

# Safety on the Fly: Constructing Robust Safety Filters via Policy Control Barrier Functions at Runtime

Luzia Knoedler<sup>1\*</sup>, Oswin So<sup>2\*</sup>, Ji Yin<sup>3</sup>, Mitchell Black<sup>4</sup>,  
Zachary Serlin<sup>4</sup>, Panagiotis Tsiotras<sup>3</sup>, Javier Alonso-Mora<sup>1</sup>, and Chuchu Fan<sup>2</sup>

**Abstract**—Control Barrier Functions (CBFs) have proven to be an effective tool for performing safe control synthesis for nonlinear systems. However, guaranteeing safety in the presence of disturbances and input constraints for high relative degree systems is a difficult problem. In this work, we propose the Robust Policy CBF (RPCBF), a practical approach for constructing robust CBF approximations online via the estimation of a value function. We establish conditions under which the approximation qualifies as a valid CBF and demonstrate the effectiveness of the RPCBF-safety filter in simulation on a variety of high relative degree input-constrained systems. Finally, we demonstrate the benefits of our method in compensating for model errors on a hardware quadcopter platform by treating the model errors as disturbances. Website including code: [www.oswinso.xyz/rpcbf/](http://www.oswinso.xyz/rpcbf/)

**Index Terms**—Robot Safety, Optimization and Optimal Control, Collision Avoidance

## I. INTRODUCTION AND RELATED WORKS

**I**N the realm of autonomous systems, providing safety guarantees is crucial, especially in critical applications such as autonomous driving and healthcare robotics. Control Barrier Functions (CBFs) [1], [2] have proven to be an effective tool to maintain and certify the safety of dynamical systems. In particular, they can be applied as a Safety Filter (SF) that minimally modifies arbitrary control inputs to ensure safety, making them especially valuable when integrated with learning-based controllers.

Despite their theoretical advantages, significant challenges remain in the practical application and construction of CBFs. First, constructing CBFs is non-trivial, specifically for high

Manuscript received: March, 11, 2025; Revised June, 5, 2025; Accepted July, 14, 2025.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.

Luzia Knoedler and Javier Alonso-Mora are supported by the Office of Naval Research Global, grant N62909-25-1-2027, project SECURE.

\*Both authors contributed equally to this work.

<sup>1</sup>Luzia Knoedler and Javier Alonso-Mora are with the Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands. {l.knoedler, j.alonsomora}@tudelft.nl

<sup>2</sup>Oswin So and Chuchu Fan are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA. {oswinso, chuchu}@mit.edu

<sup>3</sup>Ji Yin and Panagiotis Tsiotras are with the D. Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. {jyin81, tsiotras}@gatech.edu

<sup>4</sup>Mitchell Black and Zachary Serlin are with the MIT Lincoln Laboratory, Cambridge, MA, USA. {mitchell.black, zachary.serlin}@ll.mit.edu

Digital Object Identifier (DOI): see top of this page.

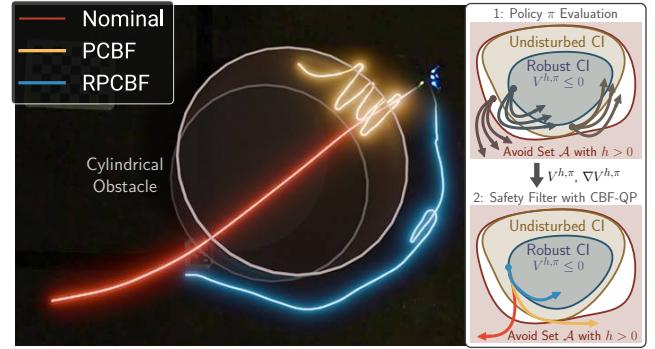


Fig. 1: We propose the Robust Policy Control Barrier Function (RPCBF), which approximates the robust value function  $V^{h,\pi}$  of a system under bounded disturbances for policy  $\pi$  at runtime. The zero sublevel set of  $V^{h,\pi}$  is a robust controlled-invariant (CI) set. We apply a RPCBF-Safety Filter (SF) to ensure safety for any **unsafe nominal policy**, demonstrating superior performance over the non-robust PCBF-SF on a quadcopter with model errors treated as disturbances.

relative degree systems with input constraints. Second, the safety guarantees of CBF-based controllers depend on having an accurate system model, which is rarely the case for systems in real life. This makes the safety guarantees of such controllers sensitive to model uncertainties.

**Learning Control Barrier Functions.** To minimize reliance on extensive domain knowledge, a recent trend is to learn neural CBFs that approximate CBFs using Neural Networks (NNs) [3], [4], [5], [6], [7], [8], [9]. Neural CBFs have been successfully applied to high-dimensional systems, including multi-agent control scenarios [7], [10], and have been extended to handle parametric uncertainties [11] and obstacles with unknown dynamics [12]. Although using NNs as CBFs offers universal approximation capabilities, it requires certifying them as valid CBFs to ensure safety guarantees and limits their interpretability. Furthermore, using a naive approach to learning neural CBFs by minimizing a loss that encourages the CBF conditions can lead to a small or even empty forward-invariant set. Thus, [5] presents a method to construct CBFs using policy evaluation of *any* policy. They show that the policy value function is a CBF and learn an NN approximation. In this setting, the policy value function represents the maximum-over-time constraint violation, indicating how suitable a state is for a system following a specific policy. However, their approach does not consider uncertainties in the system dynamics.

**Robust Safety.** Controllers that are robust to disturbances are essential for ensuring the safety of autonomous systems in the real world. This has been studied before in *robust* CBFs [11], [13], [14], which guarantee safety under bounded disturbances. However, constructing robust CBFs is inherently

more difficult than constructing standard CBFs, especially under input constraints. Hamilton-Jacobi reachability analysis [15] can be used to compute robust control-invariant sets, which can then be subsequently used for constructing robust CBFs [16], [17], [18], [19]. However, reachability analysis in itself is challenging, with grid-based partial Differential Equation (DE) solvers being limited to state dimensions below five [20], while deep learning-based solvers [21], [22], [23] require subsequent NN verification to check for solution accuracy. Moreover, both learning-based CBF approaches and deep learning-based reachability solvers depend on predefined system dynamics and disturbance assumptions, which are difficult to adapt, limiting their flexibility once deployed, as retraining cannot be performed on the system. Thus, [24] train a value network with the avoidance set and disturbance bounds as inputs, however this increases training data requirements and complicates evaluating how well the learned network represents the true value function.

As an alternative to robust safety, other works focus on risk-aware safety, which aims to ensure safety with high probability by modeling disturbances probabilistically and incorporating risk measures [25], [26]. Unlike robust methods, which ensure constraint satisfaction for all disturbances within a bounded set but may be overly conservative, risk-aware approaches typically rely on knowledge of the disturbance's probability distribution. In this work, we focus on robust safety.

We propose a practical approach for constructing a CBF approximation at runtime, which can be derived for any system dynamics and disturbance bounds without requiring (re)training. We establish conditions under which the resulting CBF approximation qualifies as a valid CBF. Our method constructs CBFs by evaluating the value function of *any* policy, which has been shown to be a valid CBF in [5]. By leveraging finite-horizon policy rollouts, we enable a more detailed analysis of safety guarantees than NN approximations. We apply this approach to construct approximations of robust CBFs.

**Contributions.** We summarize our contributions as follows.

- 1) We propose a method of constructing (robust) CBFs using the (robust) policy value function and a real-time approximation that can be used at runtime.
- 2) We demonstrate real-time performance and the benefits of our robust CBFs on a hardware quadcopter, where robustness to model errors is key for collision prevention.

## II. PRELIMINARIES

### A. Problem Statement

We consider a *disturbed* continuous-time, control-affine dynamical system of the form

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

with state  $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^n$ , control input  $\mathbf{u}_t \in \mathcal{U} \subseteq \mathbb{R}^m$  and unknown, bounded, smooth disturbance  $\mathbf{d}_{\min} \leq \mathbf{d}_t \leq \mathbf{d}_{\max}$  with  $\mathbf{d}_{\min}, \mathbf{d}_{\max} \in \mathbb{R}^d$  (e.g., estimated from empirical data), where  $\mathbf{d}_t$  can be time-varying. The functions  $f$  and  $g$  are assumed to be locally Lipschitz continuous. Let  $\mathcal{A} \subset \mathcal{X}$  denote the set of states to be avoided. This paper addresses the following SF synthesis problem:

**Problem 1** (Safety Filter Synthesis). *Given a system (1) and an avoid set  $\mathcal{A} \subset \mathcal{X}$ , find a control policy  $\pi_{\text{filt}} : \mathcal{X} \rightarrow \mathcal{U}$  that keeps the system state outside  $\mathcal{A}$  while staying close to a performant, possibly unsafe nominal policy  $\pi_{\text{nom}} : \mathcal{X} \rightarrow \mathcal{U}$ :*

$$\begin{aligned} & \min_{\pi_{\text{filt}}} \|\pi_{\text{filt}} - \pi_{\text{nom}}\| \\ & \text{s.t. } \dot{\mathbf{x}}_t = f(\mathbf{x}_t, \mathbf{d}_t) + g(\mathbf{x}_t, \mathbf{d}_t)\pi_{\text{filt}}(\mathbf{x}_t) \\ & \quad \mathbf{x}_t \notin \mathcal{A}, \forall t \geq 0, \end{aligned} \quad (2)$$

where  $\|\cdot\|$  is some distance metric.

We focus on solving Prob. 1 using (zeroing) CBFs [27].

### B. Safety Filters using Control Barrier Functions

We begin by providing a standard definition of a CBF in the non-robust case, which we extend to the robust case for Policy Control Barrier Functions (PCBFs) in the next section. Define the *undisturbed* system to be a particular case of the disturbed system (1) without disturbances ( $\mathbf{d} = 0$ ), by

$$\dot{\mathbf{x}}_t = f(\mathbf{x}_t, 0) + g(\mathbf{x}_t, 0)\mathbf{u}_t. \quad (3)$$

Let  $B : \mathcal{X} \rightarrow \mathbb{R}$  be a continuously differentiable function, with  $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} \mid B(\mathbf{x}) \leq 0\}$  as its 0-sublevel set. Let  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  be an extended class- $\kappa_\infty$  function<sup>1</sup>. Then,  $B$  is a CBF for the undisturbed system (3) on  $\mathcal{X}$  [1] if

$$B(\mathbf{x}) > 0, \forall \mathbf{x} \in \mathcal{A}, \quad (4a)$$

$$B(\mathbf{x}) \leq 0 \Rightarrow \inf_{\mathbf{u} \in \mathcal{U}} L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} \leq -\alpha(B(\mathbf{x})), \quad (4b)$$

with  $L_f B := \nabla B^\top f$  and  $L_g B := \nabla B^\top g$ . It then follows that any control input  $\mathbf{u} \in K_{\text{cbf}}$  with

$$K_{\text{cbf}}(\mathbf{x}) = \{\mathbf{u} \in \mathcal{U} \mid L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} + \alpha(B(\mathbf{x})) \leq 0\}$$

renders  $\mathcal{C}$  forward-invariant [1]. In other words, there exists an  $\mathbf{u} \in \mathcal{U}$  such that any trajectory starting within  $\mathcal{C}$  remains in  $\mathcal{C}$ . Asymptotic stability of  $\mathcal{C}$  can be achieved by extending (4b) to hold for all  $\mathbf{x} \in \mathcal{X}$  [27]. Since the right hand side of (4b) is linear in  $\mathbf{u}$ , given a CBF  $B$ , we can solve Prob. 1 for (3) using the following Quadratic Program (QP)-based controller:

$$\begin{aligned} \mathbf{u}_{\text{CBF-QP}} &= \arg \min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - \pi_{\text{nom}}(\mathbf{x})\|^2 \\ & \text{s.t. } L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} \leq -\alpha(B(\mathbf{x})). \end{aligned} \quad (\text{CBF-QP})$$

While CBFs can be applied to guarantee safety for a known undisturbed system, three major challenges remain:

- 1) How do we synthesize a valid CBF that satisfies (4b) for high relative degree systems with input constraints?
- 2) How do we synthesize a robust CBF that ensures safe control for the disturbed system?
- 3) How can we efficiently derive a CBF at runtime for different system dynamics and disturbance assumptions?

<sup>1</sup>Extended class- $\kappa_\infty$  is the set of continuous, strictly increasing functions  $\alpha : (-\infty, \infty) \rightarrow (-\infty, \infty)$  with  $\alpha(0) = 0$ .

### III. (ROBUST) POLICY CONTROL BARRIER FUNCTIONS

To address the above challenges, we leverage the insight from [5] that CBFs can be constructed by deriving the policy value function through the evaluation of *any* policy. Rather than approximating the policy value function with an NN as in [5], we propose a real-time approximation that avoids NNs and can be derived at runtime through a finite-horizon numerical approximation. We further extend this approach to the robust case and introduce Robust Policy Control Barrier Functions (RPCBFs) and subsequently propose a sampling-based approximation that can be derived at runtime. Next, we revisit the formulation of PCBFs and describe our extensions and approximations.

#### A. Constructing CBFs via Policy Evaluation

Based on [5], we first derive the PCBF formulation for the undisturbed system in (3). Assume that the avoid set  $\mathcal{A}$  can be described as the super-level set of a function  $h : \mathcal{X} \rightarrow \mathbb{R}$  (e.g., the negative distance to the constraint):

$$\mathcal{A} = \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) > 0\}. \quad (5)$$

Note that  $h(\mathbf{x}) > 0$  for states that are already in the failure set, whereas  $B(\mathbf{x}) > 0$  for states from which failure is inevitable in the future under the given dynamics and input constraints. In the absence of input constraints,  $h$  and  $B$  may coincide. We denote by  $\mathbf{x}_t^\pi$  the resulting state at time  $t$  when starting from the initial state  $\mathbf{x}_0$  and following policy  $\pi : \mathcal{X} \rightarrow \mathcal{U}$ . Furthermore, we define the *maximum-over-time* value function for the undisturbed system in (3) as

$$V_\infty^{h,\pi}(\mathbf{x}_0) := \sup_{t \geq 0} h(\mathbf{x}_t^\pi). \quad (6)$$

As stated in [5, Theorem 1], the *policy value function*  $V_\infty^{h,\pi}$  is a CBF for the undisturbed system in (3) for any  $\pi$ , since  $V_\infty^{h,\pi}$  satisfies the following two inequalities  $\forall \mathbf{x} \in \mathcal{X}$

$$V_\infty^{h,\pi}(\mathbf{x}) \geq h(\mathbf{x}), \quad (7)$$

$$\nabla V_\infty^{h,\pi}(\mathbf{x})^T (f(\mathbf{x}) + g(\mathbf{x})\pi(\mathbf{x})) \leq 0, \quad (8)$$

which imply (4a) and (4b). For details, we refer to [5]. The key intuition here is that  $V_\infty^{h,\pi}$  provides an upper bound on the worst future constraint violation  $h$  under the optimal policy since the optimal policy will do no worse than  $\pi$ . Thus, CBFs can be constructed via policy evaluation of any policy. We refer to  $\pi$  as the design policy, noting that the nominal policy  $\pi_{\text{nom}}$  differs from the design policy.

#### B. Finite Horizon Approximation of PCBFs

A key challenge with the policy value function  $V_\infty^{h,\pi}$  is that its definition requires an *infinite-horizon*. While [5] tackles this problem by using an NN to learn  $V_\infty^{h,\pi}$  with a loss derived using dynamic programming, we take a different approach and perform a *finite-horizon* approximation that can be computed *without* the use of an NN, enabling a more in-depth analysis of the resulting safety guarantees. Expanding  $V_\infty^{h,\pi}$ :

$$V_\infty^{h,\pi}(\mathbf{x}_0) = \max \left\{ \sup_{0 \leq t < T} h(\mathbf{x}_t^\pi), V_\infty^{h,\pi}(\mathbf{x}_T^\pi) \right\} \quad (9)$$

$$\approx \sup_{0 \leq t < T} h(\mathbf{x}_t^\pi) := V_T^{h,\pi}(\mathbf{x}_0), \quad (10)$$

where the approximation is made by dropping the  $V_\infty^{h,\pi}(\mathbf{x}_T^\pi)$  “tail”. The question is then whether the finite-horizon approximation  $V_T^{h,\pi}$  is a CBF and can provide safety guarantees.

We can at least answer this in the affirmative when the approximation in (10) is an equality, i.e., the maximum occurs in  $[0, T]$ . We state this formally in the following theorem.

**Theorem 1.** Suppose that for all  $\mathbf{x}_0 \in \mathcal{X}$ ,

$$V_T^{h,\pi}(\mathbf{x}_0) \leq 0 \implies \sup_{0 \leq t < T} h(\mathbf{x}_t^\pi) > V_\infty^{h,\pi}(\mathbf{x}_T^\pi). \quad (11)$$

Then,  $V_T^{h,\pi}$  is a CBF.

*Proof.* Since  $V_T^{h,\pi}(\mathbf{x}) \geq h(\mathbf{x})$  by definition, (4a) is satisfied by  $V_T^{h,\pi}$ . Moreover, by (11),  $V_T^{h,\pi}(\mathbf{x}_0) = V_\infty^{h,\pi}(\mathbf{x}_0)$  when  $V_T^{h,\pi}(\mathbf{x}_0) \leq 0$ . Hence, since  $V_\infty^{h,\pi}$  is a CBF, (4b) holds for  $V_\infty^{h,\pi}$ , and thus also holds for  $V_T^{h,\pi}$ . Thus,  $V_T^{h,\pi}$  is a CBF.  $\square$

This enables us to prove the following corollary.

**Corollary 1.** Suppose there exists a  $\tilde{T} < \inf$  such that

$$\arg \max_{t \geq 0} h(\mathbf{x}_t^\pi) < \tilde{T}, \quad \forall \mathbf{x}_0 \text{ where } V_T^{h,\pi}(\mathbf{x}_0) \leq 0. \quad (12)$$

Then,  $V_T^{h,\pi}$  is a CBF for  $T \geq \tilde{T}$ .

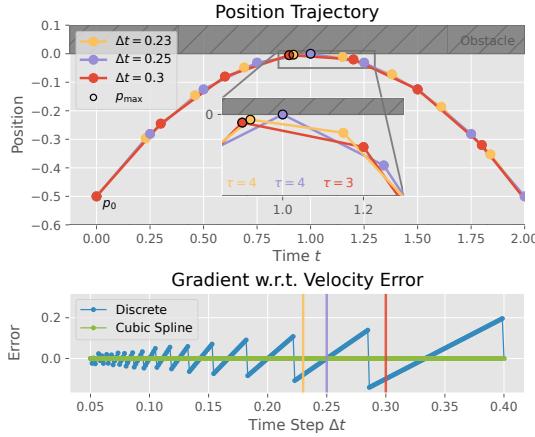
*Proof.* (12) implies (11) for  $T \geq \tilde{T}$ . Proof from Thm. 1.  $\square$

The value of  $\tilde{T}$  depends on the chosen design policy  $\pi$ . Any policy can be selected, but a conservative Controlled-Invariant (CI) set may result. If  $\pi$  is chosen as a controller that steers the system towards a safe, steady-state or CI set within a finite horizon  $\tilde{T}$ , Corollary 1 holds, thus  $V_T^{h,\pi}$  is a valid CBF.

If  $V_T^{h,\pi}(\mathbf{x}_0) = 0$ , then applying the design policy  $\pi$  exactly guarantees that the system will remain safe for at least the time horizon  $0 \leq t < T$ , even if  $V_T^{h,\pi}(\mathbf{x}_0) < V_\infty^{h,\pi}(\mathbf{x}_0)$ .

**Remark** (Connections to Backup Controller / CBFs). Since the zero sublevel set of  $V_\infty^{h,\pi}$  is a CI set under  $\pi$ , (9) can also be seen as Backup CBF [28], [29] with backup controller  $\pi$  and no known CI terminal set. Unlike this (and other similar approaches [30]), our approach replaces the need for a known CI set with the requirement of a sufficiently long horizon  $T$ . Thus, the design policy  $\pi$  can be chosen arbitrarily and is not required to steer the system into a CI set. Furthermore, we demonstrate that the naive approximation of  $h$  over a time-discretized state trajectory introduces gradient errors. To address this, we present an improved time-discretization using cubic splines in Sec. III-C.

**Remark** (Connections to Model Predictive Control (MPC)). The finite-horizon approximation here is closely related to the use of MPC by practitioners. More precisely, although a terminal constraint set is often required to theoretically guarantee recursive feasibility of finite-horizon MPC [31], [32], practitioners often apply MPC without the use of such a terminal constraint set to wide success [33], [34], [35], [36]. Our decision to drop the  $V_\infty^{h,\pi}(\mathbf{x}_T^\pi)$  term can be viewed as being similar to dropping the terminal constraint set. Another similarity is the choice of horizon  $T$ . Namely, recursive feasibility holds in MPC given a sufficiently large horizon [37], similar to Corollary 1. However, the MPC horizon length is limited,



**Fig. 2: Value Function Gradient Error for Discrete-Time Double Integrator.** We highlight the discretized trajectory (top) and corresponding gradient error (bottom) for three different choices of  $\Delta t$  (yellow, purple, red). The gradient of the naive discrete-time approximation has large errors and varies with the choice of  $\Delta t$ . Taking the maximum of the cubic spline leads to much smaller errors.

as it requires solving a potentially nonlinear and non-convex optimization problem online, with computational complexity typically scaling cubically with  $T$  [38]. A key advantage of PCBF-SFs is that they only solve the simpler (CBF-QP), whose computation time is unaffected by  $T$ , see Section IV-D.

**Remark** (Connections to Predictive Safety Filter (PSF) [39]). The finite-horizon approximation is closely related to the PSF, which implicitly represents the safe set via a finite-horizon MPC problem with terminal constraints or long horizons for recursive feasibility. Unlike PCBF-SFs, the PSF requires solving a potentially nonlinear and nonconvex optimization problem online with complexity scaling cubically in  $T$  [38]. PCBF-SFs only require solving the simpler (CBF-QP). However, while the PSF may find a locally optimal solution, the conservativeness of the PCBF-SF depends on  $\pi$ .

### C. Time Discretization of Policy Control Barrier Functions

Another challenge lies in how to compute the maximum in (10). The states  $\mathbf{x}_t^\pi$  can be solved numerically using an ordinary DE solver, resulting in a time-discretized state trajectory. It is tempting to then consider taking the maximum  $h$  over this trajectory, i.e., for time discretization  $\Delta t$ ,

$$V_T^{h,\pi}(\mathbf{x}_0) \approx \max_{0 \leq k < H} h(\mathbf{x}_{k\Delta t}^\pi). \quad (13)$$

However, the gap between (10) and (13) is particularly disastrous when computing the gradient. We illustrate this in the following example for the Double Integrator (DI).

**Example: Gradient Error on the Double Integrator.** Consider a DI with positive velocity  $v_0 > 0$  decelerating with  $\pi(\mathbf{x}) = a = -1$ . The dynamics are defined by  $\dot{v} = v$ ,  $\dot{v} = a$  with initial state  $\mathbf{x}_0 = [p_0, v_0]$  and constraints  $h(\mathbf{x}) = p \leq 0$ . For the continuous-time case, the gradient can be derived as

$$\nabla V_\infty^{h,\pi}(\mathbf{x}_0) = \frac{\partial p_{\max}}{\partial \mathbf{x}_0} = [1, v_0]. \quad (14)$$

After (exact) time discretization with timestep  $\Delta t$ , the time-discretized states can be computed as

$$p_k = p_0 + v_0 k \Delta t + 0.5 a (k \Delta t)^2, \quad (15a)$$

$$v_k = v_0 + a k \Delta t. \quad (15b)$$

We now show that the gradient of  $V_\infty^{h,\pi}$  depends on  $\Delta t$  and denote by  $\nabla V_{\infty, \Delta t}^{h,\pi}$  the resulting gradient. Let  $\tau$  be the integer time step  $k$  at which the maximum position is reached. The maximum position is then given by

$$V_{\infty, \Delta t}^{h,\pi} = p_{\max} = p_0 + v_0 \tau \Delta t + 0.5 a (\tau \Delta t)^2, \quad (16)$$

with  $\frac{\partial p_{\max}}{\partial v_0} = \tau \Delta t$ , which is a function of  $\tau$ . Although  $\tau$  also depends on  $v_0$ , it is piecewise constant and has zero derivative since it only takes integer values. Comparing the gradients of  $V_\infty^{h,\pi}$  with  $V_{\infty, \Delta t}^{h,\pi}$  in Fig. 2, we see a large error between the two with discontinuities in the discrete-time gradient in  $\Delta t$ . This is particularly problematic when the gradient is used in a gradient-based optimization algorithm such as (CBF-QP).

### Improved Time-Discretization using Cubic Splines.

To reduce the error in the time-discretized value function approximation (13), we propose to approximate  $h(\mathbf{x}_t^\pi)$  by fitting a cubic spline to the points  $\{h(\mathbf{x}_{k\Delta t}^\pi)\}_{k=0}^{H-1}$ . The max over the cubic spline can then be computed in closed-form by solving the roots of a quadratic to yield a better approximation of  $\sup_{0 \leq t < T} h(\mathbf{x}_t^\pi)$  than the naive maximization (13). Intuitively, this resolves the gradient error due to integer-valued  $\tau$  from the previous example because the maximum of the cubic spline can now happen *between* timesteps. We can also formally quantify the error in both the cubic spline value and its gradient. Let  $\tilde{h} : [0, \infty) \rightarrow \mathbb{R}$  denote the cubic spline approximation of  $h$  as a function of time, and let  $\tilde{V}_{\infty, \Delta t}^{h,\pi}(\mathbf{x}_0) := \sup_{t \geq 0} \tilde{h}(t)$ . Using [40, Chapter 5], we obtain the error bounds

$$\left\| V_\infty^{h,\pi} - \tilde{V}_{\infty, \Delta t}^{h,\pi} \right\| \leq \frac{1}{16} \Delta t^4 \max_{t \geq 0} \left\| \frac{d^4}{dt^4} h(\mathbf{x}_t) \right\|, \quad (17a)$$

$$\left\| \nabla V_\infty^{h,\pi} - \nabla \tilde{V}_{\infty, \Delta t}^{h,\pi} \right\| \leq \frac{1}{24} \Delta t^3 \max_{t \geq 0} \left\| \frac{d^4}{dt^4} h(\mathbf{x}_t) \right\|. \quad (17b)$$

In the previous example of the DI, since  $h$  is exactly quadratic, applying cubic splines results in zero gradient error (Fig. 2). If  $d^4/dt^4 h(\mathbf{x}_t)$  can be bounded, the bounds (17) can then be used to suitably modify (CBF-QP) to guarantee safety. A larger  $\Delta t$  will therefore result in a more conservative SF.

### D. Robust Extension of PCBFs

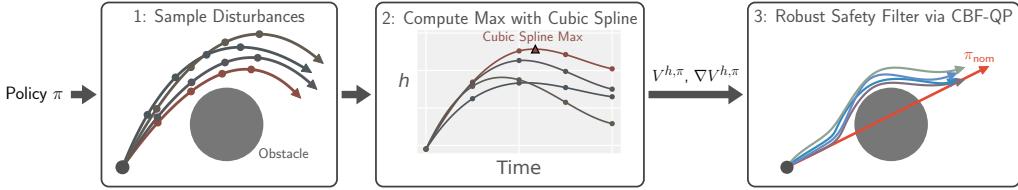
We extend PCBF to handle disturbances by defining the robust value function equivalent of (6) as

$$V_\infty^{h,\pi}(\mathbf{x}_0) := \sup_{t \geq 0} \sup_{\mathbf{d}(\cdot)} h(\mathbf{x}_t^\pi), \quad (18)$$

it can be shown with similar proof [5], [41] that  $V_\infty^{h,\pi}$  is a robust CBF [42], i.e., it satisfies (4a) and, for  $B(\mathbf{x}) \leq 0$ ,

$$\sup_{\mathbf{d} \in \mathcal{D}} \inf_{\mathbf{u} \in \mathcal{U}} \nabla B^\top (f(\mathbf{x}, \mathbf{d}) + g(\mathbf{x}, \mathbf{d}) \mathbf{u}) \leq -\alpha(B(\mathbf{x})). \quad (19)$$

Solving for robust controls that satisfy (19) renders the zero sublevel set *robust* forward-invariant [13]. However, deriving the worst-case disturbance is generally intractable because it



**Fig. 3: Summary of RPCBF Algorithm.** Given a policy  $\pi$ , we sample disturbance trajectories, then compute the maximum  $h$  with cubic splines to obtain  $V^{h,\pi}$  and  $\nabla V^{h,\pi}$  (using automatic differentiation). This is used in (CBF-QP) to obtain a robust SF.

### Algorithm 1 Robust Policy CBF (RPCBF)

```

1: Input: Initial State  $\mathbf{x}_0$ , Policy  $\pi$ , Constraint function  $h$ , Horizon  $T = H\Delta t$ , Number of disturbance samples  $N$ 
2: for  $i = 1 : N$  do
3:   Sample disturbance trajectory  $\{\mathbf{d}_k^i\}_{k=1}^{H-1}$ 
4:   Rollout the policy  $\pi$  on disturbed system (1)
5:   Compute  $\sup_{0 \leq t < T} h(\mathbf{x}_t^i)$  using cubic splines
6: end for
7: Compute  $V_{T,N}^{h,\pi}(\mathbf{x}_0)$  according to (20)
8: Compute the gradient  $\nabla V_{T,N}^{h,\pi}(\mathbf{x}_0)$  using automatic differentiation

```

requires evaluating all possible disturbance trajectories. Instead, we propose to only consider  $N$  disturbance trajectories and take the worst-case out of the  $N$  samples, resulting in the following RPCBF approximation:

$$V_T^{h,\pi}(\mathbf{x}_0) \approx V_{T,N}^{h,\pi}(\mathbf{x}_0) := \max_{i=1, \dots, N} \sup_{0 \leq t < T} h(\mathbf{x}_t^i), \quad (20)$$

$$\dot{\mathbf{x}}_t^i = f(\mathbf{x}_t^i, \mathbf{d}_t^i) + g(\mathbf{x}_t^i, \mathbf{d}_t^i)\mathbf{u}_t^i. \quad (21)$$

We summarize our approach in Alg. 1 and Fig. 3. Note that Alg. 1 must be executed once per control loop to obtain the value  $V_{T,N}^{h,\pi}$  and gradient  $\nabla V_{T,N}^{h,\pi}$  for the current state for use in (CBF-QP). Different approaches can be implemented to perform informed sampling of disturbances. For bounded disturbances, the worst-case scenario often occurs at the vertices of the disturbance set (e.g., for disturbance-affine dynamics). Consequently, we choose to sample from a mixture of the uniform distribution  $\mathcal{U}(\mathbf{d}_{\min}, \mathbf{d}_{\max})$  and the uniform distribution over the vertices. Using better optimizers to approximate the worst-case samples will be left to future work.

While the finite-sample approximation does not guarantee robustness to any disturbance, it does ensure robustness to the specific sampled disturbances within the finite horizon. As the number of informed samples approaches infinity, the approximation increasingly captures the true worst-case scenarios. Theoretical guarantees for this sampling-based approach could be established using random set theory [43], as demonstrated in [44]. Furthermore, statistical risk measures could be used instead of the worst-case out of  $N$  samples. For instance, using [45] who bound the risk measure evaluation of a random variable whose distribution is unknown. However, we leave this for future work. For the simulation and hardware experiments below, we use PCBF and RPCBF to refer to their time-discretized finite-horizon and finite-sample approximations as described in this section.

## IV. SIMULATION EXPERIMENTS

We evaluate the performance of (R)PCBF in simulation for high relative degree systems with box control constraints.

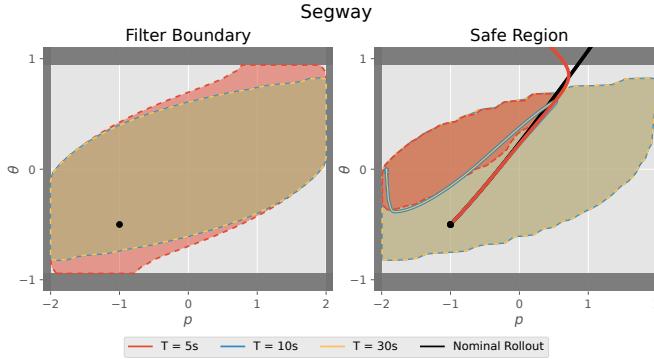
**Baselines.** We compare against the following SFs, which similarly do not incorporate NNs in their approach.

- **Handcrafted Candidate CBF (HOCBF) [46], [47]:** We construct a *candidate* CBF via a Higher-Order CBF on  $h$  without considering input constraints.
- **Approximate Nominal MPC-based PSF (MPC) [39]:** A trajectory optimization problem is solved, imposing the safety constraints while penalizing deviations from the nominal policy. We consider the undisturbed system without assuming access to a known robust forward-invariant set, thus not imposing a terminal constraint.

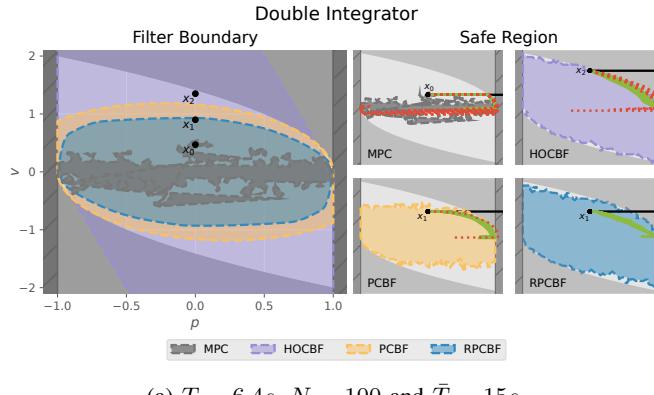
**Systems.** We consider four systems: a DI, a Segway, an F-16 fighter jet (ground collision avoidance problem) [48], [49], and AutoRally [50], a 1/5 autonomous vehicle. On the DI, we consider position bounds ( $|p| \leq 1$ ), while the Segway asks for an upright handlebar and considers position bounds ( $|\theta| \leq 0.3\pi$ ,  $|p| \leq 2$ ),  $\Delta t = 0.1$ . For the F-16, safety is defined as box constraints on states like altitude. Since this system is not control-affine in the throttle, we leave the throttle as the output of a P controller, resulting in a 16-dimensional state space and a 3-dimensional control space. In AutoRally, a crash occurs when the car stops after hitting the track boundary, while a collision involves contact without stopping. For each system, we define  $J$  continuously differentiable constraint functions  $h_j$  tailored to the problem at hand. For example, for the DI system we set  $h_0 = p - 1$  and  $h_1 = -(p + 1)$ . From these we derive  $J$  corresponding CBFs, which yield to  $J$  constraints in (CBF-QP). For the DI and the Segway, we assume unknown but bounded time-varying disturbances on the mass, for the F-16 unknown but bounded matched disturbances ( $d = 1$ ), and for the AutoRally additive truncated Gaussian noise. Keep in mind that our approach does not require designing/learning a new CBF for different systems, disturbance assumptions, or input constraints, but simply requires swapping the dynamics, specifying the disturbance and constraints. During testing, we consider a constant zero-control nominal policy for the DI, maximum acceleration for the Segway, a PID controller for the F-16, and Model Predictive Path Integral (MPPI) control [51] for AutoRally.

### A. Influence of Horizon Length on Undisturbed Segway

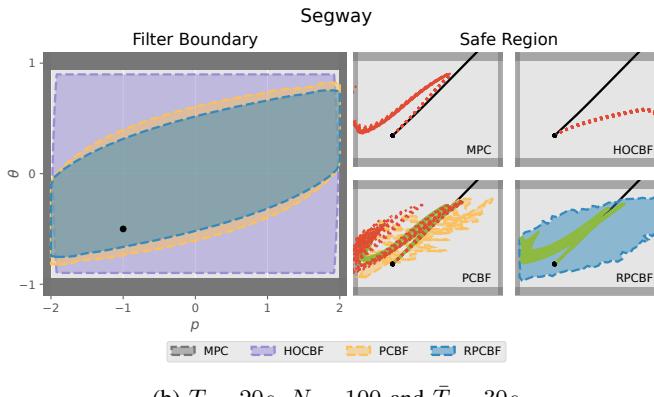
While the infinite-horizon policy value function is a CBF, we use a finite-horizon approximation, making the SF performance horizon-dependent. To illustrate this, we assess the impact of different horizon lengths on the PCBF-SF, see Fig. 4. We plot the state space from where  $\pi_{\text{nom}}$  can influence the output of the SF (*Filter Boundary*) and from where the SF preserves safety (*Safe Region*). For CBF-based filters, the filter boundary



**Fig. 4: Filter Boundary and Safe Region for PCBF-SFs with varying Horizon  $T$  on the Undisturbed Segway.** We plot where the nominal policy affects the SF’s output (Filter Boundary) and the states where the SF ensures safety over  $\bar{T} = 30s$  (Safe Region). Trajectories from an initial state within the filter boundary (black dot,  $\bullet$ ) are color-coded by horizon length of the PCBF-SF. A too-short horizon overapproximates the filter boundary, causing unsafe trajectories.



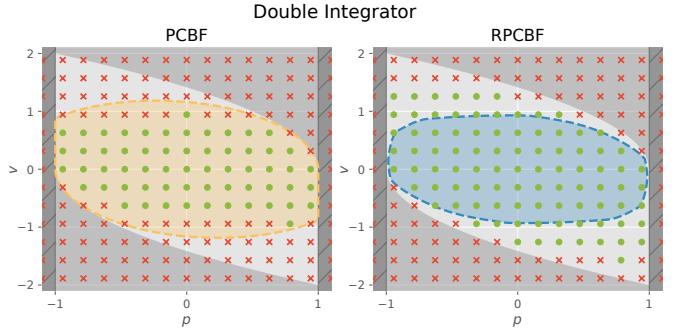
(a)  $T = 6.4s$ ,  $N = 100$  and  $\bar{T} = 15s$ .



(b)  $T = 20s$ ,  $N = 100$  and  $\bar{T} = 30s$

**Fig. 5: Comparison of Filter Boundary and Safe Region on DI (a) and Segway (b).** The true unsafe region for the undisturbed DI is shaded in gray. (R)PCBFs use horizon  $T$  and  $N$  samples to derive the value function. The safe region is determined for  $\bar{T}$ . Trajectories from selected initial states are shown for  $\bar{N} = 25$  sampled  $\mathbf{d}$  trajectories. Red dotted and green solid lines indicate unsafe and safe trajectories, respectively, with the nominal trajectory in black.

is defined by the CBF’s zero level set. The safe region is determined for a  $\pi_{\text{nom}}$  by solving (CBF-QP) and rolling out the system over a horizon  $\bar{T} = 30s$ . For a short PCBF horizon, i.e.,  $T = 5s$ , the true CI set is overapproximated. Consequently, the SF fails to preserve safety, as illustrated by the resulting unsafe example trajectory. In contrast, a longer horizon of  $T = 10s$  provides a much closer approximation of the true CI



**Fig. 6: Robust Safety Evaluation.** We plot the zero level set of the (R)PCBFs. Green dots indicate safe rollouts for all  $\bar{N} = 25$  sampled disturbance trajectories, while red crosses indicate a failure in at least one trajectory. The RPCBF-SF achieved safety for all states within its zero level set for the sampled disturbance trajectories.

set. This is evident when comparing it to an even longer horizon, such as  $T = 30s$ , which does not result in a visibly smaller safe region, indicating that  $T = 10s$  is already sufficient.

### B. Behavior on Disturbed Double Integrator and Segway

We explore the robustness of different SFs, examining the filter boundary and safe region, derived for one sampled disturbance trajectory per state, as shown in Fig. 5. We visualize rolled-out trajectories for  $\bar{N}$  sampled disturbance trajectories (uniformly sampled and on the vertices) from selected initial states within the filter boundary. On the DI, only the RPCBF-SF achieves safety for all  $\bar{N}$  sampled disturbance trajectories. Since the RPCBF accounts for the worst-case among the  $N$  sampled disturbances, the filter boundary is more conservative. On the Segway, MPC violates the safety constraints in all cases and hence has an empty filter boundary and safe region. Only the RPCBF-SF achieves safe trajectories for all considered samples. Next, we evaluate the (R)PCBF-SFs at uniformly distributed initial states, see Fig. 6. The RPCBF-SF achieves safety for all evaluated states within its zero-level set and the sampled disturbances, while the PCBF overapproximates the safe set.

### C. Simulations on AutoRally

Finally, to assess the safety improvements brought about by the proposed (R)PCBF, we integrate the HOCBF and the proposed methods with Shield-MPPI (SMPPI) [50], [52], and test them on AutoRally. Figure 7 shows that SMPPI using RPCBF generates the safest trajectories. The statistics of the safety performance of the controllers are shown in Tab. I.

**TABLE I: Collision & Crash Rate on AutoRally.** MPPI causes the most collisions and crashes. SMPPI-HOCBF and SMPPI-PCBF improve safety, while SMPPI-RPCBF minimizes collisions & crashes.

Controller	Mean Collisions per Lap	Crash Rate
MPPI	4.53	0.80
SMPPI-HOCBF	1.38	0.15
SMPPI-PCBF	1.23	0.12
SMPPI-RPCBF	<b>1.13</b>	<b>0.09</b>

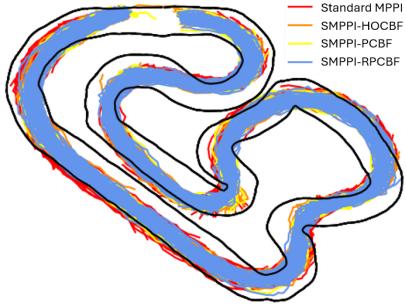


Fig. 7: **Trajectory Comparisons on AutoRally.** SMPPI-RPCBF (in blue) leads to the tightest spread of states inside the track.

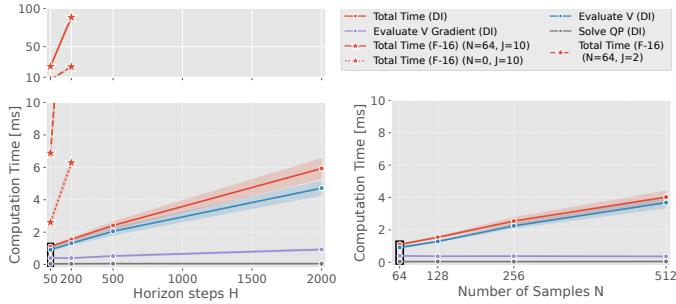


Fig. 8: **Per-Timestep Computation Times for Double Integrator and F-16.** Computation times for the total and individual components for varying  $H$  ( $N = 64$ ) and  $N$  ( $H = 50$ ), including mean and standard deviation across initial conditions and timesteps. The black box highlights the settings considered in the hardware experiments.

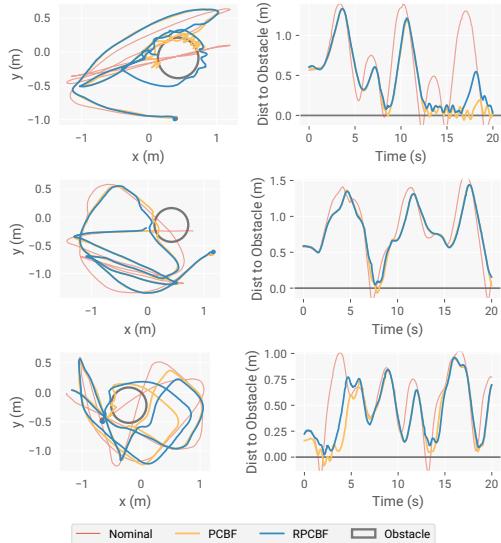


Fig. 9: **Hardware Traces.** RPCBF-SF maintains safety despite model errors, while PCBF-SF, assuming perfect dynamics, collides.

#### D. Comparison of Computation Times

Figure 8 shows the real-time feasibility of RPCBF-SF on the DI and F-16, highlighting how (component) computation times scale with increasing  $T$  and  $N$ , evaluated on a laptop CPU.

## V. HARDWARE EXPERIMENTS

We conduct hardware experiments on the Crazyflie platform to test the robustness of the proposed RPCBF-SF to real-world disturbances (see Fig. 1). Using the onboard position PID

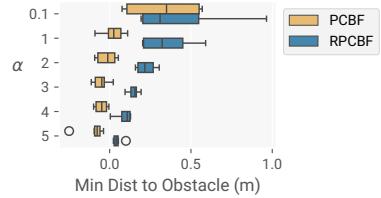


Fig. 10: **Safety for different  $\alpha$ .** Minimum distance to the obstacle over 6 random nominal trajectories. While PCBF is safe for small  $\alpha$ , it requires fine-tuning, whereas RPCBF is safe for all tested  $\alpha$ .

controller, we model the system as a DI, assuming that position setpoints are converted to accelerations onboard and treat the model error as an acceleration disturbance. We randomly generate nominal trajectories and treat the corresponding positions as the nominal control. A circular obstacle is placed at the densest part of the trajectory to encourage collisions. We use  $T = 5\text{s}$  (50 steps at  $\Delta t = 0.1\text{s}$ ) and  $N = 64$ . The RPCBF-SF runs at a frequency of 100 Hz on a laptop.

We run the (R)PCBF-SFs with  $\alpha = 5$  for 6 nominal trajectories; Fig. 9 shows results for 3 of them. The RPCBF-SF remains safe throughout, while the PCBF-SF always collides.

We next vary the choice of the class- $\kappa$  function  $\alpha$  and plot the results in Fig. 10. While the non-robust PCBF-SF does not collide with the obstacle when  $\alpha$  is sufficiently small, this requires fine-tuning and is difficult to know beforehand. On the other hand, the RPCBF-SF is safe for all values of  $\alpha$  we tested, allowing  $\alpha$  to be used as a parameter that controls the behavior without also simultaneously affecting safety.

## VI. DISCUSSION AND CONCLUSION

In this work, we proposed the Robust Policy Control Barrier Function (RPCBF), a method for constructing robust CBFs using the robust policy value function derived from rolling out a design policy. Subsequently, we introduced a real-time approximation that can be derived online, with conditions for its validity as a CBF. Simulation experiments demonstrate that a safety filter constructed using the RPCBF yields improved safety and more accurate estimation of the robust control-invariant set compared to existing methods. Hardware experiments on a quadcopter highlight the importance of accounting for model errors to ensure safety.

Future work will focus on analyzing the safety guarantees of the RPCBFs approximation, considering the finite-horizon, time-discretization, and sampling-based approach. Additionally, while the RPCBF acts as a CBF for any design policy if a long enough horizon is considered, conservativeness depends on the design policy. Deriving a policy to reduce conservativeness while maintaining infinite horizon guarantees is an important research direction. Moreover, extending our method to time-varying constraints and integrating real-time onboard perception are important directions for future work.

## ACKNOWLEDGEMENT

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense

for Research and Engineering. © 2024 Massachusetts Institute of Technology. Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work. DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

## 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, 2016.
- [2] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” *IFAC Proc. Vol.*, vol. 40, no. 12, pp. 462–467, 2007.
- [3] C. Dawson, S. Gao, and C. Fan, “Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control,” *IEEE Tr. Rob.*, vol. 39, no. 3, pp. 1749–1767, 2023.
- [4] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. Dimarogonas, S. Tu, and N. Matni, “Learning hybrid control barrier functions from data,” in *Conf. Robot. Learn. (CoRL)*. PMLR, 2021, pp. 1351–1370.
- [5] O. So, Z. Serlin, M. Mann, J. Gonzales, K. Rutledge, N. Roy, and C. Fan, “How to train your neural control barrier function: Learning safety filters for complex input-constrained systems,” in *2024 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2024, pp. 11532–11539.
- [6] M. Saveriano and D. Lee, “Learning barrier functions for constrained motion planning with dynamical systems,” in *2019 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2019, pp. 112–119.
- [7] S. Zhang, O. So, K. Garg, and C. Fan, “Gcbf+: A neural graph control barrier function framework for distributed safe multi-agent control,” *IEEE Tr. Robot.*, 2025.
- [8] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, “Synthesis of control barrier functions using a supervised machine learning approach,” in *2020 IEEE/RSJ IROS*. IEEE, 2020, pp. 7139–7145.
- [9] X. Wang, L. Knoedler, F. B. Mathiesen, and J. Alonso-Mora, “Simultaneous synthesis and verification of neural control barrier functions through branch-and-bound verification-in-the-loop training,” in *2024 Eur. Control Conf. (ECC)*. IEEE, 2024, pp. 571–578.
- [10] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, “Learning safe multi-agent control with decentralized neural barrier certificates,” in *Int. Conf. Learn. Represent. (ICLR)*, 2021.
- [11] C. Dawson, Z. Qin, S. Gao, and C. Fan, “Safe nonlinear control using robust neural lyapunov-barrier functions,” in *CoRL*. PMLR, 2022, pp. 1724–1735.
- [12] H. Yu, C. Hirayama, C. Yu, S. Herbert, and S. Gao, “Sequential neural barriers for scalable dynamic obstacle avoidance,” in *2023 IEEE/RSJ IROS*. IEEE, 2023, pp. 11241–11248.
- [13] M. Jankovic, “Robust control barrier functions for constrained stabilization of nonlinear systems,” *Autom.*, vol. 96, pp. 359–367, 2018.
- [14] M. H. Cohen, C. Belta, and R. Tron, “Robust control barrier functions for nonlinear control systems with uncertainty: A duality-based approach,” in *2022 IEEE 61st Conf. Decis. Control (CDC)*. IEEE, 2022, pp. 174–179.
- [15] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, “A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [16] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, “Robust control barrier-value functions for safety-critical control,” in *2021 60th IEEE CDC*. IEEE, 2021, pp. 6814–6821.
- [17] 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 Syst. Mag.*, vol. 43, no. 5, pp. 137–177, 2023.
- [18] S. Tonkens and S. Herbert, “Refining control barrier functions through hamilton-jacobi reachability,” in *2022 IEEE/RSJ IROS*. IEEE, 2022, pp. 1335–13362.
- [19] S. Tonkens, A. Toofanian, Z. Qin, S. Gao, and S. Herbert, “Patching neural barrier functions using hamilton-jacobi reachability,” *arXiv preprint arXiv:2304.09850*, 2023.
- [20] I. M. Mitchell, “The flexible, extensible and efficient toolbox of level set methods,” *Journal of Scientific Comput.*, vol. 35, pp. 300–329, 2008.
- [21] S. Bansal and C. J. Tomlin, “Deepreach: A deep learning approach to high-dimensional reachability,” in *2021 IEEE ICRA*. IEEE, 2021, pp. 1817–1824.
- [22] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac, “Safety and liveness guarantees through reach-avoid reinforcement learning,” *arXiv preprint arXiv:2112.12288*, 2021.
- [23] O. So and C. Fan, “Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning,” *arXiv preprint arXiv:2305.14154*, 2023.
- [24] A. Lin, S. Peng, and S. Bansal, “One filter to deploy them all: Robust safety for quadrupedal navigation in unknown environments,” *arXiv preprint arXiv:2412.09989*, 2024.
- [25] M. Ahmadi, X. Xiong, and A. D. Ames, “Risk-averse control via cvar barrier functions: Application to bipedal robot locomotion,” *IEEE Control Systems Letters*, vol. 6, pp. 878–883, 2021.
- [26] S. Liu and C. A. Belta, “Risk-aware adaptive control barrier functions for safe control of nonlinear systems under stochastic uncertainty,” *arXiv preprint arXiv:2503.19205*, 2025.
- [27] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
- [28] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, “Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2897–2904, 2022.
- [29] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, “Backup control barrier functions: Formulation and comparative study,” in *2021 60th IEEE CDC*. IEEE, 2021, pp. 6835–6841.
- [30] D. R. Agrawal, R. Chen, and D. Panagou, “gatekeeper: Online safety verification and control for nonlinear systems in dynamic environments,” *IEEE Tr. Robot.*, 2024.
- [31] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Autom.*, vol. 36, no. 6, pp. 789–814, 2000.
- [32] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, “Model predictive contouring control for collision avoidance in unstructured dynamic environments,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4459–4466, 2019.
- [33] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” *arXiv preprint arXiv:1909.06586*, 2019.
- [34] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, “Autonomous racing using linear parameter varying-model predictive control (lpv-mpc),” *Control Eng. Pract.*, vol. 95, p. 104270, 2020.
- [35] Z. Wang, O. So, J. Gibson, B. Vlahov, M. S. Gandhi, G.-H. Liu, and E. A. Theodorou, “Variational inference mpc using tsallis divergence,” *arXiv preprint arXiv:2104.00241*, 2021.
- [36] O. So, Z. Wang, and E. A. Theodorou, “Maximum entropy differential dynamic programming,” in *2022 IEEE ICRA*. IEEE, 2022, pp. 3422–3428.
- [37] A. Boccia, L. Grüne, and K. Worthmann, “Stability and feasibility of state constrained mpc without stabilizing terminal constraints,” *Syst. & Control Lett.*, vol. 72, pp. 14–21, 2014.
- [38] C. Kirches, L. Wirsching, H. G. Bock, and J. P. Schlöder, “Efficient direct multiple shooting for nonlinear model predictive control on long horizons,” *Journal of Process Control*, vol. 22, no. 3, pp. 540–550, 2012.
- [39] K. P. Wabersich and M. N. Zeilinger, “A predictive safety filter for learning-based control of constrained nonlinear dynamical systems,” *Autom.*, vol. 129, p. 109597, 2021.
- [40] C. De Boor, “A practical guide to splines,” *Springer-Verlag*, vol. 2, pp. 4135–4195, 1978.
- [41] A. Altarovici, O. Bokanowski, and H. Zidani, “A general hamilton-jacobi framework for non-linear state-constrained control problems,” *ESAIM: Control, Optim. Calc. Variat.*, vol. 19, no. 2, pp. 337–357, 2013.
- [42] Q. Nguyen and K. Sreenath, “Robust safety-critical control for dynamic robotics,” *IEEE Trans. Autom. Control*, vol. 67, no. 3, pp. 1073–1088, 2021.
- [43] I. Molchanov and I. S. Molchanov, *Theory of random sets*. Springer, 2005, vol. 19, no. 2.
- [44] T. Lew and M. Pavone, “Sampling-based reachability analysis: A random set theory approach with adversarial sampling,” in *CoRL*. PMLR, 2021, pp. 2055–2070.
- [45] P. Akella, A. Dixit, M. Ahmadi, J. W. Burdick, and A. D. Ames, “Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification,” *Art. Intell.*, vol. 336, p. 104195, 2024.
- [46] Q. Nguyen and K. Sreenath, “Exponential control barrier functions for enforcing high relative-degree safety-critical constraints,” in *2016 Amer. Control Conf. (ACC)*. IEEE, 2016, pp. 322–328.
- [47] W. Xiao and C. Belta, “Control barrier functions for systems with high relative degree,” in *2019 IEEE 58th CDC*. IEEE, 2019, pp. 474–479.

- [48] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, “Verification challenges in f-16 ground collision avoidance and other automated maneuvers.” *ARCH@ ADHS*, vol. 2018, 2018.
- [49] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.
- [50] J. Yin, C. Dawson, C. Fan, and P. Tsiotras, “Shield model predictive path integral: A computationally efficient robust mpc method using control barrier functions,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7106–7113, 2023.
- [51] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras, “Trajectory distribution control for model predictive path integral control using covariance steering,” in *2022 IEEE ICRA*, 2022, pp. 1478–1484.
- [52] J. Yin, P. Tsiotras, and K. Berntorp, “Chance-constrained information-theoretic stochastic model predictive control with safety shielding,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.00494>