lett.*, vol. 9, pp. 721–726, 2025.
- [23] W. Xiao and C. Belta, “High-order control barrier functions,” *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3655–3662, 2022.