Learning for Layered Safety-Critical Control with Predictive Control Barrier Functions

William D. Compton WCOMPTON@CALTECH.EDU
Max H. Cohen MAXCOHEN@CALTECH.EDU
Aaron D. Ames AMES@CALTECH.EDU
Caltech, Pasadena, CA, USA

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Safety filters leveraging control barrier functions (CBFs) are highly effective for enforcing safe behavior on complex systems. It is often easier to synthesize CBFs for a Reduced order Model (RoM), and track the resulting safe behavior on the Full order Model (FoM)—yet gaps between the RoM and FoM can result in safety violations. This paper introduces *Predictive CBFs* to address this gap by leveraging rollouts of the FoM to define a predictive robustness term added to the RoM CBF condition. Theoretically, we prove that this guarantees safety in a layered control implementation. Practically, we learn the predictive robustness term through massive parallel simulation with domain randomization. We demonstrate in simulation that this yields safe FoM behavior with minimal conservatism, and experimentally realize Predictive CBFs on a 3D hopping robot.

Keywords: safety, control barrier functions, reduced order models, supervised learning

1. Introduction

Safety is a primary concern in the construction of modern control architectures. Control Barrier Functions (CBFs) provide a theoretically grounded method for achieving safe behaviors of complex nonlinear systems (Ames et al., 2014). CBFs have several desirable properties, including nominal robustness properties (Xu et al., 2015) and the ability to filter desired system inputs for safety (safety filtering) (Ames et al., 2016, 2019). These benefits have established CBFs and safety filters as practical tools, commonly used in collision avoidance, and other applications, on various robotic systems (Wabersich et al., 2023). Yet synthesizing CBFs for arbitrary (often even simple) safety constraints is difficult and impractical for high-dimensional or complex nonlinear systems. Significant prior work exists surrounding constructive methods for CBF synthesis, leveraging different system properties, such as relative degree (Nguyen and Sreenath, 2016), differential flatness (Wang et al., 2017), HJ reachability (Tonkens and Herbert, 2022), backup controllers (Chen et al., 2021b), or learning (Robey et al., 2020). However, complex systems often will not satisfy required assumptions, and high dimensionality cripples learning or PDE-based methods. Hence, constructive synthesis of CBFs for complex, high-order systems is an open problem (Cohen et al., 2024b).

One recent approach to addressing this problem is through layered control architectures (Matni et al., 2024), which leverage reduced order models in conjunction with existing controllers lower in the control stack. Often, safety constraints depend on a subset of system states; for instance, collisional avoidance problems typically only depend on kinematic configuration, and safety requirements for mobile robots often concern their center of mass positions. A reduced order model (RoM) approximates the dynamics of the full order model (FoM) while capturing all of the states relevant for safety. CBF synthesis can then be carried out on the RoM, and resulting behaviors can

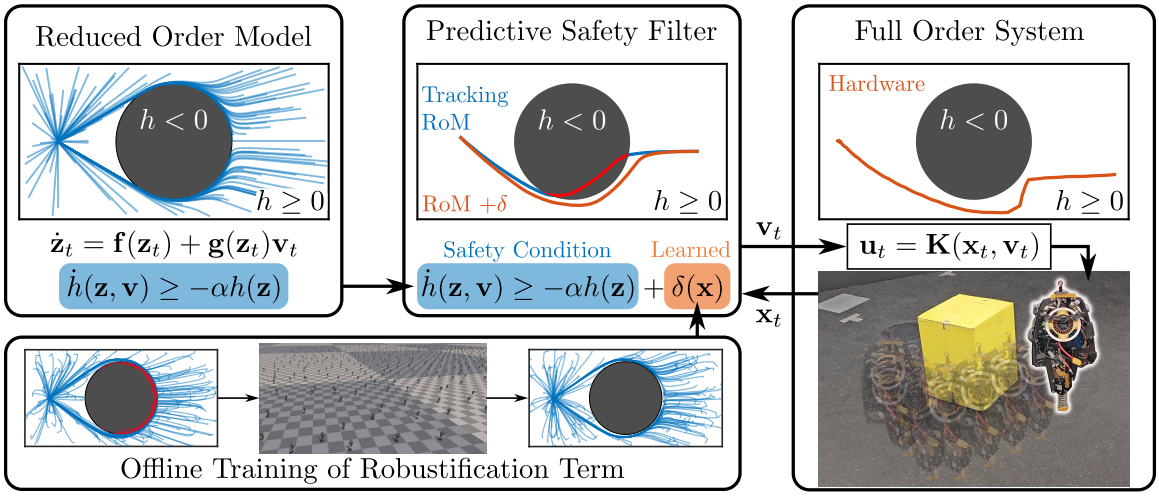

Figure 1: First, a CBF is synthesized on a RoM; tracking errors from the FoM can lead to unsafe behavior. We learn a structured robustification term $\delta(\mathbf{x})$ from massively parallel simulation, and deploy the robustified CBF on hardware to obtain safe behaviors.

be tracked by a tracking controller on the FoM. Under various assumptions regarding the tracking controller, e.g., exponential tracking (Molnar et al., 2022; Singletary et al., 2022) or the existence of a simulation function (Cohen et al., 2024a), safety can be extended to the FoM from a reduced set of initial conditions. Yet certifying the assumptions on the tracking controller is, in itself, a hard problem—and calculating the corresponding certificates is even more challenging. In practice, terms in the resulting safety filters are conservatively approximated to achieve safety empirically.

The use of RoMs to construct CBFs for safety parallels the planner-tracker paradigm, where trajectories are optimized over a RoM and then tracked by a tracking controller on the FoM. Methods in this framework focus primarily on establishing tracking error bounds, through HJ Reachability (Mitchell et al., 2005), sum of squares programming (Schweidel et al., 2022), or contraction theory (Singh et al., 2023). Once a tracking error bound has been established, tools from tube MPC (Langson et al., 2004; Compton et al., 2024) or search-based planning (Chen et al., 2021a) can be used to construct safe trajectories of the FoM. As these certification problems scale poorly with system complexity and dimensionality, we focus instead on leveraging the simplicity of the layered CBF paradigm with RoMs coupled with the power of a predictive horizon to achieve safety on the FoM.

1.1. Our Contribution:

Rather than relying upon restrictive and difficult-to-verify tracking assumptions that induce conservative behavior, we learn a robustification of the CBF synthesized on the RoM to remove safety violations over a horizon-based rollout. This robustification term, guaranteed to exist under mild assumptions on tracking error, is conditioned on the FoM state. When added to the RoM safety filter, it induces a minimal constraint back-off to ensure safety of the FoM. We both formally establish guarantees of safety, and leverage massively parallel simulation to learn this term with high accuracy. We demonstrate our method outperforming previous methods using RoMs for layered safety filtering by attaining safety under conditions where previous methods fail (due to violation of their underlying assumptions), or expanding the safe set when previous methods were successful. Finally, we demonstrate our method on a complex robotic system, a 3D hopping robot ARCHER, achieving safe navigation of cluttered environments on hardware.

2. Background

We begin by defining a framework for layered control systems consisting of a reduced order model (RoM) and full order model (FoM), interacting through a safety filter (as illustrated in Fig. 2).

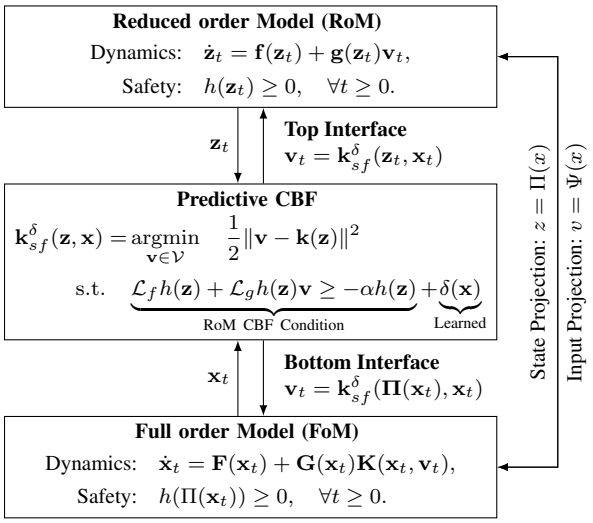

Figure 2: Layered architecture of a RoM and FoM interacting through a Predictive CBF.

Full Order Model (FoM): Consider a full order system representing the system of interest—typically these are very complex, high dimensional, and often not known. We assume that the full-order system evolves according to the nonlinear control system:

$$\dot{\mathbf{x}}_t = \mathbf{F}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x}_t)\mathbf{u}_t$$

where $\mathbf{F} : \mathcal{X} \to \mathbb{R}^N$ and $\mathbf{G} : \mathcal{X} \to \mathbb{R}^N \times \mathbb{R}^M$ are possibly *unknown* (but can be simulated), and the state $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^N$ and input $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^M$ typically have high dimension.

Reduced Order Model (RoM): Assume there is access to a reduced order model:

$$\dot{\mathbf{z}}_t = \mathbf{f}(\mathbf{z}_t) + \mathbf{g}(\mathbf{z}_t)\mathbf{v}_t \tag{1}$$

with state $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^n$, input $\mathbf{v} \in \mathcal{V} \subset \mathbb{R}^m$.

Coupling between RoM and FOM: The RoM is related to the FoM through a tracking controller—signals are sent from the RoM to the FoM and tracked by an existing (low-level) controller: $\mathbf{u} = \mathbf{K}(\mathbf{x}, \mathbf{v})$. The result is the partially closed-loop full-order dynamics:

$$\dot{\mathbf{x}}_t = \mathbf{F}_{\text{cl}}(\mathbf{x}_t, \mathbf{v}_t) := \mathbf{F}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x}_t)\mathbf{K}(\mathbf{x}_t, \mathbf{v}_t) \tag{2}$$

The RoM is related to the FoM through a projection map: $\mathbf{\Pi} : \mathcal{X} \to \mathcal{Z}$. Therefore, given a feedback controller for the RoM, $\mathbf{v} = \mathbf{k}(\mathbf{z})$, the result is the closed loop dynamics on the FoM:

$$\dot{\mathbf{x}}_t = \mathbf{F}_{\text{cl}}(\mathbf{x}_t, \mathbf{k}(\mathbf{\Pi}(\mathbf{x}_t))) = \mathbf{F}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x}_t)\mathbf{K}(\mathbf{x}_t, \mathbf{k}(\mathbf{\Pi}(\mathbf{x}_t))) \tag{3}$$

We are interested in the behavior of this closed-loop system, especially in the context of safety.

Safety: Consider a safety constraint for the RoM encoded by a smooth function $h : \mathcal{Z} \to \mathbb{R}$:

$$\mathcal{C}_{\text{RoM}} = \{\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^n \ : \ h(\mathbf{z}) \geq 0\}, \tag{4}$$

with $\partial\mathcal{C}_{\text{RoM}} = h^{-1}(0)$. For the FoM, this safety constraint takes the form:

$$\mathcal{C}_{\text{FoM}} = \{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^N \ : \ h(\mathbf{\Pi}(\mathbf{x})) \geq 0\}. \tag{5}$$

The goal is to enforce safety on the FoM through a controller applied to the RoM. That is:

Problem 1 *Given a RoM* (1)*, a closed-loop FoM* (2) *and a safe set* $\mathcal{C}_{\text{FoM}}$*, find a controller for the RoM* $\mathbf{v} = \mathbf{k}(\mathbf{z})$ *and a set* $\mathcal{S} \subseteq \mathcal{C}_{\text{FoM}}$ *such that* $\mathbf{x}_0 \in \mathcal{S}$ *implies that* $\mathbf{x}_t \in C_{\text{FoM}}$ *for all* $t \geq 0$*.*

Safety on RoMs with CBFs: There are well-established results on using RoMs to guarantee safety on the FoM. Yet these only guarantee safety under the assumption of exponential tracking *without* steady-state error—which never holds in practice. The goal of this paper is to relax this conservatism using Predictive CBFs. First, we summarize the key result for safety on FoMs using RoMs.

Assume that h is a *Control Barrier Function (CBF)* (Ames et al., 2016) for the RoM (1):

$$\sup_{\mathbf{v}\in\mathcal{V}} \left[\dot{h}(\mathbf{z},\mathbf{v}) = \underbrace{\frac{\partial h}{\partial \mathbf{z}}\Big|_{\mathbf{z}} \mathbf{f}(\mathbf{z})}_{:=\mathcal{L}_f h(\mathbf{z})} + \underbrace{\frac{\partial h}{\partial \mathbf{z}}\Big|_{\mathbf{z}} \mathbf{g}(\mathbf{z})\,\mathbf{v}}_{:=\mathcal{L}_g h(\mathbf{z})}\right] \geq -\alpha(h(\mathbf{z})), \tag{6}$$

for all $\mathbf{z}\in\mathcal{Z}$, where α is an extended class $\mathcal{K}$ function. The result is a *safety filter* for the RoM:

$$\begin{aligned} \mathbf{k}_{\text{sf}}(\mathbf{z}) = \underset{\mathbf{v}\in\mathcal{V}}{\text{argmin}} \quad & \frac{1}{2}\|\mathbf{v}-\mathbf{k}(\mathbf{z})\|^2 \\ \text{s.t.} \quad & \mathcal{L}_f h(\mathbf{z}) + \mathcal{L}_g h(\mathbf{z})\mathbf{v} \geq -\alpha h(\mathbf{z}) \end{aligned} \tag{7}$$

where for simplicity we take $\alpha(h(\mathbf{z})) = \alpha h(\mathbf{z})$ for $\alpha\in\mathbb{R}_{\geq 0}$.

This can guarantee safety for the FoM under the proper set of assumptions on the RoM and the interface between the RoM and FoM. Namely, "relative degree" and "boundedness" assumptions:

Assumption 1 *There is a projection $\boldsymbol{\Psi}:\mathcal{X}\to\mathcal{V}$ from the FoM state to the RoM inputs such that:*

$$\frac{\partial \boldsymbol{\Pi}}{\partial \mathbf{x}}\Big|_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = \mathbf{f}(\boldsymbol{\Pi}(\mathbf{x})) + \mathbf{g}(\boldsymbol{\Pi}(\mathbf{x}))\boldsymbol{\Psi}(\mathbf{x}), \qquad \frac{\partial \boldsymbol{\Pi}}{\partial \mathbf{x}}\Big|_{\mathbf{x}} \mathbf{G}(\mathbf{x}) \equiv \mathbf{0}. \tag{8}$$

Assumption 2 *$\|\mathcal{L}_g h(\boldsymbol{\Pi}(\mathbf{x}))\| \leq C_h$, for $C_h\in\mathbb{R}_{>0}$ and all $\mathbf{x}\in\mathcal{C}_{\text{FoM}}$.*

Assumption 1 and 2 are satisfied for all of the layered systems considered in this paper.

Remark 1 *Assumption 1 is often satisfied by robotic systems. In particular, in this setting, the FoM is described by: $\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{B}\mathbf{u}$ where $\mathbf{q}\in\mathcal{Q}\subset\mathbb{R}^n$ is the configuration variables (positions and angles) and $\dot{\mathbf{q}}\in T_{\mathbf{q}}\mathcal{Q}$ are the velocities and $\mathbf{u}\in\mathbb{R}^M$ the control inputs. The corresponding FoM state is therefore: $\mathbf{x} = (\mathbf{q},\dot{\mathbf{q}})\in\mathcal{X}\subset T\mathcal{Q}\subset\mathbb{R}^{2n}$. In this setting RoM are often kinematic—the most prevalent of which is the single integrator: $\dot{\mathbf{z}} = \dot{\mathbf{q}} = \mathbf{v}$. It follows that $\mathcal{Z}\subset\mathbb{R}^n$ and $\mathcal{V}\subset\mathbb{R}^n$. The projections are therefore $\boldsymbol{\Pi}:\mathcal{X}\to\mathcal{Z}$ and $\boldsymbol{\Psi}:\mathcal{X}\to\mathcal{V}$ given in coordinates by: $\boldsymbol{\Pi}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{q}$ and $\boldsymbol{\Psi}(\mathbf{q},\dot{\mathbf{q}}) = \dot{\mathbf{q}}$. It follows that $\frac{\partial \boldsymbol{\Pi}}{\partial \mathbf{x}}\Big|_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = \dot{\mathbf{q}} = \dot{\mathbf{z}}$ and $\frac{\partial \boldsymbol{\Pi}}{\partial \mathbf{x}}\mathbf{G}(\mathbf{x}) \equiv \mathbf{0}$ as desired.*

Finally, assume that the reference signal produced by the safety filter, $\mathbf{v} = \mathbf{k}_{\text{sf}}(\mathbf{z})$ can be tracked exponentially by the FoM. This is encoded by a positive definite **tracking function** $V:\mathcal{X}\to\mathbb{R}_{\geq 0}$:

$$\begin{aligned} \rho\|\boldsymbol{\Psi}(\mathbf{x}) - \mathbf{k}_{\text{sf}}(\boldsymbol{\Pi}(\mathbf{x}))\| &\leq V(\mathbf{x}), \qquad \rho\in\mathbb{R}_{>0}, \\ \dot{V}(\mathbf{x}_t) = \frac{\partial V}{\partial \mathbf{x}}\Big|_{\mathbf{x}_t} \mathbf{F}_{\text{cl}}(\mathbf{x}_t, \mathbf{k}_{\text{sf}}(\boldsymbol{\Pi}(\mathbf{x}_t))) &\leq -\lambda V(\mathbf{x}_t), \qquad \lambda\in\mathbb{R}_{>0}. \end{aligned} \tag{9}$$

The main result of (Molnar et al., 2022) establishes the safety of the FoM through safety filters on the RoM, per Problem 1, as summarized by the following (stated in the notation of this paper).

Theorem 2 *Consider a RoM (1), a closed-loop FoM (2) and a safe set $\mathcal{C}_{\text{RoM}}$ satisfying Assumptions 1 and 2. For the safety filter $\mathbf{v} = \mathbf{k}_{\text{sf}}(\mathbf{z})$ in (7), if there exists a tracking function V satisfying (9) with $\lambda\geq\alpha_x$ for any $\alpha_x > \alpha$, then the FoM is safe:*

$$\mathbf{x}_0\in\mathcal{S} := \left\{\mathbf{x}\in\mathcal{X}\colon (\alpha_x-\alpha)h(\boldsymbol{\Pi}(\mathbf{x})) - \frac{C_h}{\rho}V(\mathbf{x}) \geq 0\right\} \quad\Longrightarrow\quad \mathbf{x}_t\in\mathcal{C}_{\text{FoM}},\ \forall t\geq 0. \tag{10}$$

3. Predictive CBFs

We now present the main concept of this paper: *Predictive CBFs* (PCBFs). These account for variation between the RoM and the FoM by finding a constant to buffer the CBF condition. It does this in a minimally conservative fashion through a rollout step, i.e., through prediction. We establish two key theoretic properties of Predictive CBFs: they exist, and under reasonable assumptions guarantee safety for both the RoM and the FoM. Predictive CBFs, therefore, solve Problem 1.

3.1. Safety with Predictive CBFs

Predictive CBFs forward evaluate the FoM to find a minimal robustness term, δ, in which to buffer the safety condition for the RoM and thereby guarantee safety for the FoM.

Definition 3 *Given a RoM (1) and a FoM with tracking controller (2), h is a **Predictive CBF** if:*

(i) h is a CBF for the RoM (6),

*(ii) There exists a subset $\mathcal{C}^\delta_{\text{FoM}} \subseteq \mathcal{C}_{\text{FoM}}$ and a **predictor** $\delta : \mathcal{C}^\delta_{\text{FoM}} \subset \mathcal{X} \to \mathbb{R}_{\geq 0}$ satisfying:*

$$\begin{aligned} \delta(\mathbf{x}) = \min_{\delta \in \mathbb{R}_{\geq 0}} \quad & \delta & & \\ \text{s.t.} \quad & \dot{h}(\mathbf{\Pi}(\mathbf{x}_t), \dot{\mathbf{x}}_t) \geq -\alpha_{\mathbf{x}} h(\mathbf{\Pi}(\mathbf{x}_t)), & & \alpha_x \in \mathbb{R}_{\geq 0},\ \forall t \geq 0 \\ & \mathcal{L}_f h(\mathbf{z}) + \mathcal{L}_g h(\mathbf{z}) \mathbf{v}^\delta(\mathbf{z}) \geq -\alpha h(\mathbf{z}) + \delta, & & \mathbf{z} = \Pi(\mathbf{x}) \\ & \dot{\mathbf{x}}_t = \mathbf{F}_{\text{cl}}(\mathbf{x}_t, \mathbf{K}(\mathbf{x}_t, \mathbf{v}^\delta(\mathbf{\Pi}(\mathbf{x}_t)))), & & \mathbf{x}_0 = \mathbf{x}. \end{aligned} \tag{11}$$

*where $\mathbf{v}^\delta$ solves (7) robustified by $\boldsymbol{\delta}$. The corresponding **predictive safety filter** is given by:*

$$\begin{aligned} \mathbf{k}^\delta_{sf}(\mathbf{z}, \mathbf{x}) = \operatorname*{argmin}_{\mathbf{v} \in \mathcal{V}} \quad & \frac{1}{2} \|\mathbf{v} - \mathbf{k}(\mathbf{z})\|^2 \\ \text{s.t.} \quad & \mathcal{L}_f h(\mathbf{z}) + \mathcal{L}_g h(\mathbf{z}) \mathbf{v} \geq -\alpha h(\mathbf{z}) + \delta(\mathbf{x}). \end{aligned} \tag{12}$$

Remark 4 *The name "Predictive CBF" has appeared in the literature in different forms than considered here, notably in Breeden and Panagou (2022) and Wabersich and Zeilinger (2022). In the former, trajectories are rolled out and the impact of the barrier function is considered. In the latter, discrete time CBFs are considered in a model predictive control setting. The notion considered here differs in that (1) it is considered for continuous-time systems, (2) the predictive nature of the CBF is in the CBF condition itself via $\delta(\mathbf{x})$, and (3) we consider Predictive CBFs in the setting of layered control systems to use prediction to encapsulate the difference, δ, between the RoM and FoM.*

The closest method, philosophically, to Predictive CBFs is Projection to State Safety (PSS) (Taylor et al., 2020), which adds a delta term to the CBF condition to ensure approximate safety—input to state safety (ISSf). Yet this relied on learning δ from past trajectories, while PCBFs determine δ from future behaviors. Formally, this guarantees safety rather than approximate safety (ISSf).

Lastly, note that in situations where tracking assumptions of previous works, i.e. Molnar et al. (2022), are satisfied, this algorithm returns $\delta = 0$, but will potentially expand the FoM safe set.

Importantly, it follows by definition that Predictive CBFs formally guarantee safety on both the reduced order and full order models simultaneously solving Problem 1. Namely, the satisfaction of $\dot{h}(\mathbf{\Pi}(\mathbf{x}_t), \dot{\mathbf{x}}_t) \geq -\alpha_{\mathbf{x}} h(\mathbf{\Pi}(\mathbf{x}_t))$ in (11) implies $h(\mathbf{\Pi}(\mathbf{x}_t)) \geq 0$, cf. (Ames et al., 2016), yielding:

Theorem 5 (Predictive CBFs $\implies$ Safety) *If h is a Predictive CBF, then the FoM is safe:*

$$\dot{\mathbf{x}}_t = \mathbf{F}_{\text{cl}}(\mathbf{x}_t, \mathbf{K}(\mathbf{x}_t, \mathbf{k}^\delta_{sf}(\mathbf{z}_t, \mathbf{x}_t)) \quad \implies \quad \text{if } \mathbf{x}_0 \in \mathcal{C}^\delta_{\text{FoM}} \quad \text{then} \quad \mathbf{x}_t \in \mathcal{C}_{\text{FoM}} \quad \forall t \geq 0.$$

3.2. Existence of Predictive CBFs

If we make a few key assumptions, it is possible to show that Predictive CBFs are well-defined. These are analogous to the assumptions needed for Theorem 2, except that the addition of the δ term allows us to relax the tracking assumption to bounded tracking—this is critical in real-world applications where exact tracking is never achieved.

Consider the reference signal produced by the predictive safety filter, $\mathbf{v} = \mathbf{k}_{\text{sf}}^{\delta}(\mathbf{z}, \mathbf{x})$ given in (12). Assume a positive definite δ**-bounded tracking function** $V_\delta : \mathcal{X} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfying:

$$\begin{aligned} \rho \|\mathbf{\Psi}(\mathbf{x}) - \mathbf{k}_{\text{sf}}^{\delta}(\mathbf{\Pi}(\mathbf{x}), \mathbf{x})\| &\leq V_\delta(\mathbf{x}) && \rho \in \mathbb{R}_{>0} \\ \dot{V}_\delta(\mathbf{x}_t) &\leq -\lambda V_\delta(\mathbf{x}_t) + \mu, && \mu \in \mathbb{R}_{\geq 0},\ \lambda \in \mathbb{R}_{>0}. \end{aligned} \tag{13}$$

where μ is a steady state tracking error. Note that when $\mu = 0$, $V_{\delta(\mathbf{x})\equiv 0}$ is a tracking function as in (9). Bounded tracking functions, therefore, capture the more challenging case of tracking only to an error tube (dictated by μ). It is the addition of δ in the safety filter (12) that guarantees safety.

Theorem 6 (Existence of Predictive CBFs) *Consider a RoM (1), a closed-loop FoM (2) and a safe set $\mathcal{C}_{\text{RoM}}$ satisfying Assumptions 1 and 2. Let $\delta_0(\mathbf{x}) \equiv \delta_0$ for $\delta_0 \in \mathbb{R}_{\geq 0}$. If there exists a δ_0-bounded tracking function, V_{δ_0}, with $\lambda \geq \alpha_x$ for any $\alpha_x > \alpha$, then the FoM is safe:*

$$\begin{aligned} \delta_0 &\geq \tfrac{C_h \mu}{\alpha_x \rho} \\ \mathbf{x}_0 \in \mathcal{S}_{\delta_0} &:= \mathcal{C}_{\text{FoM}} \cap \left\{ \mathbf{x} \in \mathcal{X} \colon (\alpha_x - \alpha) h(\mathbf{\Pi}(\mathbf{x})) + \delta_0 - \tfrac{C_h}{\rho} V_{\delta_0}(\mathbf{x}) \geq 0 \right\} \end{aligned} \implies \begin{aligned} &\mathbf{x}_t \in \mathcal{C}_{\text{FoM}} \\ &\forall t \geq 0. \end{aligned}$$

Moreover, the set $\mathcal{S}_{\delta_0} \subset \mathcal{C}_{\text{FoM}}$ and the constant function $\delta_0 : \mathcal{S}_{\delta_0} \to \mathbb{R}_{\geq 0}$ satisfies $\delta(\mathbf{x}) \leq \delta_0$ for all $\mathbf{x} \in \mathcal{S}_{\delta_0}$. Therefore, $\delta : \mathcal{C}_{\text{FoM}}^{\delta} \to \mathbb{R}_{\geq 0}$ exists with $\mathcal{S}_{\delta_0} \subset \mathcal{C}_{\text{FoM}}^{\delta}$.

The proof of this theorem, provided in the appendix, constructed an approximation to $\delta(\mathbf{x})$ that did not require direct knowledge of the FoM—only the corresponding performance of the tracking controller for that system via V_δ. Determining this is not easy, but can be estimated in practice. This motivates the idea of directly learning δ. Additionally, it will be seen that learning δ enables us to ensure safety even when all of the assumptions of the theorem do not hold, i.e., when $\lambda < \alpha_x$.

Simulation Results. To demonstrate improvements of the Predictive CBF over previous methods, we will consider a double integrator tracking velocity commands synthesized from single integrator dynamics. The RoM, FoM and tracking controller, i.e., the key elements in Fig. 2, are therefore:

$$\dot{\mathbf{z}} = \mathbf{v} \qquad \dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{u} \qquad \begin{aligned} \mathbf{\Pi}(\mathbf{x}) &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{x} \\ \mathbf{\Psi}(\mathbf{x}) &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x} \end{aligned} \qquad \mathbf{K}(\mathbf{x}, \mathbf{v}) = -k_v \left(\mathbf{\Psi}(\mathbf{x}) - \mathbf{v} \right) \tag{14}$$

Example 1 (Single/Double Integrator) To examine properties of $\delta(\mathbf{x})$, consider the scalar RoM case of (14), i.e., $z \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^2$, i.e., the RoM is $\dot{z} = v$ and the FoM is $\dot{x}_1 = x_2$ and $\dot{x}_2 = u = -k_v(x_2 - v)$. The safety constraint is to maintain nonnegative position, $h(z) = z$, i.e., $\mathcal{C}_{\text{RoM}} = \{z \in \mathbb{R} : h(z) \geq 0\}$ and $\mathcal{C}_{\text{FoM}} = \{\mathbf{x} \in \mathbb{R}^2 : h(\mathbf{\Pi}(\mathbf{x})) = x_1 \geq 0\}$. Finally, fix the desired controller in the safety filter (12) to be $k(z) = -\frac{1}{2}$, which attempts to drive the single integrator into the unsafe region. Following Theorem 6, we instantiate a predictive safety filter using the constant function $\delta_0(\mathbf{x}) \equiv \delta_0$ and examine the properties of the closed-loop system for different values of δ_0 (Fig. 3, top row). Here, when $\delta_0 = 0$ (left), the conditions of Theorem 6 are not satisfied, leading to safety violations even when starting within $\mathcal{S}_{\delta_0}$. On the other hand, increasing δ_0 to 1 ensures the conditions of this result are satisfied, leading to safety for initial conditions within $\mathcal{S}_{\delta_0}$ (middle).

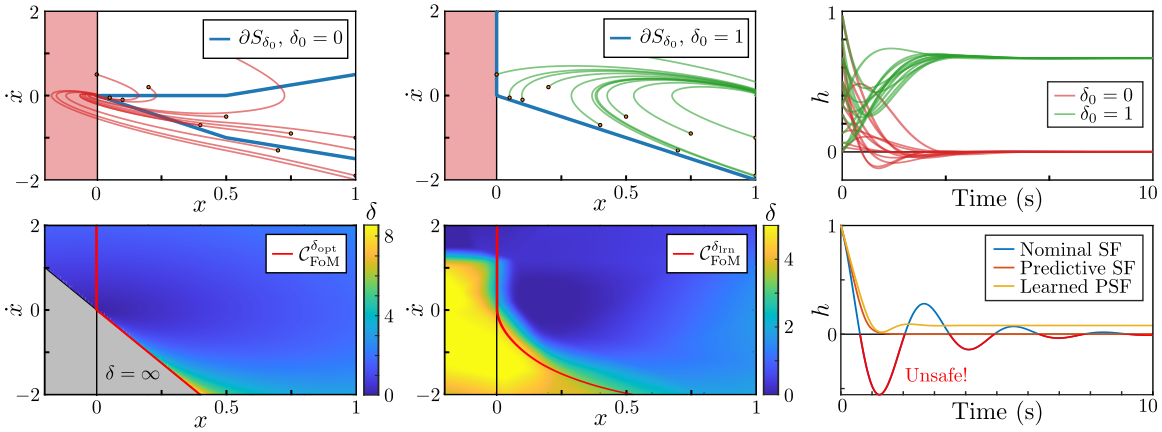

Figure 3: **Top:** Comparison of the set $\mathcal{S}_{\delta_0}$ from Theorem 6 for different values of δ_0 and trajectories of the closed-loop FoM in Example 1. **Bottom:** Comparison between the optimized (left) and learned (middle) values of $\delta(\mathbf{x})$, as well as the corresponding $\mathcal{C}^{\delta}_{\text{FoM}}$. (Right) compares the performance of these Predictive CBFs to the nominal CBF without δ.

4. Learning Predictive CBFs

Predictive CBFs are amiable to implementation, as facilitated through learning. We first consider the algorithmic computation of PCBFs through simulation roll-outs. While this effectively approximates PCBFs, the computational overhead prevents their rapid evaluation. Learning the δ term circumvents this limitation. We demonstrate the performance of the Predictive CBF and its learned counterpart in simulation in this section, followed by experimentally in the following section.

Computing PCBFs. The optimization problem (11) while difficult to solve exactly, lends itself to a receding horizon implementation. Here, we iteratively approximate solutions to (11) via simulated rollouts to estimate $\delta(\mathbf{x})$ online. In particular, for $\mathbf{x}_0 \in \mathcal{X}$, CBF h, RoM controller $\mathbf{k}$, and tracking controller $\mathbf{K}$, define $\delta^i(\mathbf{x}_0)$ through the following iterative algorithm:

$$\delta^{i+1}(\mathbf{x}_0) := \max\left(0, \delta^i(\mathbf{x}_0) - \eta e_{\delta_i}(\mathbf{x}_0)\right) \qquad \begin{aligned} \delta^0(\mathbf{x}_0) &= 0, \quad i \in \{0, \dots, \bar{N}\}, \\ \mathbf{x}'_t &\approx \mathbf{x}_0 + \textstyle\int_0^t \mathbf{F}_{\text{cl}}(\mathbf{x}'_\tau, \mathbf{k}^{\delta_i}_{sf}(\mathbf{\Pi}(\mathbf{x}'_\tau)))d\tau, \\ e_{\delta_i}(\mathbf{x}_0) &= \min_{t\in[0,T]} \dot{h}(\mathbf{x}'_t) + \alpha h(\mathbf{\Pi}(\mathbf{x}'_t)) \end{aligned} \tag{15}$$

where $T \in \mathbb{R}_{>0}$ is the rollout horizon, $\eta \in \mathbb{R}_{>0}$ is a step size, $\bar{N} \in \mathbb{N}_{\geq 0}$ is the number of iterations, and $\mathbf{x}'_t$ is an approximated (simulated) solution. At each iteration, i, we roll out a trajectory under our current value of $\delta = \delta^i$. We compute the maximum violation of the barrier condition, e, and then take a proportional step on δ^{i+1} to reduce this violation. This algorithm is iterated until convergence, approximating solutions to (11). Online, running to convergence is slow; we can instead take a real-time iterations approach (Diehl et al., 2005, 2002), where we take only one step on the optimization problem before updating the system state (and tackling a *new* optimization problem). Assuming safety violation does not change too rapidly, we can achieve convergence to approximately optimal δ as we progress through time; this may be enough to realize safety practically for complex systems online, as discussed in Section 5.

Algorithm 1 Predictive CBF Learning

input : CBF h, RoM controller $\mathbf{k}$, Tracking controller $\mathbf{K}$, randomly initialized NN parameters $\boldsymbol{\theta}_0$
output: CBF Robustification term $\delta_{\boldsymbol{\theta}_{N_{\text{epochs}}}}$
for $j = 1, \dots, N_{\text{epochs}}$ **do**

$$\mathcal{X}_j \leftarrow \left\{ \mathbf{x}_t^i : \dot{\mathbf{x}}_t^i = \mathbf{F}_{\text{cl}}^{\delta_{\boldsymbol{\theta}_{j-1}}}(\mathbf{x}_t^i) \right\}_{i=1}^{N} \quad \texttt{// Collect dataset under } \delta_{\boldsymbol{\theta}}$$

$$\mathcal{E}_j \leftarrow \left\{ e_t^i : e_t^i = \min_{\tau \in [t,T]} \dot{h}(\mathbf{x}_t^i) + \alpha h(\boldsymbol{\Pi}(\mathbf{x}_t^i)) \right\}_{i=1}^{N} \quad \texttt{// Compute violation}$$

$$\mathcal{D}_j \leftarrow \left\{ \hat{\delta}_t^i : \hat{\delta}_t^i = \delta_{\boldsymbol{\theta}_{j-1}}(\mathbf{x}_t^i) - \eta_j e_t^i \right\}_{i=1}^{N} \quad \texttt{// Compute new targets}$$

$$\boldsymbol{\theta}_j = \text{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{X}_j, \mathcal{D}_j) \quad \texttt{// Train NN on new targets}$$

end

Learning PCBFs. In practice, solving the optimization (11) requires several roll-outs of the system dynamics. Computation time for these roll-outs may be a limiting factor when executing this method online; additionally, accuracy of the simulator may introduce some brittleness, as differences between the simulator dynamics and hardware could lead to safety violations. We aim to tackle both of these issues by proposing a learning algorithm to approximate $\delta(\mathbf{x})$ from simulation data, where we can utilize domain randomization to facilitate transfer to hardware.

The learning algorithm, outlined in Algorithm 1, takes in the RoM CBF and nominal controller, and the tracking controller; it trains the CBF robustifcation term $\delta_{\boldsymbol{\theta}}$, a neural network with parameters $\boldsymbol{\theta}$. For each iteration of the algorithm, the algorithm first collects a set of rollouts, $\mathcal{X}_j$, under the current robustification term, according to the dynamics: $\mathbf{F}_{\text{cl}}^{\delta}(\mathbf{x}_t) = \mathbf{F}_{\text{cl}}(\mathbf{x}_t, \mathbf{k}_{\text{sf}}^{\delta}(\boldsymbol{\Pi}(\mathbf{x}_t)))$. Next, the maximum safety violation (or margin) over a horizon of length T is computed for each point along the trajectory. New targets for δ, denoted $\hat{\delta}$, are computed by subtracting the violation from the current approximation of δ, where η_j is a step size parameter. Finally, a new neural network is fit to set of targets $\hat{\delta}$ by minimizing the loss function (16), where $L_{c,\sigma}$ is the check loss function (Koenker and Bassett Jr, 1978), parameterized by $\sigma \in (0, 1)$. Choosing $\sigma > 0.5$ will more heavily penalize cases where $x \leq y$; in our application, this will more heavily penalize underestimates of $\hat{\delta}$, as underestimates may not guarantee safety (while overestimates will). This algorithm is run iteratively, until convergence (or a maximum number of iterations).

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{X}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\substack{\mathbf{x}_t^i \in \mathcal{X} \\ \hat{\delta}_t^i \in \mathcal{D}}} L_{c,\sigma} \left(\delta_{\boldsymbol{\theta}}(\mathbf{x}_t^i), \text{clip}(\hat{\delta}_t^i, 0, \delta_{\max}) \right)$$

$$L_{c,\sigma}(x, y) = \begin{cases} \sigma(y - x) & x \leq y \\ (1 - \sigma)(x - y) & x > y \end{cases} \tag{16}$$

We implement this algorithm in pyTorch (Paszke et al., 2017) on an NVIDIA GTX 4080 GPU, training Relu networks with two layers of 128 nodes. Toy dynamics are implemented in pyTorch, while complex robotic systems are simulated on GPU in IsaacGym (Makoviychuk et al., 2021) for massively parallelized data collection.

Example 2 (Single/Double Integrator) Consider again the setup in Example 1. In this setting, Fig. 3 highlights the differences between the optimized and learned values of $\delta(\mathbf{x})$, as well as a comparison between evolutions of the system under the various safety filters. In the left pane, we see the result of solving the optimization problem (11). The gray regions indicate the optimization problem being unsolvable, and the corresponding safe set is outlined in red. Critically, note that the boundaries with the region where $\delta(\mathbf{x}) = \infty$ are quite sharp; to make the learning problem tractable, we cap the target δ values at 5. Incapable of learning the sharp boundary of the optimization problem, the learned $\delta_{\boldsymbol{\theta}}(\mathbf{x})$ is slightly conservative around the boundary (see right pane).

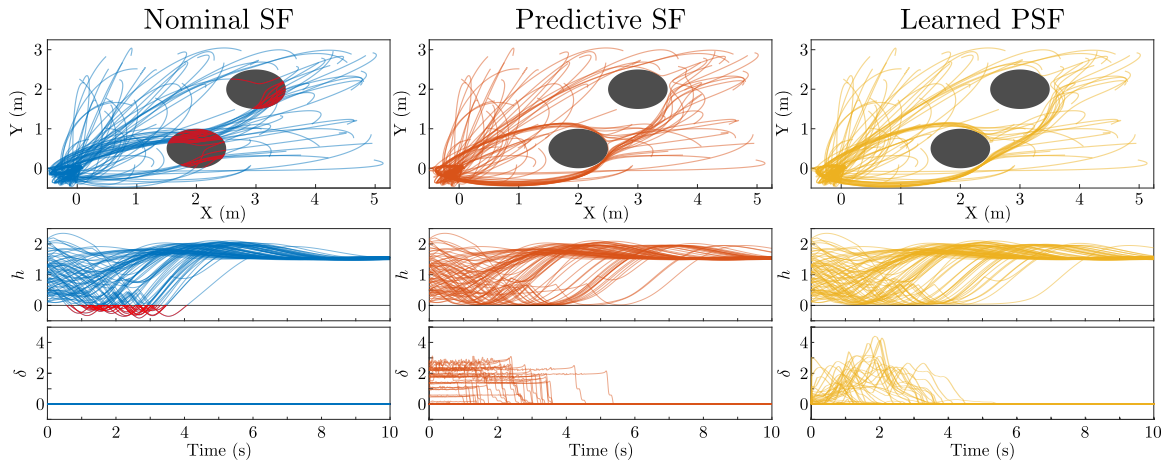

Figure 4: Comparison between the performance of the Predictive SF and Learned PSF to the Nominal SF. Both the Predictive SF and Learned PSF ensure safety for the FoM, while the Nominal SF violates the safety constraint.

Example 3 (Obstacle Avoidance) In a more realistic example, we now take the two-dimensional version of (14) with $\mathbf{z} \in \mathbb{R}^2, \mathbf{x} \in \mathbb{R}^4$, with the goal performing collision avoidance—following from (Singletary et al., 2021). The nominal controller is a saturated proportional controller and the nominal barrier function is the distance to the closest obstacle, indexed by i:

$$\mathbf{k}(\mathbf{z}) = -\min\left\{\frac{v_{\max}}{k_p\|\mathbf{z}\|}, 1\right\} k_p\mathbf{z} \qquad h(\mathbf{z}) = \min_i \{\|\mathbf{z} - c_i\| - r_i\}_{i=1}^{N_{\text{obs}}}$$

Note that this barrier function can be smoothed (Glotfelter et al., 2017); we use the non-smooth version as the safe set is more intuitive to characterize and visualize. Fig. 4 displays the difference between the Nominal SF, Predictive SF, and Learned PSF. Because the used tracking controller is slower than the rate the system is allowed to approach the barrier, $\lambda < \alpha$, the assumptions of the nominal safety filter are not satisfied, and the FoM violates the safety constraint. Despite differing δ profiles, the optimized and learned trajectories are exceptionally similar; the learned version has a slightly smoother δ profile, but both remain safe over the entire trajectory.

5. Application to Hardware: 3D Hopping

We apply the proposed method to the 3D hopping robot, ARCHER, shown in Fig. 5. The system contains $N = 16$ states $\mathbf{x} = [\mathbf{p}^\top \ \mathbf{q}^\top \ \mathbf{v}^\top \ \boldsymbol{\omega}^\top \ \dot{\boldsymbol{\theta}}^\top]^\top$, where $\mathbf{p} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{S}^3$ the position and orientation, $\mathbf{v} \in \mathbb{R}^3, \boldsymbol{\omega} \in \mathbb{R}^3$ the linear and angular velocities, and $\dot{\boldsymbol{\theta}} \in \mathbb{R}^3$ the flywheels velocities. The control input is torque applied to each of the three flywheels. The foot spring is compressed to a preset distance during the flight phase, and released during the ground phase to maintain constant hop height. More hardware details can be found (Csomay-Shanklin et al., 2023). The system dynamics are simulated in IsaacGym (Makoviychuk et al., 2021) for data generation and learning.

To accomplish a collision avoidance task, we design a CBF on the 2D single integrator, and track the resulting velocity commands. Tracking controllers for ARCHER have been studied in previous works (Csomay-Shanklin et al., 2024, 2023). Here we will use the same tracking controller as (Compton et al., 2024), where a Raibert Heuristic (Raibert et al., 1984) determines desired impact orientations, and these orientations are tracked by a geometrically consistent PD controller.

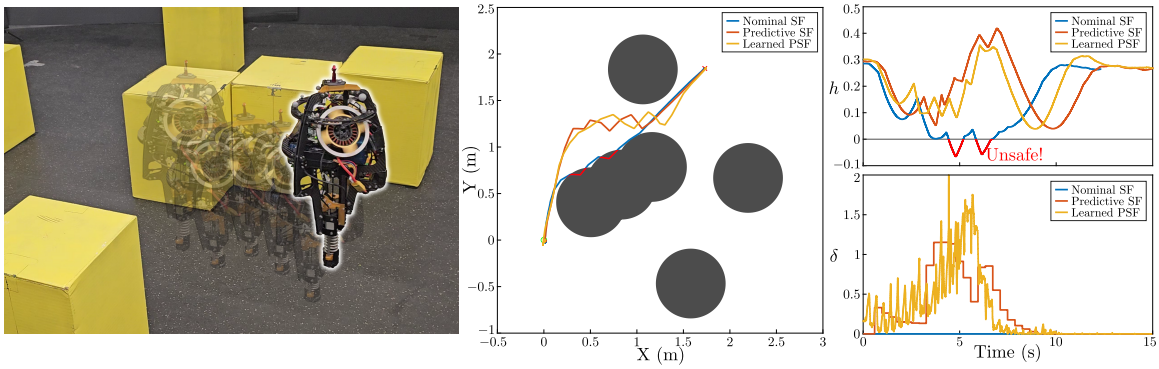

Figure 5: The 3D hopping robot ARCHER (left) navigates a cluttered environment using each of the Nominal SF, Predictive SF and Learned PSF. (Middle) The trajectories of the hopper in space; (Right) values of h and δ plotted over time for each approach. The Nominal SF is unsafe, but both the Predictive SF and Learned PSF maintain safety.

Using the same nominal controller and barrier function as in the double integrator example, we deploy both real-time iterations of (15) (Predictive SF) and the Learned PCBF on hardware. To facilitate transfer to hardware (which has worse tracking performance than simulation, primarily due to uncertainty in the position of the center of mass, to which the system is highly sensitive), we domain randomize the robot's center of mass, and use the check loss to upper bound the distribution of resulting behaviors. A comparison with the nominal controller, are shown in Fig. 5. Due to unaccounted for tracking errors, the nominal controller violates the safety constraint multiple times on its trajectory to the origin. However, both the Predictive SF and the Learned PSF maintain safety over the entire trial. Experiment video is available (Compton, 2024).

Due to the computationally intensive rollout, the online optimization approach runs at only around 5Hz on hardware, as seen in the bottom right of Fig. 5. At this rate, it does not capture how safety prediction varies over the course of a hop, and is reliant upon some native robustness of the CBF condition to enforce safety, as the convergence of δ to its optimal value is quite slow. On the other hand, the learned algorithm is evaluated at 100Hz on hardware; δ for the Learned PSF captures variation of safety over the course of a single hop, where delta increases during flight, as the hopper is unable to change its velocity until the next contact. During contact, δ drops dramatically as the hopper's velocity corrects toward the safe desired velocity. Because of its faster evaluation, the learning approach scales to complex systems and tracking controllers or more dynamic environments, where the computational cost of rollouts is high.

6. Conclusion

Synthesis of control barrier functions to ensure safety of complex nonlinear systems is difficult. As safety often depends only on a subset of the states, reduced order models are often used to synthesize safe behaviors, and the behavior of the reduced order model is then tracked on the full order system. We propose a novel, prediction based CBF to account for a very general class of tracking abilities; we robustify the safety filter applied to the RoM by a minimal amount to ensure safety of the FoM over a simulated horizon. As this optimization problem solving for the robustification factor, $\delta(\mathbf{x})$, can be prohibitively expensive to run online in real time, we instead learn this term. This method achieves safe navigation through cluttered environments on the 3D hopping robot ARCHER.

Acknowledgments

We would like to thank Noel Csomay-Shanklin for his help with experiments. This research is supported in part by Boeing, and NSF CPS Award #1932091.

References

Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE conference on decision and control*, pages 6271–6278. IEEE, 2014.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2016.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

Joseph Breeden and Dimitra Panagou. Predictive control barrier functions for online safety critical control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 924–931. IEEE, 2022.

Mo Chen, Sylvia L Herbert, Haimin Hu, Ye Pu, Jaime Fernandez Fisac, Somil Bansal, SooJean Han, and Claire J Tomlin. Fastrack: a modular framework for real-time motion planning and guaranteed safe tracking. *IEEE Transactions on Automatic Control*, 66(12):5861–5876, 2021a.

Yuxiao Chen, Mrdjan Jankovic, Mario Santillo, and Aaron D Ames. Backup control barrier functions: Formulation and comparative study. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6835–6841. IEEE, 2021b.

Max H Cohen, Noel Csomay-Shanklin, William D Compton, Tamas G Molnar, and Aaron D Ames. Safety-critical controller synthesis with reduced-order models. *arXiv preprint arXiv:2411.16479*, 2024a.

Max H Cohen, Tamas G Molnar, and Aaron D Ames. Safety-critical control for autonomous systems: Control barrier functions via reduced-order models. *Annual Reviews in Control*, 57:100947, 2024b.

William D Compton. Experiment Compilation: Predictive Control Barrier Functions. YouTube, 2024. URL https://youtu.be/6pY7T6yucBs.

William D Compton, Noel Csomay-Shanklin, Cole Johnson, and Aaron D Ames. Dynamic tube mpc: Learning tube dynamics with massively parallel simulation for robust safety in practice. *arXiv preprint arXiv:2411.15350*, 2024.

Noel Csomay-Shanklin, Victor D. Dorobantu, and Aaron D. Ames. Nonlinear model predictive control of a 3d hopping robot: Leveraging lie group integrators for dynamically stable behaviors. *Proceedings - IEEE International Conference on Robotics and Automation*, 2023.

Noel Csomay-Shanklin, William D Compton, Ivan Dario Jimenez Rodriguez, Eric R Ambrose, Yisong Yue, and Aaron D Ames. Robust agility via learned zero dynamics policies. *arXiv preprint arXiv:2409.06125*, 2024.

Moritz Diehl, Rolf Findeisen, Stefan Schwarzkopf, Ilknur Uslu, Frank Allgöwer, Hans Georg Bock, Ernst-Dieter Gilles, and Johannes P Schlöder. An efficient algorithm for nonlinear model predictive control of large-scale systems part i. *Automation technology*, 2002.

Moritz Diehl, Hans Georg Bock, and Johannes P Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on control and optimization*, 43(5): 1714–1736, 2005.

Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315, 2017.

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

Wilbur Langson, Ioannis Chryssochoos, SV Raković, and David Q Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

V Makoviychuk, L Wawrzyniak, Y Guo, M Lu, K Storey, M Macklin, D Hoeller, N Rudin, A Allshire, A Handa, and G State. Isaac gym: High performance gpu based physics simu-lation for robot learning. *Neural Information Processing Systems*, 2021.

Nikolai Matni, Aaron D Ames, and John C Doyle. A quantitative framework for layered multirate control: Toward a theory of control architecture. *IEEE Control Systems Magazine*, 44(3):52–94, 2024.

Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

Tamas G. Molnar, Ryan K. Cosner, Andrew W. Singletary, Wyatt Ubellacker, and Aaron D. Ames. Model-free safety-critical control for robotic systems. *IEEE Robotics Automation Letters*, 7(2): 944–951, April 2022. ISSN 2377-3774.

Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*, pages 322–328. IEEE, 2016.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *Neural Information Processing*, 2017.

Marc H Raibert, H Benjamin Brown Jr, and Michael Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 3(2):75–92, 1984.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3717–3724. IEEE, 2020.

Katherine S Schweidel, He Yin, Stanley W Smith, and Murat Arcak. Safe-by-design planner–tracker synthesis with a hierarchy of system models. *Annual Reviews in Control*, 53:138–146, 2022.

Sumeet Singh, Benoit Landry, Anirudha Majumdar, Jean-Jacques Slotine, and Marco Pavone. Robust feedback motion planning via contraction theory. *The International Journal of Robotics Research*, 42(9):655–688, 2023.

Andrew Singletary, Karl Klingebiel, Joseph Bourne, Andrew Browning, Phil Tokumaru, and Aaron Ames. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8129–8136. IEEE, 2021.

Andrew Singletary, William Guffey, Tamas G Molnar, Ryan Sinnet, and Aaron D Ames. Safety-critical manipulation for collision-free food preparation. *IEEE Robotics and Automation Letters*, 7(4):10954–10961, 2022.

Andrew J Taylor, Andrew Singletary, Yisong Yue, and Aaron D Ames. A control barrier perspective on episodic learning via projection-to-state safety. *IEEE Control Systems Letters*, 5(3):1019–1024, 2020.

Sander Tonkens and Sylvia Herbert. Refining control barrier functions through hamilton-jacobi reachability. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13355–13362. IEEE, 2022.

Kim P Wabersich and Melanie N Zeilinger. Predictive control barrier functions: Enhanced safety mechanisms for learning-based control. *IEEE Transactions on Automatic Control*, 68(5):2638–2651, 2022.

Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

Li Wang, Aaron D Ames, and Magnus Egerstedt. Safe certificate-based maneuvers for teams of quadrotors using differential flatness. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3293–3298. IEEE, 2017.

Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.