Safety-Critical Control with Bounded Inputs: A Closed-Form Solution for Backup Control Barrier Functions

David E. J. van Wijk[1], Ersin Daş[1], Tamas G. Molnar[2], Aaron D. Ames[1], and Joel W. Burdick[1]

***Abstract*— Verifying the safety of controllers is critical for many applications, but is especially challenging for systems with bounded inputs. Backup control barrier functions (bCBFs) offer a structured approach to synthesizing safe controllers that are guaranteed to satisfy input bounds by leveraging the knowledge of a backup controller. While powerful, bCBFs require solving a high-dimensional quadratic program at run time, which may be too costly for computationally-constrained systems such as aerospace vehicles. We propose an approach that optimally interpolates between a nominal controller and the backup controller, and we derive the solution to this optimization problem in closed form. We prove that this closed-form controller is guaranteed to be safe while obeying input bounds. We demonstrate the effectiveness of the approach on a double integrator and a nonlinear fixed-wing aircraft example.**

I. Introduction

Controlling dynamic systems with constraints that arise from physical limitations, environmental interactions, or user-defined requirements is a fundamental aspect of control applications. As such, constrained control [1]—also called *safety-critical control* when constraints encode safety requirements [2]—has become a top priority in many autonomous or semi-autonomous cyber-physical systems ranging from robotics to aeronautics. *Control barrier functions (CBFs)* [3] have emerged as a framework for synthesizing safe controllers that guarantee the forward invariance of a specified safe set. While CBFs have demonstrated success in ensuring safety in many domains [4], [5], they also highlight certain challenges. Namely, conventional CBFs require one to verify that the safe control inputs are feasible at all points within the safe set, which may be difficult in the presence of input constraints.

Motivated by this feasibility challenge, *backup control barrier functions (bCBFs)* [6] have been proposed to provide theoretical guarantees for the safety of control systems with input bounds. The bCBF technique is applicable for complex nonlinear systems, and has shown success across a wide range of applications [7]–[15]. However, a potential downside of this approach is its online computational requirements: it involves solving a quadratic program (QP) with a large number of constraints at run-time, and to construct these constraints, one must forward integrate a

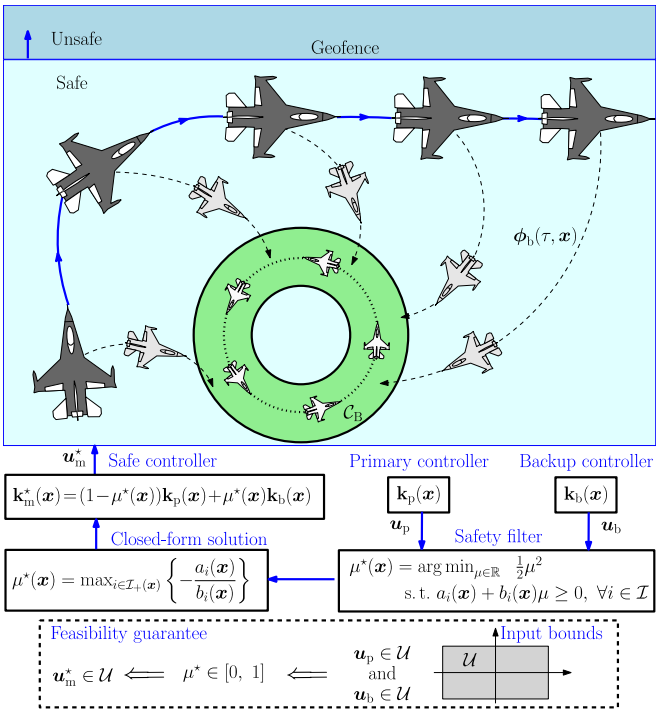

Fig. 1. Overview of the proposed safety-critical control synthesis framework, illustrated on a fixed-wing aircraft geofencing scenario. Our *optimally interpolated (OI)* controller guarantees the safety of nonlinear affine systems with input bounds and is obtained in closed-form.

number of ordinary differential equations (ODEs) whose number increases quadratically with the state dimension. This may not be feasible for high-dimensional nonlinear systems with limited online computational resources, such as aerospace vehicles. Thus, many safety-critical control solutions for aerospace vehicles have utilized CBFs rather than bCBFs (see e.g., [16]–[20]).

To remedy this, the authors of [21], [22] have introduced a technique for safe controller synthesis that builds on the concept of bCBFs without requiring optimization for the safe input. Instead, this method – which we call *safe blending* – utilizes a smooth blending function to safely combine a nominal controller and a backup controller. This approach does not require expensive gradient computations, nor does it involve solving a QP online. While effective, this solution is inherently sub-optimal, as the blending function is chosen subjectively and must be hand-tuned. Furthermore, this approach may exhibit undesirable oscillations when the nominal and backup controllers act against each other, resulting

*This work was supported by DARPA under the LINC Program and the AFOSR Test and Evaluation program, grant FA9550-22-1-0333.

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

[2]Mechanical Engineering, Wichita State University, Wichita, KS 67260, USA, tamas.molnar@wichita.edu.

in controller chattering and degraded system performance.

Inspired by these previous works, we propose a middle ground between the bCBF and the blended approach. Our main contribution is the *optimally interpolated (OI)* controller—a novel safety-critical control technique for systems with bounded inputs (see Figure 1). The proposed method optimally interpolates between a nominal controller and a pre-verified backup controller to ensure that the dynamical system remains within a controlled invariant safe set for all time. We derive a closed-form expression for the optimal safe controller, and prove that this solution is feasible even in the presence of input bounds. We demonstrate that the proposed OI controller does not experience the undesirable oscillations observed for the blending approach. Furthermore, compared to the bCBF approach, the number of ODEs to be integrated is smaller, while the closed-form expression eliminates the need for solving a high-dimensional QP numerically.

The rest of the paper is organized as follows. Section II overviews CBFs and bCBFs. Section III discusses the function-based controller blending method. In Section IV, we present the proposed optimally interpolated safety-critical control approach. In Section V, we cover the results of our numerical simulations, and we conclude with Section VI.

II. Preliminaries

A. Control Barrier Functions

Consider a nonlinear control affine system of the form

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

where $f : \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{X} \to \mathbb{R}^{n\times m}$ are smooth functions. We assume that $\mathcal{U}$ is an m-dimensional convex polytope. For an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathcal{X}$, if $\boldsymbol{u}$ is given by a locally Lipschitz feedback controller $\boldsymbol{k} : \mathcal{X} \to \mathcal{U}$, $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$, the closed-loop system has a unique solution.

Safety is defined by membership to a set $\mathcal{C}_\mathrm{S}$, and safe controllers render this safe set forward invariant.

Definition 1. A set $\mathcal{C} \subseteq \mathbb{R}^n$ is *forward invariant* along the closed-loop system if $\boldsymbol{x}(0) \in \mathcal{C} \implies \boldsymbol{x}(t) \in \mathcal{C}$, for all $t > 0$.

Consider the safe set $\mathcal{C}_\mathrm{S} \triangleq \{\boldsymbol{x} \in \mathcal{X} : h(\boldsymbol{x}) \geq 0\}$ as the 0-superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$, where the gradient of h along the boundary of $\mathcal{C}_\mathrm{S}$ is nonzero. The function h is a CBF [3] for (1) on $\mathcal{C}_\mathrm{S}$ if there exists a class-$\mathcal{K}_\infty$ function[1] α such that for all $\boldsymbol{x} \in \mathcal{C}_\mathrm{S}$

$$\sup_{\boldsymbol{u}\in\mathcal{U}} \big[\dot{h}(\boldsymbol{x},\boldsymbol{u}) \triangleq \nabla h(\boldsymbol{x}) \cdot \big(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}\big)\big] > -\alpha(h(\boldsymbol{x})).$$

Theorem 1 ([3])**.** *If h is a CBF for* (1) *on $\mathcal{C}_\mathrm{S}$, then any locally Lipschitz controller $\boldsymbol{k} : \mathcal{X} \to \mathcal{U}$, $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ satisfying*

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

for all $\boldsymbol{x} \in \mathcal{C}_\mathrm{S}$ renders the set $\mathcal{C}_\mathrm{S}$ forward invariant.

[1] A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class-$\mathcal{K}_\infty$ if it is continuous, strictly increasing, $\alpha(0) = 0$, and $\lim_{r\to\infty} \alpha(r) = \infty$.

For an arbitrary *primary* or *nominal controller*, $\boldsymbol{k}_\mathrm{p} : \mathcal{X} \to \mathcal{U}$, one can ensure the safety of (1) by solving a quadratic program (QP) for the safe control, $\boldsymbol{k}_\mathrm{safe} : \mathcal{X} \to \mathcal{U}$:

$$\boldsymbol{k}_\mathrm{safe}(\boldsymbol{x}) = \underset{\boldsymbol{u}\in\mathcal{U}}{\operatorname{argmin}}\ \ \|\boldsymbol{u} - \boldsymbol{k}_\mathrm{p}(\boldsymbol{x})\|^2 \qquad \text{(CBF-QP)}$$
$$\text{s.t.}\quad \dot{h}(\boldsymbol{x},\boldsymbol{u}) \geq -\alpha(h(\boldsymbol{x})).$$

If h is a CBF, then the (CBF-QP) is feasible, and when $\mathcal{U} = \mathbb{R}^m$, its closed-form solution becomes [23]

$$\boldsymbol{k}_\mathrm{safe}(\boldsymbol{x}) = \boldsymbol{k}_\mathrm{p}(\boldsymbol{x}) + \hat{\mu}(\boldsymbol{x})\big(\nabla h(\boldsymbol{x}) \cdot g(\boldsymbol{x})\big)^\top, \quad (3)$$
$$\hat{\mu}(\boldsymbol{x}) = \lambda\big(\hat{a}(\boldsymbol{x}), \hat{b}(\boldsymbol{x})\big),$$

where $\hat{a}$ and $\hat{b}$ stem from the coefficients in the constraint:

$$\hat{a}(\boldsymbol{x}) = \dot{h}\big(\boldsymbol{x}, \boldsymbol{k}_\mathrm{p}(\boldsymbol{x})\big) + \alpha\big(h(\boldsymbol{x})\big),$$
$$\hat{b}(\boldsymbol{x}) = \|\nabla h(\boldsymbol{x}) \cdot g(\boldsymbol{x})\|^2,$$

whereas λ is defined using $\mathrm{ReLU}(\cdot) \triangleq \max\{0, \cdot\}$ as

$$\lambda(a,b) = \begin{cases} 0 & \text{if } b \leq 0, \\ \mathrm{ReLU}(-a/b) & \text{if } b > 0. \end{cases} \quad (4)$$

Ensuring the feasibility of the (CBF-QP) can be challenging, especially for high dimensional systems. This motivates the use of an extension of CBFs known as backup CBFs.

B. Backup Control Barrier Functions

Backup CBFs (bCBFs) [6], [24] can overcome the feasibility issue of traditional CBFs by constructing controlled invariant sets at run-time for systems with input constraints.

Definition 2. A set $\mathcal{C} \subseteq \mathbb{R}^n$ is *controlled invariant* if there exists a controller $\boldsymbol{k} : \mathcal{X} \to \mathcal{U}$, $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ rendering $\mathcal{C}$ forward invariant for (1).

First, one must obtain a controlled invariant subset of $\mathcal{C}_\mathrm{S}$, known as the *backup set*:

$$\mathcal{C}_\mathrm{B} \triangleq \{\boldsymbol{x} \in \mathcal{X} : h_\mathrm{b}(\boldsymbol{x}) \geq 0\} \subseteq \mathcal{C}_\mathrm{S}, \quad (5)$$

where $h_\mathrm{b} : \mathcal{X} \to \mathbb{R}$ is continuously differentiable with a nonzero gradient along the boundary of $\mathcal{C}_\mathrm{B}$. Second, one must define a smooth *backup control law* $\boldsymbol{k}_\mathrm{b} : \mathcal{X} \to \mathcal{U}$ which renders the backup set forward invariant along the system

$$\dot{\boldsymbol{x}} = f_\mathrm{cl}(\boldsymbol{x}) \triangleq f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{k}_\mathrm{b}(\boldsymbol{x}). \quad (6)$$

Finding a small controlled invariant subset of $\mathcal{C}_\mathrm{S}$ is typically much easier than verifying that $\mathcal{C}_\mathrm{S}$ itself is controlled invariant. For example, a backup set can often be defined by a level set of a quadratic Lyapunov function centered on a stabilizable equilibrium point for the linearized dynamics, and can be rendered forward invariant by a simple feedback controller [24]. One could directly constrain the system (1) to operate within $\mathcal{C}_\mathrm{B}$ to guarantee safety. However, as $\mathcal{C}_\mathrm{B}$ may be a very small subset of $\mathcal{C}_\mathrm{S}$, this would lead to conservative behavior. Instead, it is possible to expand this set implicitly, thereby obtaining a new, larger controlled invariant set.

To construct a controlled invariant set online, we allow the system to evolve beyond $\mathcal{C}_\mathrm{B}$ by forward integrating the

backup dynamics (6) over a finite horizon. If the system can stay in $\mathcal{C}_\mathrm{S}$ and safely reach $\mathcal{C}_\mathrm{B}$ from the current state using $\boldsymbol{k}_\mathrm{b}$, this state is classified as safe. More precisely, the controlled invariant set, $\mathcal{C}_\mathrm{I} \subseteq \mathcal{C}_\mathrm{S}$, is defined as

$$\mathcal{C}_\mathrm{I} \triangleq \left\{ \boldsymbol{x} \in \mathcal{X} \;\middle|\; \begin{array}{c} h(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})) \geq 0, \forall \tau \in [0, T], \\ h_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x})) \geq 0 \end{array} \right\}, \quad (7)$$

where $\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})$ is the *flow* of the backup system (6) over the interval $\tau \in [0, T]$ for a horizon $T > 0$ starting at state $\boldsymbol{x}$:

$$\frac{\partial}{\partial \tau} \boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}) = f_\mathrm{cl}(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})), \;\; \boldsymbol{\phi}_\mathrm{b}(0, \boldsymbol{x}) = \boldsymbol{x}. \quad (8)$$

We note that the backup horizon T is a design parameter which directly affects the size of $\mathcal{C}_\mathrm{I}$. Taking $T = 0$ yields $\mathcal{C}_\mathrm{I} = \mathcal{C}_\mathrm{B}$, whilst the largest possible parameterization of $\mathcal{C}_\mathrm{I}$ is achieved when $T \to \infty$. Since the flow $\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})$ is typically computed numerically, picking a larger T increases the computation time, therefore T shall be selected such that computations are quick enough to be run online at high rates.

We now review a few important results about $\mathcal{C}_\mathrm{I}$.

Lemma 1 ([24][25, Lem. 1])**.** *The set $\mathcal{C}_\mathrm{I}$ is controlled invariant, and the backup controller $\boldsymbol{k}_\mathrm{b}$ renders $\mathcal{C}_\mathrm{I}$ forward invariant along* (6) *as* $\boldsymbol{x} \in \mathcal{C}_\mathrm{I} \implies \boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}) \in \mathcal{C}_\mathrm{I} \subseteq \mathcal{C}_\mathrm{S}, \forall \tau \geq 0$.

As a direct consequence of Lemma 1, the backup control law can be shown to satisfy safety constraints similar to (2).

Lemma 2 ([24][25, Lem. 2])**.** *There exist class-$\mathcal{K}_\infty$ functions α, α_b such that*

$$\dot{h}\big(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}), \boldsymbol{k}_\mathrm{b}(\boldsymbol{x})\big) \geq -\alpha\big(h(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}))\big), \quad (9a)$$

$$\dot{h}_\mathrm{b}\big(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x}), \boldsymbol{k}_\mathrm{b}(\boldsymbol{x})\big) \geq -\alpha_\mathrm{b}\big(h_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x}))\big), \quad (9b)$$

for all $\tau \in [0, T]$ and $\boldsymbol{x} \in \mathcal{C}_\mathrm{I}$.

This therefore leads to the main result of backup CBFs.

Theorem 2 ([24])**.** *There exist class-$\mathcal{K}_\infty$ functions α, α_b and a locally Lipschitz controller $\boldsymbol{k} : \mathcal{X} \to \mathcal{U}$, $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ satisfying*

$$\dot{h}(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}), \boldsymbol{u}) \geq -\alpha\big(h(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}))\big), \;\; \forall \tau \in [0, T], \quad (10a)$$

$$\dot{h}_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x}), \boldsymbol{u}) \geq -\alpha_\mathrm{b}\big(h_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x}))\big), \quad (10b)$$

for all $\boldsymbol{x} \in \mathcal{C}_\mathrm{I}$. Such a controller renders $\mathcal{C}_\mathrm{I} \subseteq \mathcal{C}_\mathrm{S}$ forward invariant and satisfies input constraints.

Proof. By Lemma 1, the backup control law $\boldsymbol{k}_\mathrm{b}$ is one such Lipschitz controller. Applying Theorem 1 to system (1), the satisfaction of (10) yields the forward invariance of $\mathcal{C}_\mathrm{I}$. ■

Because (10a) represents an infinite number of constraints, in practice the constraint is discretized and enforced at discrete points along the flow, such that $\tau \in \{0, \Delta, \ldots, T\}$ where $\Delta > 0$ is a discretization step satisfying $T/\Delta \in \mathbb{N}$. Then, one can construct a point-wise optimization problem for the least-invasive safe control signal, like the (CBF-QP):

$$\boldsymbol{k}_\mathrm{safe}(\boldsymbol{x}) = \underset{\boldsymbol{u} \in \mathcal{U}}{\operatorname{argmin}} \;\; \|\boldsymbol{u} - \boldsymbol{k}_\mathrm{p}(\boldsymbol{x})\|^2 \quad \text{(bCBF-QP)}$$
$$\text{s.t. (10a), (10b).}$$

Corollary 1. *There exist class-$\mathcal{K}_\infty$ functions α, α_b such that the optimization problem posed in* (bCBF-QP) *is feasible for all $\boldsymbol{x} \in \mathcal{C}_\mathrm{I}$, even if h is not a CBF.*

Thus, bCBFs are powerful tools for guaranteeing safety of nonlinear affine systems in the presence of input constraints. However, the price for the feasibility guarantees endowed by bCBFs is additional computation. Computing the total derivatives of h and h_b:

$$\dot{h}(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x}), \boldsymbol{u}) = \nabla h(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})) \cdot \boldsymbol{\Phi}_\mathrm{b}(\tau, \boldsymbol{x}) \dot{\boldsymbol{x}}, \quad (11a)$$

$$\dot{h}_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x}), \boldsymbol{u}) = \nabla h_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x})) \cdot \boldsymbol{\Phi}_\mathrm{b}(T, \boldsymbol{x}) \dot{\boldsymbol{x}}, \quad (11b)$$

where $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$, and $\boldsymbol{\Phi}_\mathrm{b}(\tau, \boldsymbol{x}) \triangleq \frac{\partial \boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})}{\partial \boldsymbol{x}}$, requires solving a coupled initial value problem for the sensitivities of the flow to changes in the initial condition. Namely, the sensitivity matrix $\boldsymbol{\Phi}_\mathrm{b}(\tau, \boldsymbol{x})$ is the solution to

$$\frac{\partial}{\partial \tau} \boldsymbol{\Phi}_\mathrm{b}(\tau, \boldsymbol{x}) = F_\mathrm{cl}(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})) \boldsymbol{\Phi}_\mathrm{b}(\tau, \boldsymbol{x}), \;\; \boldsymbol{\Phi}_\mathrm{b}(0, \boldsymbol{x}) = \boldsymbol{I}, \quad (12)$$

where $F_\mathrm{cl}(\boldsymbol{x}) = \partial f_\mathrm{cl}(\boldsymbol{x}) / \partial \boldsymbol{x}$ is the Jacobian of f_cl in (6), that is evaluated at $\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})$, and $\boldsymbol{I}$ is the $n \times n$ identity matrix.

To compute the constraints in (10), at each time step one must forward integrate $n + n^2$ ordinary differential equations (ODEs) in (8) and (12). For many applications this may be allowable, but for resource-constrained platforms, such as aerospace systems, computing these gradients may be a bottleneck, especially if the backup horizon is long. Further, formal system certification requirements may exclude methods that numerically solve a high-dimensional QP online.

III. Function-Based Controller Blending

While bCBFs offer an elegant solution to feasible safety-critical control, their main downside is the additional computation requirements. To overcome this downside, the authors of [21] offer a solution deemed *safe blending*, which seeks to mix the control inputs from the nominal and the backup controller, to obtain a safe but more performant controller than just the backup controller alone. This blended or *mixed* controller $\boldsymbol{k}_\mathrm{m} : \mathcal{X} \to \mathcal{U}$ is written as

$$\boldsymbol{k}_\mathrm{m}(\boldsymbol{x}) = (1 - \mu(\boldsymbol{x})) \boldsymbol{k}_\mathrm{p}(\boldsymbol{x}) + \mu(\boldsymbol{x}) \boldsymbol{k}_\mathrm{b}(\boldsymbol{x}), \quad (13)$$

where $\mu : \mathbb{R}^n \to [0, 1]$ is a blending function that regulates the portion that the backup and primary control signals contribute to the mixed control signal. It is trivial to verify that restricting the range of μ from $[0, 1]$ guarantees that for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{k}_\mathrm{m}(\boldsymbol{x}) \in \mathcal{U}$ as long as $\boldsymbol{k}_\mathrm{p}(\boldsymbol{x}), \boldsymbol{k}_\mathrm{b}(\boldsymbol{x}) \in \mathcal{U}$, as $\mathcal{U}$ is convex.

To ensure safety with the mixed controller, the authors of [21] define a function h_I associated with the set $\mathcal{C}_\mathrm{I}$ in (7) as

$$h_\mathrm{I}(\boldsymbol{x}) \triangleq \min \left\{ \min_{\tau \in [0, T]} h(\boldsymbol{\phi}_\mathrm{b}(\tau, \boldsymbol{x})), h_\mathrm{b}(\boldsymbol{\phi}_\mathrm{b}(T, \boldsymbol{x})) \right\}.$$

Using this definition, a blending function is proposed as

$$\mu(\boldsymbol{x}) = \Lambda(h_\mathrm{I}(\boldsymbol{x})), \quad (14)$$

where $\Lambda : \mathbb{R} \to [0, 1]$ is a continuous function that satisfies $\Lambda(r) = 1$ when $r \leq 0$, and $\frac{d\Lambda}{dr} < 0$ when $r > 0$. For example,

$$\Lambda(h_\mathrm{I}(\boldsymbol{x})) = \mathrm{e}^{-\eta \max\{h_\mathrm{I}(\boldsymbol{x}), 0\}}, \quad (15)$$

with $\eta > 0$. This particular choice of blending function and blending controller is provably safe.

Proposition 1 ([22, Proposition 1])**.** *Any locally Lipschitz mixed controller $\boldsymbol{k}_{\mathrm{m}} : \mathcal{X} \to \mathcal{U}$ given by* (13)*, with a blending function μ in* (14) *and $\Lambda : \mathbb{R} \to [0, 1]$ satisfying $\Lambda(0) = 1$, renders $\mathcal{C}_{\mathrm{I}}$ forward invariant along* (1)*.*

Proof. At the boundary of $\mathcal{C}_{\mathrm{I}}$, where $h_{\mathrm{I}}(\boldsymbol{x}) = 0$, the blending parameter in (13) is $\mu(\boldsymbol{x}) = 1$ and the mixed controller in (14) reduces to $\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}) = \boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})$. Therefore, because the set $\mathcal{C}_{\mathrm{I}}$ is forward invariant under the backup controller based on Lemma 1, $\mathcal{C}_{\mathrm{I}}$ is rendered forward invariant under $\boldsymbol{k}_{\mathrm{m}}$ as well. Furthermore, $\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}) \in \mathcal{U}$ for all $\boldsymbol{x} \in \mathcal{X}$ by design. ■

The mixed controller in (13) is not only provably safe, but it also circumvents the need to solve an optimization problem online and to calculate the sensitivity of the backup flow by solving n^2 ODEs in (12). However, the mixed controller can lead to undesirable behavior near the boundary of $\mathcal{C}_{\mathrm{I}}$ if the goals of the primary controller and backup controller conflict. We use a simple example to highlight this point.

Example 1 (Double Integrator)**.** Consider a double integrator system as in [8] given by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 & u \end{bmatrix}^\top, \tag{16}$$

with position x_1, velocity x_2, state $\boldsymbol{x} = [x_1 \; x_2]^\top \in \mathbb{R}^2$, and control input $u \in \mathcal{U} = [-1, 1]$. The safe set is defined as the left half-plane $\mathcal{C}_{\mathrm{S}} \triangleq \{\boldsymbol{x} \in \mathbb{R}^2 : -x_1 \geq 0\}$. The backup control law, $\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x}) = -1$, brings the system to the backup set $\mathcal{C}_{\mathrm{B}} \triangleq \{\boldsymbol{x} \in \mathbb{R}^2 : -x_1 \geq 0, -x_2 \geq 0\}$ and renders $\mathcal{C}_{\mathrm{B}}$ forward invariant. The primary controller, $\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x}) = 1$, drives (16) to the unsafe right half-plane.

Figure 2 compares the (bCBF-QP) to the blended controller from (13) using the blending function in (14)-(15). Initially, the behavior of the controllers is similar; however, at the origin, the blended controller exhibits undesirable oscillations. This is due to the competing goals of the backup and the primary controller, which the blended controller mixes. The primary controller seeks to enter the right half-plane while the backup controller seeks to apply maximum braking. Thus, near the boundary of the safe set, the blended controller modulates the importance between these two control signals, but cannot do so in an optimal manner, as done by the (bCBF-QP). This motivates our central contribution to find a simple and computationally efficient method for input-constrained safe control without such undesirable behavior.

IV. Optimal Controller Interpolation

As discussed in Section III, the main advantage of function-based blending is that it is a *gradient-free* approach (i.e., computing the sensitivity via (12) is not needed) and it does not require solving a QP online. However, the choice of blending function is subjective and may lead to undesirable behavior (see Example 1). Therefore, we propose a middle ground between function-based blending and the (bCBF-QP), which seeks to select an optimal blending parameter.

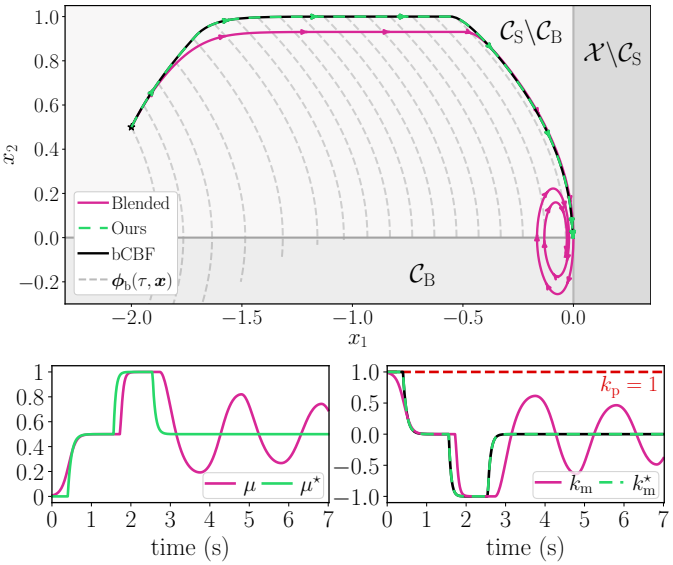

Fig. 2. Simulation of the double integrator (16) comparing the (bCBF-QP) (black), the function-based blended controller in (13) with tuning parameter (14)-(15) (pink) and the proposed (OI-QP) approach (green dashed). The blended controller experiences undesirable oscillations at the boundary of the safe region (**top**), while our approach does not. In this scenario, our approach is equivalent to the (bCBF-QP) (**bottom right**), but with significantly less computational overhead.

We introduce the following optimization problem, where the parameter μ is minimized subject to conditions on the forward invariance of the implicit set $\mathcal{C}_{\mathrm{I}} \subseteq \mathcal{C}_{\mathrm{S}}$, which we denote as the *optimally interpolated (OI)* controller:

$$\begin{aligned} \mu^\star(\boldsymbol{x}) = \underset{\mu \in [0,1]}{\operatorname{argmin}} \; & \frac{1}{2}\mu^2 \qquad \text{(OI-QP)} \\ \text{s.t. } & \dot{h}(\boldsymbol{\phi}_{\mathrm{b}}(\tau, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})) \geq -\alpha\big(h(\boldsymbol{\phi}_{\mathrm{b}}(\tau, \boldsymbol{x}))\big), \;\; \forall \tau \in [0, T], \\ & \dot{h}_{\mathrm{b}}(\boldsymbol{\phi}_{\mathrm{b}}(T, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})) \geq -\alpha_{\mathrm{b}}\big(h_{\mathrm{b}}(\boldsymbol{\phi}_{\mathrm{b}}(T, \boldsymbol{x}))\big). \end{aligned}$$

Based on (13), the total derivatives are computed as

$$\begin{aligned} \dot{h}(\boldsymbol{\phi}_{\mathrm{b}}(\tau, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})) &= \dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big) \qquad \text{(17a)} \\ &\quad + \mu\Big(\dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big) - \dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big)\Big), \\ \dot{h}_{\mathrm{b}}(\boldsymbol{\phi}_{\mathrm{b}}(T, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x})) &= \dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big) \qquad \text{(17b)} \\ &\quad + \mu\Big(\dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big) - \dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T, \boldsymbol{x}), \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big)\Big), \end{aligned}$$

leading to a QP with affine constraints in μ.

Similar to the (bCBF-QP), the (OI-QP) can be shown to be feasible and guarantee safety.

Lemma 3. *There exist class-$\mathcal{K}_\infty$ functions α, α_{b} such that the* (OI-QP) *is feasible for all $\boldsymbol{x} \in \mathcal{C}_{\mathrm{I}}$.*

Proof. Based on (17), the constraints of the (OI-QP) reduce to those in (9) for $\mu = 1$. Therefore, based on Lemma 2, there exist α, α_{b} such that $\mu = 1$ satisfies the constraints of the optimization problem, which implies feasibility. ■

Theorem 3. *Consider the system* (1) *with a primary controller $\boldsymbol{k}_{\mathrm{p}}$, a backup controller $\boldsymbol{k}_{\mathrm{b}}$, and the set $\mathcal{C}_{\mathrm{I}}$ in* (7)*. The*

controller $\boldsymbol{k}_{\mathrm{m}}^{\star}:\mathcal{X}\to\mathcal{U}$ *given by*

$$\boldsymbol{k}_{\mathrm{m}}^{\star}(\boldsymbol{x})=(1-\mu^{\star}(\boldsymbol{x}))\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})+\mu^{\star}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x}), \tag{18}$$

where $\mu^{\star}(\boldsymbol{x})$ *satisfies the* (OI-QP)*, renders* $\mathcal{C}_{\mathrm{I}}\subseteq\mathcal{C}_{\mathrm{S}}$ *forward invariant and satisfies the input constraint,* $\boldsymbol{k}_{\mathrm{m}}^{\star}(\boldsymbol{x})\in\mathcal{U}$ *for all* $\boldsymbol{x}\in\mathcal{X}$.

Proof. The forward invariance of $\mathcal{C}_{\mathrm{I}}$ follows from Theorem 2, since the controller in (18) satisfies the constraints in (10) by construction. The controller in (18) satisfies the input constraint because for all $\boldsymbol{x}\in\mathcal{X}$, $\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\in\mathcal{U}$, $\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\in\mathcal{U}$, $\mu^{\star}(\boldsymbol{x})\in[0,1]$, and $\mathcal{U}$ is convex. ■

While maintaining feasibility and safety, the (OI-QP) offers major computational advantages over the (bCBF-QP). Namely, the (OI-QP) can be solved in closed-form, which eliminates the need for implementing a high-dimensional QP solver. Moreover, the number of ODEs to be integrated at each time step can be reduced by using the (OI-QP). At the same time, the (OI-QP) can also eliminate the undesirable behavior observed for the function-based blending approach. We demonstrate these properties next. We remark that the price of these benefits is the restriction of the control input to the linear combination of the primary and backup control signals as in (18) as opposed to the (bCBF-QP) that offers a point-wise optimal input from the entire input set $\mathcal{U}$.

A. Closed-form Solution for Optimal Interpolation

First, we present the solution of the (OI-QP) in closed-form. As a preliminary, we introduce the notation:

$$a(\tau,\boldsymbol{x})\triangleq\dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}),\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big)+\alpha\big(h(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}))\big), \tag{19a}$$
$$b(\tau,\boldsymbol{x})\triangleq\dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}),\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big)-\dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}),\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big), \tag{19b}$$
$$a_{\mathrm{b}}(\boldsymbol{x})\triangleq\dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x}),\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big)+\alpha_{\mathrm{b}}\big(h_{\mathrm{b}}(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x}))\big), \tag{19c}$$
$$b_{\mathrm{b}}(\boldsymbol{x})\triangleq\dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x}),\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big)-\dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x}),\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big), \tag{19d}$$

which simplifies the constraints to $a(\tau,\boldsymbol{x})+b(\tau,\boldsymbol{x})\mu\geq 0$ and $a_{\mathrm{b}}(\boldsymbol{x})+b_{\mathrm{b}}(\boldsymbol{x})\mu\geq 0$. Then, as for the (bCBF-QP), we discretize the constraints on the safety of the backup flow by introducing $a_i(\boldsymbol{x})=a(\tau_i,\boldsymbol{x})$ and $b_i(\boldsymbol{x})=b(\tau_i,\boldsymbol{x})$ with $\tau_i=i\Delta$, $\Delta=T/N$, $i\in\{0,1,\dots,N\}$. This leads to

$$\begin{aligned}\mu^{\star}(\boldsymbol{x})=\underset{\mu\in\mathbb{R}}{\operatorname{argmin}}\ &\frac{1}{2}\mu^2\\ &\text{s.t. } a_i(\boldsymbol{x})+b_i(\boldsymbol{x})\mu\geq 0,\ \forall i\in\mathcal{I},\end{aligned} \tag{20}$$

where $\mathcal{I}=\{0,1,\dots,N+3\}$, whereas $a_{N+1}(\boldsymbol{x})=a_{\mathrm{b}}(\boldsymbol{x})$, $b_{N+1}(\boldsymbol{x})=b_{\mathrm{b}}(\boldsymbol{x})$, while the domain $\mu\in[0,1]$ is captured by $a_{N+2}(\mathbf{x})=0$, $b_{N+2}(\mathbf{x})=1$, $a_{N+3}(\mathbf{x})=1$, $b_{N+3}(\mathbf{x})=-1$.

Theorem 4. *The QP* (20) *has the closed-form solution*

$$\mu^{\star}(\boldsymbol{x})=\max_{i\in\mathcal{I}_+(\boldsymbol{x})}\Big\{-\frac{a_i(\boldsymbol{x})}{b_i(\boldsymbol{x})}\Big\}\in[0,1], \tag{21}$$

where $\mathcal{I}_+(\boldsymbol{x})\triangleq\{i\in\mathcal{I}:b_i(\boldsymbol{x})>0\}$.

Proof. We first notice that the constraints of the discretized QP (20) are a subset of the constraints of the (OI-QP). Therefore, the QP (20) is feasible by Lemma 3 (considering appropriate α and α_{b}). Because the primal problem in (20) is a QP and is feasible, Slater's conditions hold and thus strong duality holds. Therefore, any points $(\mu^{\star},\lambda_i)$ that satisfy the Karush–Kuhn–Tucker (KKT) conditions are primal and dual optimal, and have zero duality gap [26, Ch. 5]. That is, the solution of the QP (20) must satisfy the KKT conditions:

$$a_i+b_i\mu^{\star}\geq 0, \tag{22a}$$
$$\lambda_i\geq 0, \tag{22b}$$
$$\lambda_i(a_i+b_i\mu^{\star})=0, \tag{22c}$$
$$\mu^{\star}-\sum_{i\in\mathcal{I}}\lambda_i b_i=0, \tag{22d}$$

for all $i\in\mathcal{I}$. Note that, throughout this proof, we omit the dependencies on $\boldsymbol{x}$ to simplify our notation.

For ease of notation, let $j\triangleq\operatorname{argmax}_{i\in\mathcal{I}_+}\{-a_i/b_i\}$. If multiple indices yield the maximum, j can be any of them since the following equations hold for any choice of j. We prove that the KKT conditions in (22) are satisfied by

$$\mu^{\star}=-\frac{a_j}{b_j}, \tag{23a}$$
$$\lambda_i=0,\ \forall i\in\mathcal{I}\backslash\{j\}, \tag{23b}$$
$$\lambda_j=-\frac{a_j}{b_j^2}, \tag{23c}$$

where $\mu^{\star}$ matches (21). Note that, by definition of j,

$$-\frac{a_j}{b_j}\geq-\frac{a_i}{b_i},\ \forall i\in\mathcal{I}_+, \tag{24a}$$
$$b_j>0 \text{ and } a_j\leq 0, \tag{24b}$$

where $a_j\leq 0$ holds because $a_{N+2}=0$ and $b_{N+2}=1$. Then, the expressions in (23) satisfy (22d) because

$$\sum_{i\in\mathcal{I}}\lambda_i b_i\overset{(23b)}{=}\lambda_j b_j\overset{(23c)}{=}-\frac{a_j}{b_j}\overset{(23a)}{=}\mu^{\star}.$$

The condition in (22c) holds, because (23b) gives $\lambda_i=0$ for $i\in\mathcal{I}\backslash\{j\}$, whereas (23a) gives $a_i+b_i\mu^{\star}=0$ for $i=j$. The condition in (22b) holds with (23b) and (23c)-(24b).

Finally, to show that (22a) holds, we leverage the fact that the QP (20) is feasible (per Lemma 3), that is, there exists μ such that $a_i+b_i\mu\geq 0$ for all $i\in\mathcal{I}$. This implies $a_i\geq 0$ for $b_i=0$, $\mu\leq-a_i/b_i$ for $b_i<0$, and $\mu\geq-a_i/b_i$ for $b_i>0$. Therefore, the following must hold for μ to exist:

$$a_i\geq 0 \text{ if } b_i=0, \tag{25a}$$
$$-\frac{a_i}{b_i}\geq-\frac{a_j}{b_j} \text{ if } b_i<0, \tag{25b}$$

where (25b) ensures that the upper bounds on μ are higher than the lower bounds. Then, (22a) is satisfied because

$$a_i+b_i\mu^{\star}\overset{(23a)}{=}a_i+b_i\Big(-\frac{a_j}{b_j}\Big)\geq 0,$$

where the last inequality holds for $b_i=0$ due to (25a), for $b_i>0$ because of (24a), and also for $b_i<0$ due to (25b). Thus, the expressions in (23) satisfy all the conditions for primal and dual optimality. ■

The solution (21) of the QP (20) and the corresponding

controller can be written equivalently as

$$\begin{aligned} \boldsymbol{k}_{\mathrm{m}}^{\star}(\boldsymbol{x}) &= \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x}) + \mu^{\star}(\boldsymbol{x})\big(\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x}) - \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big), \\ \mu^{\star}(\boldsymbol{x}) &= \max_{i\in\mathcal{I}} \lambda\big(a_i(\boldsymbol{x}), b_i(\boldsymbol{x})\big), \end{aligned} \tag{26}$$

with λ in (4), where we used $a_{N+2}(\boldsymbol{x}) = 0$, $b_{N+2}(\boldsymbol{x}) = 1$. Notice the similarity to the solution (3) of the (CBF-QP). While the (CBF-QP) combines the primary controller $\boldsymbol{k}_{\mathrm{p}}$ with the input $(\nabla h(\boldsymbol{x}) \cdot g(\boldsymbol{x}))^{\top}$, that represents the safest control direction based on h, the proposed QP (20) blends $\boldsymbol{k}_{\mathrm{p}}$ with the backup controller $\boldsymbol{k}_{\mathrm{b}}$ to ensure the satisfaction of input constraints in addition to safety.

By refining the discretization and taking the continuous limit ($N \to \infty$), the solution (26) of the discretized QP (20) leads to the closed-form solution of the (OI-QP):

$$\begin{aligned} \mu^{\star}(\boldsymbol{x}) = \max\Big\{ &\max_{\tau\in[0,T]} \lambda\big(a(\tau,\boldsymbol{x}), b(\tau,\boldsymbol{x})\big), \\ &\lambda\big(a_{\mathrm{b}}(\boldsymbol{x}), b_{\mathrm{b}}(\boldsymbol{x})\big), 0\Big\}, \end{aligned} \tag{27}$$

where a, b, a_{b}, b_{b} are defined in (19), while λ is given by (4).

B. Sensitivity Computations for Optimal Interpolation

Next, we establish that the (OI-QP) requires fewer ODEs to be integrated than the (bCBF-QP). In the form (17), the derivatives can be computed *exactly*:

$$\dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}), \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big) = \nabla h(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x})) \cdot \boldsymbol{q}_{\mathrm{p}}(\tau,\boldsymbol{x}), \tag{28a}$$

$$\dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x}), \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big) = \nabla h_{\mathrm{b}}(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x})) \cdot \boldsymbol{q}_{\mathrm{p}}(T,\boldsymbol{x}), \tag{28b}$$

$$\dot{h}\big(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}), \boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big) = \nabla h(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x})) \cdot \boldsymbol{q}_{\mathrm{b}}(\tau,\boldsymbol{x}), \tag{28c}$$

$$\dot{h}_{\mathrm{b}}\big(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x}), \boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big) = \nabla h_{\mathrm{b}}(\boldsymbol{\phi}_{\mathrm{b}}(T,\boldsymbol{x})) \cdot \boldsymbol{q}_{\mathrm{b}}(T,\boldsymbol{x}), \tag{28d}$$

where $\boldsymbol{q}_{\mathrm{p}}(\tau,\boldsymbol{x}) \triangleq \boldsymbol{\Phi}_{\mathrm{b}}(\tau,\boldsymbol{x})\big(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})\big)$ and $\boldsymbol{q}_{\mathrm{b}}(\tau,\boldsymbol{x}) \triangleq \boldsymbol{\Phi}_{\mathrm{b}}(\tau,\boldsymbol{x})\big(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x})\big)$ are the push forward of the dynamics, representing the change of the backup flow under $\boldsymbol{k}_{\mathrm{p}}$ and $\boldsymbol{k}_{\mathrm{b}}$, respectively.

Although these terms involve the sensitivity matrix $\boldsymbol{\Phi}_{\mathrm{b}}(\tau,\boldsymbol{x})$, the key intuition is that we do not need to compute the entire sensitivity matrix by solving (12). Instead, we only need to obtain directional components of the sensitivity matrix, by solving the following initial value problems:

$$\begin{aligned} \frac{\partial}{\partial\tau}\boldsymbol{q}_{\mathrm{p}}(\tau,\boldsymbol{x}) &= F_{\mathrm{cl}}(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}))\boldsymbol{q}_{\mathrm{p}}(\tau,\boldsymbol{x}),\ \boldsymbol{q}_{\mathrm{p}}(0,\boldsymbol{x}) = f_{\mathrm{p}}(\boldsymbol{x}), \\ \frac{\partial}{\partial\tau}\boldsymbol{q}_{\mathrm{b}}(\tau,\boldsymbol{x}) &= F_{\mathrm{cl}}(\boldsymbol{\phi}_{\mathrm{b}}(\tau,\boldsymbol{x}))\boldsymbol{q}_{\mathrm{b}}(\tau,\boldsymbol{x}),\ \boldsymbol{q}_{\mathrm{b}}(0,\boldsymbol{x}) = f_{\mathrm{cl}}(\boldsymbol{x}), \end{aligned} \tag{29}$$

where $f_{\mathrm{p}}(\boldsymbol{x}) \triangleq f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x})$, while f_{cl} is given in (6), and $F_{\mathrm{cl}}(\boldsymbol{x}) = \partial f_{\mathrm{cl}}(\boldsymbol{x})/\partial\boldsymbol{x}$; cf. (12).

Thus, this method requires integrating forward $3n$ ODEs, given by (8) and (29), rather than the $n + n^2$ ODEs required by the standard bCBF approach, reducing the number of ODEs if $n \geq 2$. This may save a significant amount of computation for high-dimensional systems.

The downside to this approach (and the approach in Section III) is that the components of the backup and primary controllers are mixed with equal weighting. To improve performance, one could construct an optimization problem similar to that posed in (OI-QP), but for a multi-dimensional blending variable, $\boldsymbol{\mu} = \begin{bmatrix}\mu_1 & \mu_2 & \dots & \mu_m\end{bmatrix}^{\top}$ with the corresponding mixed controller $\boldsymbol{k}_{\mathrm{m}}(\boldsymbol{x}) = \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x}) + \Delta_{\boldsymbol{u}}(\boldsymbol{x})\boldsymbol{\mu}(\boldsymbol{x})$ where $\Delta_{\boldsymbol{u}}(\boldsymbol{x}) \triangleq \mathrm{diag}(\boldsymbol{k}_{\mathrm{b}}(\boldsymbol{x}) - \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{x}))$. Unlike the one-dimensional case, this setup has no simple closed-form solution, but requires standard iterative QP solving methods to solve for the optimal $\boldsymbol{\mu}$ vector (see e.g., active set methods [27, Ch. 16]). Thus, we opt for the interpolation approach, as it can be solved in closed-form, providing significant computational savings.

V. Simulation Results

To demonstrate the effectiveness of the proposed approach, we consider two simulation examples: the running double integrator example, and a nonlinear fixed-wing aircraft model[2].

A. Double Integrator

We consider the same setup as in Example 1, and compare the optimally interpolated controller (18) with the (bCBF-QP) from Section II-B and the blended controller (13) from Section III. Figure 2 demonstrates that our approach eliminates the undesired oscillations at the boundary of $\mathcal{C}_{\mathrm{S}}$. While the blended approach subjectively selects μ, our approach chooses the *optimal* value, which yields a safe controller with better performance. We note that for this very special case, the (bCBF-QP) and the (OI-QP) are equivalent. Because there is a single input, and the primary and backup controllers are state-independent signals on either extreme of the domain $\mathcal{U}$, the (OI-QP) can search over the entire domain of inputs to find a feasible, safe controller.

B. Fixed-Wing Aircraft

Next, consider a nonlinear fixed-wing aircraft model [28]:

$$\underbrace{\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\\ \dot{p}_{\mathrm{N}}\\ \dot{p}_{\mathrm{E}}\\ \dot{H}\\ \dot{P}\\ \dot{N}_{\mathrm{z}}\end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} P + \frac{N_{\mathrm{z}} g_{\mathrm{D}}}{V_{\mathrm{T}}}\sin\phi\tan\theta,\\ \frac{g_{\mathrm{D}}}{V_{\mathrm{T}}}\left(N_{\mathrm{z}}\cos\phi - \cos\theta\right)\\ \frac{N_{\mathrm{z}} g_{\mathrm{D}}\sin\phi}{V_{\mathrm{T}}\cos\theta}\\ V_{\mathrm{T}}\cos\theta\cos\psi\\ V_{\mathrm{T}}\cos\theta\sin\psi\\ V_{\mathrm{T}}\sin\theta\\ -\frac{1}{\tau_{\mathrm{p}}}P\\ -\frac{1}{\tau_{\mathrm{z}}}N_{\mathrm{z}}\end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ \frac{1}{\tau_{\mathrm{p}}} & 0\\ 0 & \frac{1}{\tau_{\mathrm{z}}}\end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix}u_1\\ u_2\end{bmatrix}}_{\boldsymbol{u}}, \tag{30}$$

which is derived by assuming that the angle of attack and the angle of side slip are zero, the load factor dynamics and roll mode are modeled as a first-order system, and that gravity and load factor are the only accelerations on the aircraft. In (30), the states ϕ, θ, ψ, p_{N}, p_{E}, H, and P, and N_{z} represent the roll, pitch, and yaw angles, the north and east position coordinates, altitude, roll rate, and normal load factor, respectively. The constants V_{T}, g_{D}, τ_{p} and τ_{z} are the true airspeed, gravitational acceleration, and the time constants for the roll mode and load factor dynamics, respectively. The bounded inputs are the commanded roll rate $u_1 \in [-\frac{\pi}{2}, \frac{\pi}{2}]\,\mathrm{rad/s}$ and commanded load factor $u_2 \in [-1, 4]$ (representing normal accelerations measured in g_{D} that pull the aircraft up).

[2]Code is available at: https://github.com/davidvwijk/OI-CBF

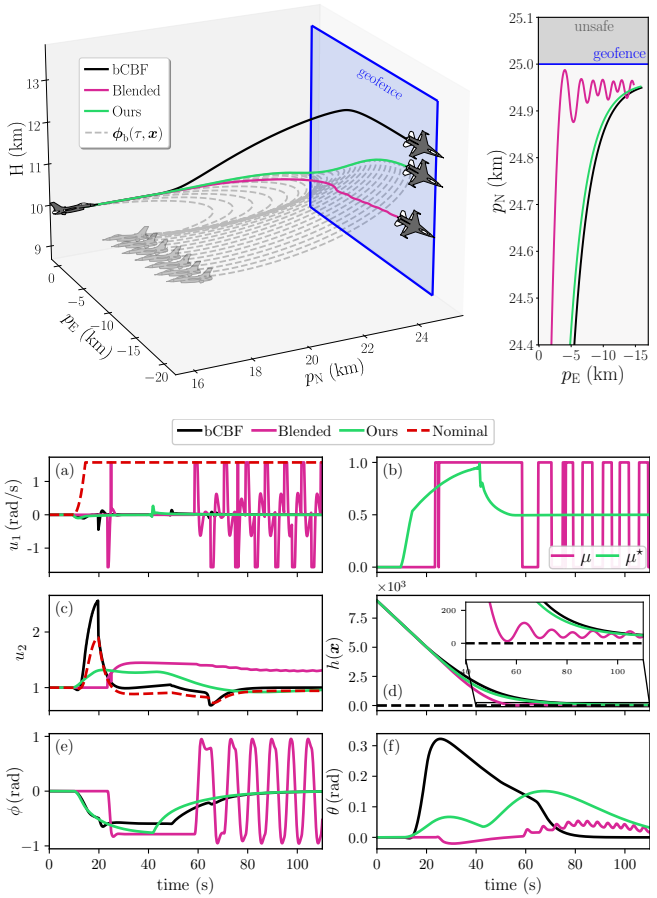

Fig. 3. Simulation of the fixed-wing aircraft (30) comparing the (bCBF-QP) (black), the function-based blended controller in (13) (pink) and the proposed (OI-QP) approach (green). The (bCBF-QP) and (OI-QP) ensure the safety of (30) by avoiding crossing the geofence (blue) and gliding near its surface (**top left**). The blended controller also ensures safety but experiences oscillations along the geofence (**top right**), causing hysteresis most notably in the roll rate command (**a**) as the blending parameter switches between 0 and 1 (**b**). This in turn produces stark oscillations in the aircraft's roll angle, which is highly undesirable for stable flight (**e**). In contrast, the (OI-QP) smoothly interpolates between $\boldsymbol{k}_\mathrm{b}$ and $\boldsymbol{k}_\mathrm{p}$ (**b**) to guarantee the safety of the aircraft (**d**), whilst avoiding evaluating a QP online as required by the (bCBF-QP). Further, the closed-form solution from Theorem 4 always yields a safe control signal which respects the input bounds, as per Lemma 3.

We consider a scenario where the aircraft must stay within a prespecified airspace, described by a geofence with a known location $\boldsymbol{p}_\mathrm{g} \in \mathbb{R}^2$ and normal vector $\boldsymbol{n}_\mathrm{g} \in \mathbb{R}^2$ with $\|\boldsymbol{n}_\mathrm{g}\| = 1$. The safe set can be written as

$$\mathcal{C}_\mathrm{S} \triangleq \{\boldsymbol{x} \in \mathcal{X} : h(\boldsymbol{x}) = \boldsymbol{n}_\mathrm{g} \cdot (\boldsymbol{p} - \boldsymbol{p}_\mathrm{g}) \geq 0\}, \tag{31}$$

where $\boldsymbol{p} = [p_\mathrm{N} \;\; p_\mathrm{E}]^\top$. The nominal controller seeks to push the aircraft into the geofence while the backup controller executes a coordinated turn with $\boldsymbol{k}_\mathrm{b}(\boldsymbol{x}) = [u_{\mathrm{b}1} \;\; u_{\mathrm{b}2}]^\top$ where

$$\begin{aligned} u_{\mathrm{b}1} &= \mathrm{sat}\big(P + \tau_\mathrm{p}\big(K_\phi(\phi^* - \phi) - K_\mathrm{p} P\big)\big), \\ u_{\mathrm{b}2} &= \mathrm{sat}\big(N_\mathrm{z} + \tau_\mathrm{z}\big(K_\mathrm{N}(N_\mathrm{z}^* - N_\mathrm{z}) + K_\mathrm{H}(H^* - H) - K_\theta \theta\big)\big), \end{aligned} \tag{32}$$

and $K_\phi, K_\mathrm{p}, K_\mathrm{N}, K_\mathrm{H}, K_\theta > 0$ are gains, H^* is the altitude, $\phi^* \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ is the bank angle, and $N_\mathrm{z}^* = 1/\cos(\phi^*)$ is the load factor in the turn. Note that $u_{\mathrm{b}1}$ and $u_{\mathrm{b}2}$ are saturated using a softplus function which retains the smoothness of f_cl. Such a controller drives the aircraft to a circular path and renders the following set forward invariant:

$$\mathcal{C}_\mathrm{B} \triangleq \left\{ \boldsymbol{x} \in \mathcal{X} \;\middle|\; \begin{array}{c} \phi = \phi^*,\ \theta = 0,\ H = H^*, \\ P = 0,\ N_\mathrm{z} = N_\mathrm{z}^*,\ h_6(\boldsymbol{x}) \geq 0 \end{array} \right\}, \tag{33}$$

with

$$h_6(\boldsymbol{x}) \triangleq h(\boldsymbol{x}) + \rho\big(\boldsymbol{n}_\mathrm{g} \cdot \boldsymbol{n}(\psi) - 1\big) - c_6, \;\; \boldsymbol{n}(\psi) \triangleq \begin{bmatrix} -\sin\psi \\ \cos\psi \end{bmatrix}.$$

Here, $\rho = V_\mathrm{T}^2/(g_\mathrm{D}\tan(\phi^*))$ is the turning radius under $\boldsymbol{k}_\mathrm{b}$ and $c_6 > 0$. This backup set describes the set of periodic orbits whose centers are at least $\rho + c_6$ away from the geofence, at a height of H^*.

Proposition 2. *The backup set $\mathcal{C}_\mathrm{B}$ in* (33) *is forward invariant for the system* (30) *with the backup controller* (32)*, and it satisfies $\mathcal{C}_\mathrm{B} \subseteq \mathcal{C}_\mathrm{S}$ for the safe set in* (31)*.*

Proof. For all $\boldsymbol{x} \in \mathcal{C}_\mathrm{B}$, the states ϕ, θ, H, P, and N_z are constant along the closed-loop backup dynamics under the backup control law (32). The backup flow of the yaw angle evolves with $\frac{\partial}{\partial \tau}\psi_\mathrm{b}(\tau, \boldsymbol{x}) = V_\mathrm{T}/\rho$ which has the solution $\psi_\mathrm{b}(\tau, \boldsymbol{x}) = (V_\mathrm{T}/\rho)\tau + \psi$. Similarly, the closed-form solution for the backup flow of the position of the aircraft is

$$\boldsymbol{p}_\mathrm{b}(\tau, \boldsymbol{x}) = \boldsymbol{p} + \rho\big(\boldsymbol{n}(\psi) - \boldsymbol{n}(\psi_\mathrm{b}(\tau, \boldsymbol{x}))\big). \tag{34}$$

Thus, using (34), it can be verified that $h_6(\boldsymbol{p}_\mathrm{b}(\tau, \boldsymbol{x})) \geq 0$ along the backup flow for all $\boldsymbol{x} \in \mathcal{C}_\mathrm{B}$ and $\tau \geq 0$:

$$\begin{aligned} h_6\big(\boldsymbol{p}_\mathrm{b}(\tau, \boldsymbol{x})\big) &= \boldsymbol{n}_\mathrm{g} \cdot \big(\boldsymbol{p}_\mathrm{b}(\tau, \boldsymbol{x}) - \boldsymbol{p}_\mathrm{g}\big) \\ &\quad + \rho\big(\boldsymbol{n}_\mathrm{g} \cdot \boldsymbol{n}(\psi_\mathrm{b}(\tau, \boldsymbol{x})) - 1\big) - c_6 \\ &= \boldsymbol{n}_\mathrm{g} \cdot (\boldsymbol{p} - \boldsymbol{p}_\mathrm{g}) + \rho\big(\boldsymbol{n}_\mathrm{g} \cdot \boldsymbol{n}(\psi) - 1\big) - c_6 \\ &= h_6(\boldsymbol{x}) \geq 0. \end{aligned}$$

Further, because $\rho\big(\boldsymbol{n}_\mathrm{g} \cdot \boldsymbol{n}(\psi) - 1\big) \leq 0$ and $-c_6 \leq 0$, we have $h(\boldsymbol{x}) \geq h_6(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$, and thus $\mathcal{C}_\mathrm{B} \subseteq \mathcal{C}_\mathrm{S}$. ∎

In practice, a neighborhood of $\mathcal{C}_\mathrm{B}$ may be used as a backup set. Such a set can be constructed via $h_1(\boldsymbol{x}) = c_1^2 - (\phi^* - \phi)^2$, $h_2(\boldsymbol{x}) = c_2^2 - \theta^2$, $h_3(\boldsymbol{x}) = c_3^2 - (H^* - H)^2$, $h_4(\boldsymbol{x}) = c_4^2 - P^2$, $h_5(\boldsymbol{x}) = c_5^2 - (N_\mathrm{z}^* - N_\mathrm{z})^2$ with constants $c_1, c_2, c_3, c_4, c_5 > 0$. Namely, one may use a smooth under-approximation of the minimum of these functions as in [29], [30]:

$$\widehat{\mathcal{C}}_\mathrm{B} \triangleq \left\{ \boldsymbol{x} \in \mathcal{X} : h_\mathrm{b}(\boldsymbol{x}) = -\frac{1}{\kappa} \ln\left(\sum_{i \in I} \mathrm{e}^{-\kappa h_i(\boldsymbol{x})}\right) \geq 0 \right\},$$

with smoothing parameter $\kappa > 0$ and $I = \{1, \ldots, 6\}$.

The nominal controller uses (32) but with a desired roll angle $\phi^* = \mathrm{sat}(K_\psi(\psi^* - \psi)V_\mathrm{T}/g_\mathrm{D})$ and yaw angle $\psi^* = \arctan((p_\mathrm{E}^* - p_\mathrm{E})/(p_\mathrm{N}^* - p_\mathrm{N})) \in \mathbb{S}^1$, where $K_\psi > 0$ and p_E^*, p_N^* is a desired setpoint which is purposefully placed far behind the geofence.

Figure 3 shows the simulation results for system (30) comparing the methods from Sections II-B and III with the

proposed (OI-QP) controller[3]. The (OI-QP) guarantees the safety of the aircraft by smoothly selecting the parameter $\mu^\star(\boldsymbol{x})$ which minimizes the utilization of $\boldsymbol{k}_\mathrm{b}$, whilst ensuring that (30) stays within the controlled invariant set $\mathcal{C}_\mathrm{I}$ defined in (7). Thanks to Theorem 3 and Theorem 4, this controller satisfies the safety constraint, and the optimal blending parameter $\mu^\star(\boldsymbol{x})$ is obtained in closed-form. Further, this parameter ensures that the safe controller always satisfies input constraints, as can be seen in Figure 3. While the (bCBF-QP) ensures the safety of (30) under input bounds, it requires solving a QP with a very large number of constraints online, and forward integrating a large number of ODEs (i.e., $n^2+n=72$ ODEs, while the optimal interpolation uses $3n=24$ ODEs). The blended approach (13)-(15) also obeys the geofence constraint and the input bounds, but experiences hysteresis and subsequent oscillations near the geofence.

VI. Conclusion

We presented a novel framework to guarantee the safety of nonlinear control-affine systems by optimally interpolating between a verified backup controller and a nominal controller. We derived a closed-form solution for the resulting safe controller, and proved that this controller respects input bounds while guaranteeing safety. Through simulations, including a double integrator and a geofencing scenario on a fixed-wing aircraft, we demonstrated the overall performance improvement of our approach over the existing function-based controller blending method. This advancement simplifies the deployment of bCBF-based safety filters, providing feasibility guarantees in computationally limited systems. Future work will explore OI-QP implementations on computationally-constrained hardware platforms such as a fixed-wing aircraft.

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[2] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, 2023.

[3] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[4] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[5] K. Garg, J. Usevitch, J. Breeden, M. Black, D. Agrawal, H. Parwana, and D. Panagou, "Advances in the theory of control barrier functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems," *Annual Reviews in Control*, vol. 57, p. 100945, 2024.

[6] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An Online Approach to Active Set Invariance," in *57th IEEE Conference on Decision and Control*, pp. 3592–3599, 2018.

[7] K. Dunlap, M. Hibbard, M. Mote, and K. Hobbs, "Comparing run time assurance approaches for safe spacecraft docking," *IEEE Control Systems Letters*, vol. 6, pp. 1849–1854, 2022.

[3]The simulation uses $g_\mathrm{D}=9.81\,\mathrm{m/s^2}$, $V_\mathrm{T}=200\,\mathrm{m/s}$, $\tau_\mathrm{p}=1\,\mathrm{s}$, $\tau_\mathrm{z}=1\,\mathrm{s}$, $\phi^*=-\frac{\pi}{4}\,\mathrm{rad}$, $H^*=10\,\mathrm{km}$.

[8] D. E. J. van Wijk, S. Coogan, T. G. Molnar, M. Majji, and K. L. Hobbs, "Disturbance-robust backup control barrier functions: Safety under uncertain dynamics," *IEEE Control Systems Letters*, vol. 8, pp. 2817–2822, 2024.

[9] K. L. Hobbs, M. L. Mote, M. C. Abate, S. D. Coogan, and E. M. Feron, "Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems," *IEEE Control Systems Magazine*, vol. 43, no. 2, pp. 28–65, 2023.

[10] T. Gurriet, *Applied Safety Critical Control*. PhD thesis, California Institute of Technology, 2020.

[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 Transactions on Mechatronics*, pp. 1–12, 2024.

[12] W. Zeng, L. Li, H. Xiong, H. Jiang, B. R. Noack, W. Lu, and H. Liu, "Onboard operational safety filter for a quadrotor in an environment with dynamic obstacles," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 5, pp. 3776–3785, 2025.

[13] 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 *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11676–11682, IEEE, 2024.

[14] P. Rabiee and J. B. Hoagg, "Soft-minimum and soft-maximum barrier functions for safety with actuation constraints," *Automatica*, vol. 171, p. 111921, 2025.

[15] P. M. Rivera, "Forward and control invariance analysis of backup control barrier function induced safe sets for online safety of nonlinear systems," in *63rd Conference on Decision and Control*, pp. 8150–8157, 2024.

[16] E. Ç. Altunkaya and İ. Özkol, "Nuisance-free ground collision avoidance design with adaptive exponential control barrier functions," *Journal of Guidance, Control, and Dynamics*, pp. 1–15, 2025.

[17] E. Squires, P. Pierpaoli, R. Konda, S. Coogan, and M. Egerstedt, "Composition of safety constraints for fixed-wing collision avoidance amidst limited communications," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 4, pp. 714–725, 2022.

[18] Z. Zheng, J. Li, Z. Guan, and Z. Zuo, "Constrained moving path following control for UAV with robust control barrier function," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 7, pp. 1557–1570, 2023.

[19] J. Breeden and D. Panagou, "Quadratic programs for high relative degree spatial constraints and spatiotemporal specifications with spacecraft applications," in *59th IEEE Conference on Decision and Control*, pp. 1496–1502, 2020.

[20] T. G. Molnar, S. K. Kannan, J. Cunningham, K. Dunlap, K. L. Hobbs, and A. D. Ames, "Collision avoidance and geofencing for fixed-wing aircraft with control barrier functions," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 5, pp. 1493–1508, 2025.

[21] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2897–2904, 2022.

[22] A. Singletary, T. Gurriet, P. Nilsson, and A. D. Ames, "Safety-critical rapid aerial exploration of unknown environments," in *IEEE International Conference on Robotics and Automation*, pp. 10270–10276, 2020.

[23] M. H. Cohen, P. Ong, G. Bahati, and A. D. Ames, "Characterizing smooth safety filters via the implicit function theorem," *IEEE Control Systems Letters*, vol. 7, pp. 3890–3895, 2023.

[24] 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.

[25] T. G. Molnar and A. D. Ames, "Safety-critical control with bounded inputs via reduced order models," in *American Control Conference*, pp. 1414–1421, 2023.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[27] J. Nocedal and S. Wright, "Numerical optimization," *Springer Series in Operations Research and Financial Engineering*, pp. 1–664, 2006.

[28] S. Stephens, *A Real-time Algorithm to Achieve Precise Coordinated Arrival Times in a Time-variant Environment*. PhD thesis, Air Force Institute of Technology, 2021.

[29] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2019.

[30] T. G. Molnar and A. D. Ames, "Composing control barrier functions for complex safety specifications," *IEEE Control Systems Letters*, vol. 7, pp. 3615–3620, 2023.