RPCBF: Constructing Safety Filters Robust to Model Error and Disturbances via Policy Control Barrier Functions

Luzia Knoedler[1*], Oswin So[2*], Ji Yin[3], Mitchell Black[4],
Zachary Serlin[4], Panagiotis Tsiotras[3], Javier Alonso-Mora[1], and Chuchu Fan[2]

***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 method of constructing CBF approximations that is easy to implement and robust to disturbances via the estimation of a value function. We demonstrate the effectiveness of our method in simulation on a variety of high relative degree input-constrained systems. Finally, we demonstrate the benefits of RPCBF in compensating for model errors on a hardware quadcopter platform by treating the model errors as disturbances.**

I. Introduction and Related Works

In the realm of autonomous systems, providing safety guarantees is crucial, especially in critical applications such as autonomous driving [1] and healthcare robotics [2]. Control Barrier Functions (CBFs) [3, 4] 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 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 can result in the safety of these controllers being 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) [5–13]. Leveraging the flexibility of NNs, neural CBFs have been successfully applied to high-dimensional systems, including multi-agent control scenarios [11, 14]. Furthermore, they have been extended to handle parametric uncertainties [15] and obstacles with unknown dynamics [16]. Although utilizing NNs as CBFs offers universal approximation capabilities, it requires their certification as CBF to provide safety guarantees and limits their interpretability.

*Both authors contributed equally to this work.
[1]Delft University of Technology, Delft, The Netherlands
[2]Massachusetts Institute of Technology, Cambridge, USA
[3]Georgia Institute of Technology, Atlanta, USA
[4]MIT Lincoln Laboratory, Cambridge, USA

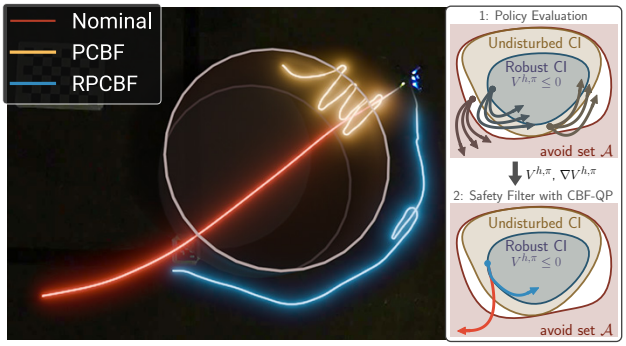


Fig. 1: We propose the Robust Policy Control Barrier Function (RPCBF), which approximates the value function $V^{h,\pi}$ of a system under bounded disturbances for policy π online. The zero sublevel set of the RPCBF is a robust controlled-invariant set (CI) and can be used online as a safety filter to ensure safety for any unsafe nominal policy. We demonstrate its superior performance compared to a safety filter using a PCBF on a quadcopter platform with model errors treated as disturbances.

Furthermore, using a naive approach of learning neural CBFs by minimizing a loss that encourages the CBF conditions can lead to a small or even empty forward-invariant set. Thus, [8] present 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 of it. In this setting, the policy value function represents the maximum-over-time constraint violation and is generally a measure of how good a particular state is for a system following a specific policy over infinite time. However, their approach does not consider uncertainties in the system dynamics.

Robust Safety. Controllers robust to disturbances in the real world are important for safe autonomous systems. This has been studied before in *robust* CBFs [15, 17, 18] which guarantee safety under bounded disturbances. However, constructing robust CBFs is more difficult than regular CBFs, especially under input constraints. Hamilton-Jacobi reachability analysis [19] can be used to compute robust control-invariant sets, which can then be subsequently used for constructing robust CBFs [20–23]. However, reachability analysis in itself is challenging, with grid-based PDE solvers being limited to state dimensions less than 5 [24], while deep learning-based solvers [25–28] require subsequent neural-network verification to check for solution accuracy.

To address these challenges, we propose a practical approach to construct robust CBFs that guarantee safety under worse-case bounded disturbances through policy evaluation of *any* policy, by estimating the policy value

function, which has been demonstrated to be a valid CBF in [8]. Our approach leverages finite horizon policy rollouts and extends the method to handle the robust case.

Contributions. We summarize our contributions as follows.

1) We propose a method of constructing robust CBFs using the robust policy value function and subsequently propose a real-time approximation that can be used online.
2) We examine the benefits of robust CBFs on a hardware quadcopter, where the additional robustness to model error is crucial to prevent collisions.

II. Preliminaries

A. Problem Statement

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

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

with state $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$, (unknown) bounded smooth disturbance $\mathbf{d}_{\min} \leq \mathbf{d} \leq \mathbf{d}_{\max}$ with $\mathbf{d}_{\min}, \mathbf{d}_{\max} \in \mathbb{R}^d$ and f, g locally Lipschitz continuous. Let $\mathcal{A} \subset \mathcal{X}$ denote the set of states to be avoided. In this paper, we address the following SF synthesis problem:

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

$$\begin{aligned} &\min_{\pi} \|\pi_{\text{filt}} - \pi_{\text{nom}}\| \\ &\quad \textit{s.t. } \mathbf{x}_t \notin \mathcal{A},\ \forall t \geq 0, \end{aligned} \tag{2}$$

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

We focus on solving Prob. 1 using (zeroing) CBFs [29].

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

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

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

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

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

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

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

renders $\mathcal{C}$ forward-invariant [3]. 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 $\mathbf{x} \in \mathcal{X}$ [29]. Since the right hand side of condition (4b) is linear in $\mathbf{u}$, given a CBF B where (4b) is satisfied, we can solve Prob. 1 for (3) using the following Quadratic Program (QP)-based controller:

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

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

1) How to synthesize a B that satisfies (4b) so it is a CBF?
2) How do we guarantee that using the CBF for control synthesis results in safe controllers when applied to the original *disturbed* system?

III. Robust Policy Control Barrier Functions

To address the above two challenges, we leverage the insight from [8] 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 [8], we propose an alternative that avoids NNs by instead performing a finite-horizon numerical approximation. We further extend this approach to the robust case and introduce Robust Policy CBFs (RPCBFs). We begin by revisiting the formulation of PCBFs and describe our extensions.

A. Constructing CBFs via Policy Evaluation

Based on [8], 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 $h : \mathcal{X} \to \mathbb{R}$ (e.g., the negative distance to the constraint):

$$\mathcal{A} = \{\mathbf{x} \in \mathcal{X} \,|\, h(\mathbf{x}) > 0\}. \tag{5}$$

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 $\pi : \mathcal{X} \to \mathcal{U}$. Furthermore, we define the *maximum-over-time* value function for the undisturbed system in (3) as

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

As stated in [8, Theorem 1], the *policy value function* $V_\infty^{h,\pi}$ is a CBF for the undisturbed system in (3) for any π, since $V_\infty^{h,\pi}$ satisfies the following two inequalities $\forall \mathbf{x} \in \mathcal{X}$

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

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

which imply (4a) and (4b). For details, we refer to [8]. 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 π. Thus, CBFs can be constructed via policy evaluation of any policy.

[1] Extended class-κ_∞ is the set of continuous, strictly increasing functions $\alpha : (-\infty, \infty) \to (-\infty, \infty)$ with $\alpha(0) = 0$.

B. Finite Horizon Approximation of PCBFs

A key challenge with the value function $V_\infty^{h,\pi}$ is that its definition requires an *infinite-horizon*. While [8] 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 NN, enabling a more in-depth analysis of the resulting safety guarantees. Expanding $V_\infty^{h,\pi}$:

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

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

where the approximation is made by dropping the $V_\infty^{h,\pi}(\mathbf{x}_T)$ "tail". The question is then whether the finite-horizon approximation $V_T^{h,\pi}$ is a CBF and has 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 in the following theorem.

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

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

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

Proof. Since $V_T^{h,\pi}(\mathbf{x}) \ge h(\mathbf{x})$ 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 $V_T^{h,\pi}(\mathbf{x}_0) \le 0$. 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. □

This enables us to prove the following corollary.

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

$$\arg\max_{t\ge 0} h(\mathbf{x}_t^\pi) < \bar{T}, \quad \forall \mathbf{x}_0 \text{ where } V_T^{h,\pi}(\mathbf{x}_0) \le 0 \tag{12}$$

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

Proof. (12) implies (11) for $T \ge \bar{T}$. The proof then follows from Theorem 1. □

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 [30, 31], practitioners often apply MPC without the use of such a terminal constraint set to wide success [32–35]. 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 [36], similar to corollary 1.

Remark (Connections to Backup Controller / CBFs)**.** Since the zero sublevel set of $V_\infty^{h,\pi}$ is a controlled-invariant set under π, (9) can also be seen as Backup CBF [37, 38] with backup controller π. Unlike this (and other similar approaches [39]), our approach replaces the need for a known forward-invariant set with the requirement of a sufficiently long horizon T.

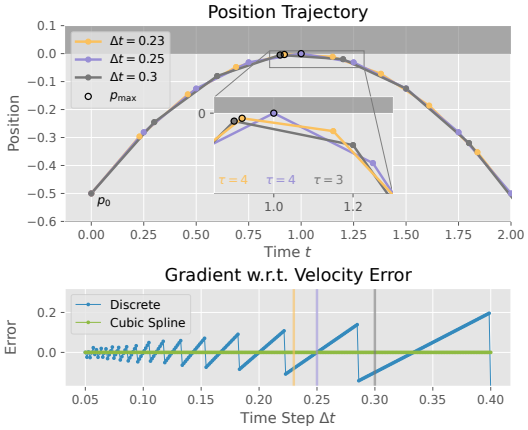


Fig. 2: **Value Function Gradient Error for Discrete-Time Double Integrator.** We highlight the discretized trajectory (top) and corresponding gradient error (bottom) for three different choices of Δt (yellow, purple, grey). The gradient of the naive discrete-time approximation has large errors and varies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

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 ODE 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 Δt,

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

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 $\dot{p} = v$, $\dot{v} = a$ with initial state $\mathbf{x}_0 = [p_0, v_0]$ and constraints $h(\mathbf{x}) = p \le 0$. For the continuous-time case, the gradient can be derived as

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

After (exact) time discretization with timestep Δt, the time-discretized states can be computed as

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

$$v_k = v_0 + ak\Delta t. \tag{15b}$$

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

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

with $\frac{\partial p_{\max}}{\partial v_0} = \tau\Delta t$, which is a function of τ. Although τ 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 Δt.

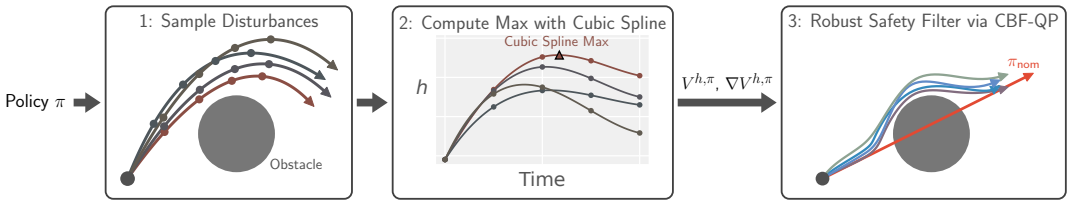


Fig. 3: **Summary of RPCBF Algorithm.** Given a policy π, we sample disturbance trajectories, then compute the maximum h with cubic splines to obtain $V^{h,\pi}$ and $\nabla V^{h,\pi}$ (using autodiff). This is used in a CBF-QP to obtain a robust safety filter.

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 $\sup_{0\le t<T} h(\mathbf{x}_t^\pi)$ than the naive maximization (13). Intuitively, this resolves the gradient error due to integer-valued τ from the previous example because the maximum of the cubic spline can now happen *between* timesteps. Applying this method to the previous example of the DI results in a significantly smaller gradient error (Fig. 2).

D. Robust Extension of PCBFs

In this subsection, we now consider a *robust* extension of PCBF to handle disturbances. Defining the *robust* value function equivalent of (6) as

$$V_\infty^{h,\pi}(\mathbf{x}_0) := \sup_{t\ge 0} \sup_{\mathbf{d}(\cdot)} h(\mathbf{x}_t^\pi), \tag{17}$$

it can be shown with similar proof [8, 40] that $V_\infty^{h,\pi}$ is a *robust* CBF [41], i.e., it satisfies (4a) and, for $B(\mathbf{x}) \le 0$,

$$\sup_{\mathbf{d}\in\mathcal{D}} \inf_{\mathbf{u}\in\mathcal{U}} \nabla B^\mathsf{T}\big(f(\mathbf{x},\mathbf{d}) + g(\mathbf{x},\mathbf{d})\mathbf{u}\big) \le -\alpha\big(B(\mathbf{x})\big). \tag{18}$$

Solving for robust controls that satisfy (18) renders the zero sublevel set *robust* forward-invariant [17].

However, deriving the worst-case disturbance is generally intractable because it 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 practical Robust Policy Control Barrier Function (RPCBF) approximation.

$$V_T^{h,\pi}(\mathbf{x}_0) \approx V_{T,N}^{h,\pi}(\mathbf{x}_0) := \max_{i=1,\cdots,N} \sup_{0\le t<T} h(\mathbf{x}_t^i), \tag{19}$$

$$\dot{\mathbf{x}}^i = f(\mathbf{x}^i,\mathbf{d}^i) + g(\mathbf{x}^i,\mathbf{d}^i)\mathbf{u}^i. \tag{20}$$

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 the CBF-QP (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. This can be extended to better optimizers to approximate the worst-case samples which we leave as 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. 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.

Algorithm 1 Robust Policy CBF (RPCBF)

Input: Initial State $\mathbf{x}_0$, Policy π, Constraint function h, Horizon $T = H\Delta t$, Number of disturbance samples N
for $i = 1 : N$ **do**
  Sample disturbance trajectory $\{\mathbf{d}_k^i\}_{k=1}^{H-1}$
  Rollout the policy π on disturbed system (1)
  Compute $\sup_{0\le t<T} h(\mathbf{x}_t^i)$ using cubic splines
end for
Compute $V_{T,N}^{h,\pi}(\mathbf{x}_0)$ according to (19)
Compute the gradient $\nabla V_{T,N}^{h,\pi}(\mathbf{x}_0)$ using autodiff

IV. Simulation Experiments

To study the performance of PCBF and RPCBF, we perform a series of simulation experiments on high relative degree systems under box control constraints.

Baselines. We compare against the following SFs that also do not use NNs in their approach.

- **Handcrafted Candidate CBF (HOCBF) [42, 43]:** We construct a *candidate* CBF via a Higher-Order CBF on h without considering input constraints.
- **Approximate Nominal MPC-based Predictive Safety Filter (MPC) [44]:** A trajectory optimization problem is solved, imposing the safety constraints while penalizing deviations from the nominal policy. We consider the undisturbed system and do not assume access to a known robust forward-invariant set and hence do not impose this terminal constraint.

Systems. We consider three systems: a DI, a Segway, and AutoRally [45], an 1/5 autonomous vehicle. On the DI, safety is defined via position bounds ($|p| \le 1$), while the segway asks for the handlebar to stay upright and for position bounds to be maintained ($|\theta| \le 0.3\pi$, $|p| \le 2$), $\Delta t = 0.1$. For AutoRally, a crash occurs when the car hits the track boundary and comes to a complete stop, while a collision is defined as the car making contact with the track boundary without coming to a halt. For the DI and the Segway, we

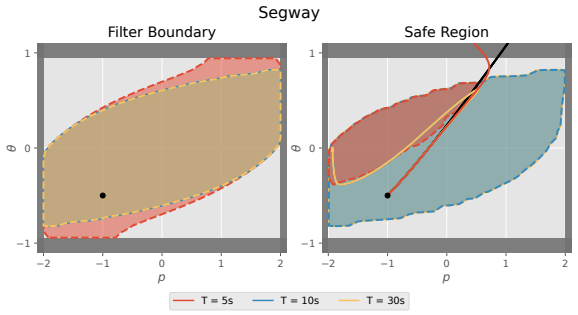


Fig. 4: **Filter Boundary and Safe Region for PCBFs with varying Horizon T on the Segway**. We plot the states where the nominal policy can influence the output of the safety filter (Filter Boundary) and the initial states from which the safety filter can preserve safety over a horizon $\bar{T} = 30s$ (Safe Region). A trajectory from an initial state inside the filter boundary marked by a black dot (•) is displayed. If the horizon is too short, the filter boundary is overapproximated, resulting in an unsafe trajectory.

assume an unknown but bounded mass and treat the unknown mass as a disturbance to the system. For the AutoRally, we assume additive truncated Gaussian noise. During testing, we consider a constant zero-control nominal policy for the DI, a maximum-acceleration control nominal policy for the segway, and use model predictive path integral (MPPI) [46] as the nominal policy for AutoRally. Note that the nominal policy π_{nom} differs from the policy π used to derive the PCBF.

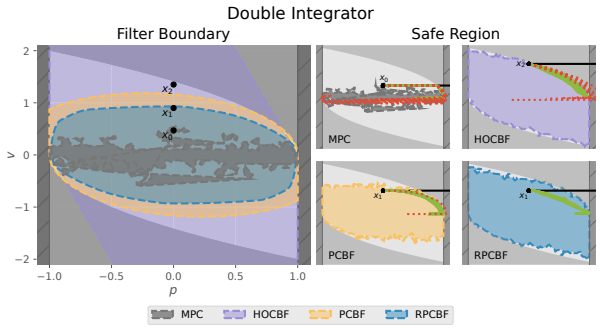
A. *Influence of Horizon Length on Segway*

While the infinite-horizon policy value function is a CBF, we consider a finite-horizon approximation. The performance of the SF is thus dependent on the chosen horizon length. To gain a better intuition, we evaluate how different horizon lengths affect the performance of the PCBF-SF. Here, we consider the undisturbed Segway.

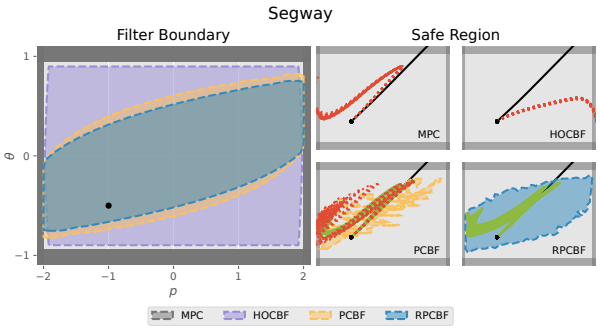
We visualize the results in Fig. 4, plotting the state space from where π_{nom} can influence the output of the SF (Filter Boundary) and from where the SF preserves safety (Safe Region). For CBF-based filters, the Filter Boundary is defined by the zero level set of the CBF. The Safe Region is determined for a π_{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 Control Invariant (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 $T = 20s$ provides a much closer approximation of the true CI set. This is evident when comparing it to an even longer horizon, such as $T = 30s$, which does not result in a visibly larger safe region, indicating that $T = 20s$ is already sufficient to capture the safety requirements accurately.

B. *Behavior on Disturbed Double Integrator and Segway*

Next, we explore the robustness of the different SFs. We again examine the filter boundary and safe region, this time they are derived for one sampled disturbance/ disturbance trajectory per state and are illustrated in Fig. 5. We visualize rolled-out trajectories for $\bar{N}$ sampled disturbances (uniformly sampled and on the vertices) from initial states within the filter boundary of the considered method. On the DI, only



(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 the Double Integrator (a) and Segway (b).** The true unsafe region for the undisturbed system is shaded in gray for the double integrator. PCBF-based methods use a horizon T and N samples to derive the policy value function. The Safe Region is determined for a horizon $\bar{T}$. Trajectories from selected initial states for $\bar{N} = 25$ sampled disturbance trajectories are displayed. Red dotted lines and green solid lines indicate unsafe and safe trajectories, respectively. The nominal trajectory is shown in black.

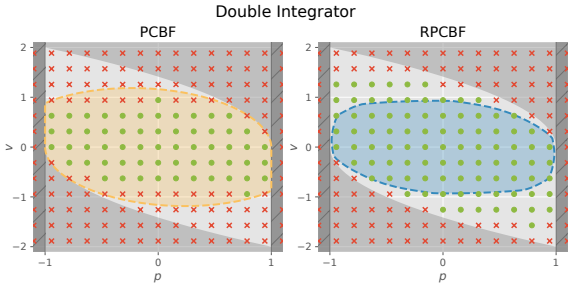


Fig. 6: **Robust Safety Evaluation.** We plot the zero level set of the CBFs. Green dots indicate safe rollouts for all $\bar{N} = 25$ sampled disturbance trajectories, while red crosses indicate a failure in at least one trajectory. RPCBF achieved safety for all states within its zero level set for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

RPCBF achieved safety for all $\bar{N}$ sampled disturbances. Since RPCBF accounts for the worst-case scenario among the sampled disturbance trajectories, the filter boundary is more conservative. On the segway, MPC violated safety constraints in all cases and hence has an empty filter boundary and safe region. Only RPCBF achieves safe trajectories for all considered samples. Furthermore, we evaluate the PCBF-based SFs on states uniformly distributed across the state space for $\bar{N}$ sampled disturbances. The results are displayed in Fig. 6. The RPCBF-based SF achieves safety

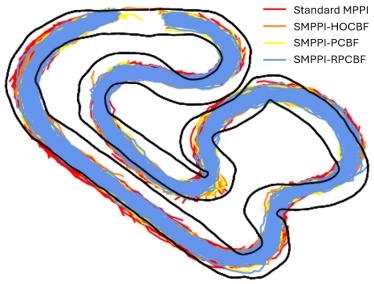


Fig. 7: **Trajectory Comparisons on AutoRally.** We visualize the trajectories for each controller, where SMPPI-RPCBF (in blue) leads to the tightest spread of states inside the track.

for all evaluated states within its zero level set.

C. Simulations on AutoRally

Finally, to assess the safety improvements brought about by the proposed PCBF and RPCBF, we integrate the HOCBF and the proposed methods with Shield-MPPI [45, 47], and evaluate their performance on AutoRally. The Fig. 7 shows that Shield-MPPI using RPCBF generates the safest trajectories, while other controllers generate trajectories that collide and crash more often. The statistics of the safety performance of the controllers are shown in Tab. I.

TABLE I: **Collision and Crash Rate on AutoRally.** The standard MPPI leads to most collisions and crashes. While Shield-MPPI with HOCBF and PCBF significantly improves safety, Shield-MPPI using RPCBF achieves the lowest rate of collision and 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	**1.13**	**0.09**

V. Hardware Experiments

We conduct hardware experiments on the Crazyflie platform to determine whether the proposed RPCBF can be robust to disturbances encountered in the real world (see Fig. 1). We use the onboard position PID controller on the Crazyflie, treating the system as a DI, and assume that the position setpoints are converted into accelerations onboard. The error between this simple model and the true dynamics is treated as an acceleration disturbance. We randomly generate a nominal trajectory and treat the corresponding positions as the nominal control. A circular obstacle is placed at the densest part of the nominal trajectory to encourage collisions. We use a $T = 5s$ (50 steps at $\Delta t = 0.1\,\text{s}$) and sample 64 disturbance trajectories. The RPCBF controller runs at a frequency of 100 Hz on a laptop.

We first run the PCBF and RPCBF controllers with $\alpha = 5$ on 6 different nominal trajectories and plot the results from 3 of the random trajectories in Fig. 8. The RPCBF maintains safety in all cases, while the PCBF collides in all cases.

We next vary the choice of the class-κ function α and plot the results in Fig. 9. While the non-robust PCBF does not collide with the obstacle when α is sufficiently small, this requires fine-tuning and is difficult to know beforehand.

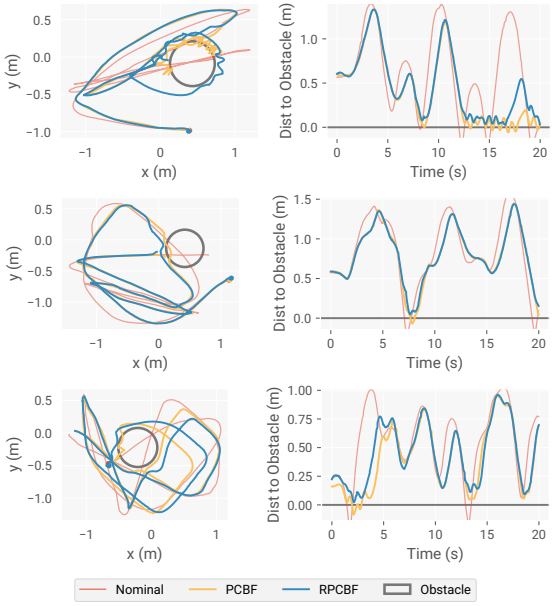


Fig. 8: **Hardware Traces.** RPCBF maintains safety while following the unsafe nominal trajectory despite error between the modeled and true Crazyflie dynamics. On the other hand, PCBF assumes the modeled dynamics are perfect and collides with the obstacle.

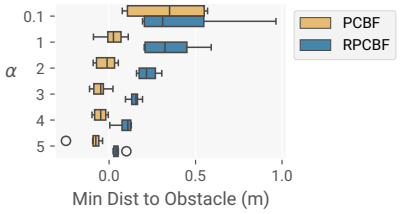


Fig. 9: **Safety with different values of α.** We plot the minimum distance to the obstacle, over 6 random nominal trajectories. Although PCBF is safe when α is sufficiently small, this requires fine-tuning and is difficult to know beforehand. RPCBF is safe for all values of α we tested.

On the other hand, RPCBF is safe for all values of α we tested, allowing α to be used as a parameter that controls the behavior without also simultaneously affecting safety.

VI. Discussion and Conclusion

In this work, we have proposed the Robust Policy Control Barrier Function (RPCBF), a method of constructing SFs robust to disturbances via robust CBFs. Simulation experiments show that RPCBF yields improved safety and more accurate estimation of the robust control-invariant set compared to existing methods. Hardware experiments on the quadcopter show that considering model errors is crucial to maintaining safety.

Future directions include performing a more thorough analysis of the safety guarantees of RPCBFs. In particular, although we have provided a simple theorem proving that a long enough horizon is sufficient, this does not account for time-discretization of the rollouts. Also, as noted in Sec. IV-A, the forward-invariant set of the RPCBF is sometimes larger than the zero sublevel set. Whether rigorous statements can be made about this case is another interesting future direction to explore.

ACKNOWLEDGEMENT

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. 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.

REFERENCES

[1] J. Betz, A. Heilmeier, A. Wischnewski, T. Stahl, and M. Lienkamp, "Autonomous driving—a crash explained in detail," *Applied Sciences*, vol. 9, no. 23, p. 5126, 2019.

[2] T. Haidegger, "Autonomy for surgical robots: Concepts and paradigms," *IEEE Transactions on Medical Robotics and Bionics*, vol. 1, no. 2, pp. 65–76, 2019.

[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, 2016.

[4] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[5] 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 Transactions on Robotics*, vol. 39, no. 3, pp. 1749–1767, 2023.

[6] A. Peruffo, D. Ahmed, and A. Abate, "Automated and formal synthesis of neural barrier certificates for dynamical models," in *International conference on tools and algorithms for the construction and analysis of systems*. Springer, 2021, pp. 370–388.

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

[8] 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 International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 532–11 539.

[9] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3717–3724.

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

[11] S. Zhang, O. So, K. Garg, and C. Fan, "Gcbf+: A neural graph control barrier function framework for distributed safe multi-agent control," *arXiv preprint arXiv:2401.14554*, 2024.

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

[13] 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 European Control Conference (ECC)*. IEEE, 2024, pp. 571–578.

[14] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," *arXiv preprint arXiv:2101.05436*, 2021.

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

[16] H. Yu, C. Hirayama, C. Yu, S. Herbert, and S. Gao, "Sequential neural barriers for scalable dynamic obstacle avoidance," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11 241–11 248.

[17] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[18] 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 Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 174–179.

[19] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[20] 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 Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6814–6821.

[21] 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.

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

13 355–13 362.

[23] 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.

[24] i. m. mitchell, "the flexible, extensible and efficient toolbox of level set methods," *journal of scientific computing*, vol. 35, pp. 300–329, 2008.

[25] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1817–1824.

[26] 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.

[27] M. Ganai, S. Gao, and S. Herbert, "Hamilton-jacobi reachability in reinforcement learning: A survey," *IEEE Open Journal of Control Systems*, 2024.

[28] O. So and C. Fan, "Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning," *arXiv preprint arXiv:2305.14154*, 2023.

[29] 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.

[30] 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.

[31] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[32] 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.

[33] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (lpv-mpc)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.

[34] 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.

[35] O. So, Z. Wang, and E. A. Theodorou, "Maximum entropy differential dynamic programming," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3422–3428.

[36] A. Boccia, L. Grüne, and K. Worthmann, "Stability and feasibility of state constrained mpc without stabilizing terminal constraints," *Systems & control letters*, vol. 72, pp. 14–21, 2014.

[37] 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.

[38] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6835–6841.

[39] D. R. Agrawal, R. Chen, and D. Panagou, "gatekeeper: Online safety verification and control for nonlinear systems in dynamic environments," *IEEE Transactions on Robotics*, 2024.

[40] A. Altarovici, O. Bokanowski, and H. Zidani, "A general hamilton-jacobi framework for non-linear state-constrained control problems," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 19, no. 2, pp. 337–357, 2013.

[41] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1073–1088, 2021.

[42] ——, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.

[43] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th conference on decision and control (CDC)*. IEEE, 2019, pp. 474–479.

[44] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[45] 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 Robotics and Automation Letters*, vol. 8, no. 11, pp. 7106–7113, 2023.

[46] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras, "Trajectory distribution control for model predictive path integral control using covariance steering," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1478–1484.

[47] 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