Safe Exploration in Model-based Reinforcement Learning using Control Barrier Functions ⋆

Max H. Cohen [a], Calin Belta [a]

[a]*Department of Mechanical Engineering, Boston University, 110 Cummington Mall, Boston, MA, USA*

Abstract

This paper develops a model-based reinforcement learning (MBRL) framework for learning online the value function of an infinite-horizon optimal control problem while obeying safety constraints expressed as control barrier functions (CBFs). Our approach is facilitated by the development of a novel class of CBFs, termed Lyapunov-like CBFs (LCBFs), that retain the beneficial properties of CBFs for developing minimally-invasive safe control policies while also possessing desirable Lyapunov-like qualities such as positive semi-definiteness. We show how these LCBFs can be used to augment a learning-based control policy to guarantee safety and then leverage this approach to develop a safe exploration framework in a MBRL setting. We demonstrate that our approach can handle more general safety constraints than comparative methods via numerical examples.

Key words: Adaptive control; Control barrier functions; Reinforcement learning.

1 Introduction

Learning-based control methods, such as reinforcement learning (RL), have shown success in solving complex control problems. Although such methods have demonstrated success in simulated environments, the lack of provable safety guarantees has limited the application of these techniques to real-world safety-critical systems. To address these challenges, the field of *safe learning-based control* has emerged with the objective of ensuring the safety of learning-enabled systems. The concept of safety in control systems is often formalized through the forward invariance [3] of designated safe sets. Popular approaches to safe learning include reachability analysis [10], model predictive control [12,13,23], and barrier functions (BFs) [4,33]. Similar to how Lyapunov functions are used to study the stability of equilibrium points without computing a system's solution, BFs [1,2,28,34, 37,38] allow for one to study the invariance of sets without explicitly computing a system's reachable set. Motivated by the need for general tools that facilitate the design of controllers with safety guarantees, the concept of a control BF (CBF) was introduced in [1,2,37], and has been successfully applied to systems such as autonomous vehicles and multi-agent systems [1].

In this paper we unite CBFs and model-based reinforcement learning (MBRL) to develop a safe exploration framework for jointly learning online the dynamics of an uncertain control affine system and the optimal value function/policy of an infinite-horizon optimal stabilization problem. Our approach is facilitated by the class of Lyapunov-like barrier functions introduced in [28], which are used to develop a partially model-free robust safeguarding controller that can be combined with an arbitrary learning-based control policy to guarantee safety. This safeguarding controller is leveraged to build upon the MBRL framework from [16–18] to develop a safe exploration scheme in which the value function is learned online via "simulation of experience." In this approach, the tension between exploration and safety is addressed by simulating on-the-fly an approximated model of the system at unexplored points in the state space to generate data for learning the value function without risking safety violation of the original system. We provide proofs of convergence of this approximation scheme and demonstrate numerically the advantages of our approach over related safe online RL approaches.

Related work The online RL method considered herein is rooted in the seminal work of [35], where techniques from adaptive control [21] were used to solve online an unconstrained infinite-horizon optimal control problem for a nonlinear system with known dynamics. This approach was quickly extended to uncertain sys-

---

⋆ This paper was not presented at any IFAC meeting. Corresponding author M. H. Cohen.

Email addresses: maxcohen@bu.edu (Max H. Cohen), cbelta@bu.edu (Calin Belta).

tems using model-free and model-based RL methods (see [20, 22] for surveys of model-free methods and [18] for a monograph of model-based methods). Although various extensions have been proposed over the past decade, these techniques have been limited to unconstrained problems or those with actuator constraints [7, 36]. More recently, these RL techniques have begun to consider *safety constraints* by incorporating BF-based terms in the problem's cost function [6, 11, 26, 27, 39]. The approaches from [11, 26, 39] rely on a barrier Lyapunov function (BLF) [34] based system transformation to map a constrained optimal control problem to an unconstrained one, which can then be solved using existing approaches. However, these techniques are limited to rectangular state constraints (i.e., the safe set is a hyperrectangle), which do not encompass the complex safety specifications, such as collision avoidance, encountered in applications such as robotics. These BF-based techniques were generalized to safe sets defined by the zero-superlevel set of a general continuously differentiable function in [6, 27] by including a CBF-based term in the problem's cost function, which was inspired by the motion planning framework from [9]. However, as argued in [26], the approaches from [6, 9, 27] are facilitated by the strong assumption that the resulting value function is continuously differentiable - a condition that may fail to hold for various systems and safe sets. Although the aforementioned approaches have demonstrated success in practice, an important limitation faced by [6, 9, 11, 26, 27, 39] in the context of safety-critical control is the use of the resulting value function, which acts as a BLF, as a safety certificate. Since the ultimate objective of these approaches is to learn the value function/safety certificate, safety and learning become tightly coupled - safety guarantees are conditioned upon convergence of the RL algorithm, which is predicated on excitation conditions that cannot be verified in practice. It is the aim of this paper to address this limitation by *decoupling* learning from safety, with the latter being guaranteed at all times independent of any conditions associated with learning. Since the value function of the class of optimal control problems considered herein is a control Lyapunov function (CLF) [31, Ch. 8.5], our method can be seen as an approach to safely learn an optimal CLF for an uncertain system using data from a single trajectory. Compared to traditional CBF-based approaches that unite CLFs and CBFs to achieve dual objectives of stability and safety, this implies one need only to construct a valid CBF to achieve stability and safety, with the additional benefit of accounting for partially uncertain dynamics.

Contributions The contributions of this paper are threefold. First, we introduce a new class of CBFs, termed Lyapunov-like CBFs (LCBFs), inspired by the Lyapunov-like barrier functions introduced in [28], that retain the important properties of CBFs for making

safety guarantees while possessing desirable Lyapunov-like qualities that facilitate the development of safe and stabilizing controllers. We illustrate how the gradient of this LCBF can be used to construct a safeguarding controller that shields a performance-driven RL policy in a minimally invasive fashion to guarantee safety. Second, we extend the MBRL architecture from [16] to develop a safe exploration framework in which the value function and optimal policy of an unconstrained optimal control problem are *safely learned* online for an uncertain nonlinear system. The fundamental distinction between our approach and those of related works [6, 9, 11, 26, 27, 39] is that the aforementioned methods aim to *learn a safe policy*, whereas our safe exploration framework allows for *safely learning a performance-driven policy* - an approach more aligned with works such as [4, 10]. Although, at a high-level, our technical approach is similar to those of [16, 17], the approach in this paper presents a departure from those in [16, 17] in that *different* policies are used for exploration (learned policy) and deployment on the original system (safe policy), which complicates the resulting convergence analysis. Third, we present numerical examples demonstrating the improved safety guarantees of the proposed MBRL method compared to those that enforce safety through the minimization of a suitably constructed cost functional. In an additional numerical example, we illustrate empirically that our approach allows for simultaneous stabilization and obstacle avoidance via dynamic time-varying feedback.

Notation A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be a class $\mathcal{K}$ function, denoted by $\alpha \in \mathcal{K}$, if it is strictly increasing and $\alpha(0) = 0$. The derivative of a continuously differentiable function $f(x, y)$ with respect to its first argument is denoted by $\nabla f(x, y) = \frac{\partial f(x,y)}{\partial x}$ and is interpreted as a row vector. The operator $\| \cdot \|$ denotes the standard Euclidean norm, $I_n \in \mathbb{R}^{n \times n}$ denotes an $n \times n$ identity matrix, and $\lambda_{\max}(M)$, $\lambda_{\min}(M)$ return the maximum and minimum eigenvalues of a matrix M, respectively. Given a signal $d : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$, we define $\|d\|_\infty := \sup_t \|d(t)\|$. Given a set $\mathcal{C}$, the notation $\partial \mathcal{C}$ denotes the boundary of $\mathcal{C}$ and $\text{Int}(\mathcal{C})$ denotes the interior of $\mathcal{C}$. We define $\mathcal{B}_r(x_0) := \{x \in \mathbb{R}^n \mid \|x - x_0\| \leq r\}$ as a closed ball of radius $r > 0$ centered at $x_0 \in \mathbb{R}^n$.

2 Preliminaries and Problem Formulation

Consider a nonlinear control affine system of the form

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

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz, and $u \in \mathbb{R}^m$ is the control input. We assume that $f(0) = 0$ so that the origin is in equilibrium point of the uncontrolled system. Let $k(x, t)$ be a

feedback control policy such that the closed-loop vector field $f_{cl}(x,t) := f(x) + g(x)k(x,t)$ is locally Lipschitz in x and piecewise continuous in t. Under these assumptions, given an initial condition $x_0 := x(t_0) \in \mathbb{R}^n$ at time $t = t_0$, there exists some maximal interval of existence $\mathcal{I}(x_0) = [t_0, \tau_{\max})$ such that $x(t)$ satisfies

$$\dot{x}(t) = f(x(t)) + g(x(t))k(x(t),t), \quad x(t_0) = x_0, \quad (2)$$

for all $t \in \mathcal{I}(x_0)$. We say that a set $\mathcal{C} \subset \mathbb{R}^n$ is *forward invariant* for (2) if, for each $x_0 \in \mathcal{C}$, the solution to (2) satisfies $x(t) \in \mathcal{C}$ for all $t \in \mathcal{I}(x_0)$. If $\mathcal{C}$ is forward invariant for (2), we say the closed-loop system is *safe* with respect to $\mathcal{C}$ and that $\mathcal{C}$ is a *safe set*. In this paper, we consider candidate safe sets of the form

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}, \quad (3a)$$
$$\partial\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) = 0\}, \quad (3b)$$
$$\text{Int}(\mathcal{C}) = \{x \in \mathbb{R}^n \mid h(x) > 0\}, \quad (3c)$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. A powerful tool for designing control policies that render sets of the form (3) forward invariant is the concept of a *control barrier function* (CBF) [1,2], defined as follows:

Definition 1 ([2]) *A continuously differentiable function $b : Int(\mathcal{C}) \to \mathbb{R}_{\geq 0}$ is said to be a* control barrier function *(CBF) for (1) over a set $\mathcal{C} \subset \mathbb{R}^n$ as in (3) if there exist $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}$ such that for all $x \in Int(\mathcal{C})$*

$$\frac{1}{\alpha_1(h(x))} \leq b(x) \leq \frac{1}{\alpha_2(h(x))}, \quad (4a)$$

$$\inf_{u \in \mathbb{R}^m} \{\nabla b(x)f(x) + \nabla b(x)g(x)u\} \leq \alpha_3(h(x)). \quad (4b)$$

We refer to a continuously differentiable function b satisfying (4a) as a *candidate* CBF. The main result regarding CBFs is that the existence of such a function implies the existence of a policy $u = k(x)$ that renders $\text{Int}(\mathcal{C})$ forward invariant for (1) [2, Cor. 1].

Problem 1 *Consider system (1) and a set $\mathcal{C} \subset \mathbb{R}^n$ as in (3). Given a candidate CBF $b : Int(\mathcal{C}) \to \mathbb{R}_{\geq 0}$, find a control policy $u = k(x)$ such that $Int(\mathcal{C})$ is forward invariant for (1) and the origin is stable for (1).*

Our first step to addressing Problem 1 is to introduce a new class of CBFs that retain the crucial properties of CBFs but are also positive semi-definite on $\text{Int}(\mathcal{C})$, which becomes important when ensuring that the origin is an equilibrium point for the closed-loop system. To find a stabilizing control policy, we take a RL-based approach in which the value function of an optimal control problem is safely learned online.

3 Lyapunov-like Control Barrier Functions

In this section, we propose a new class of CBFs and present various technical results that illustrate their properties. Based upon the development from [28], we consider the following *recentered* barrier function [38]

$$B(x) := (b(x) - b(0))^2, \quad (5)$$

where $b : \text{Int}(\mathcal{C}) \to \mathbb{R}_{\geq 0}$ is a candidate CBF with $0 \in \text{Int}(\mathcal{C})$. Throughout this paper we refer to (5) as a *Lyapunov-like CBF* (LCBF) candidate as B can be shown to satisfy the important properties

$$\inf_{x \in \text{Int}(\mathcal{C})} B(x) \geq 0, \qquad B(0) = 0, \qquad \lim_{x \to \partial\mathcal{C}} B(x) = \infty.$$

The primary distinction between standard CBFs and LCBFs is that B vanishes at the origin, which will become important when combining the subsequently designed safeguarding controller with a nominal stabilizing policy. The following lemma establishes conditions on B that ensure the forward invariance of $\mathcal{C}$.

Lemma 1 *Consider system (2) and assume the origin is contained in $Int(\mathcal{C})$. Provided $B(x(t)) < \infty$ for all $t \in \mathcal{I}(x_0)$, then $Int(\mathcal{C})$ is forward invariant for (2).*

PROOF. Since $b(x)$ shares the same properties as $1/\alpha(h(x))$ for $\alpha \in \mathcal{K}$ [2, Remark 1] and since $0 \in \text{Int}(\mathcal{C})$ implies $b(0) < \infty$, we have $B(x) < \infty \implies 1/\alpha(h(x)) < \infty \implies h(x) > 0$. Thus, if $B(x(t)) < \infty$, $\forall t \in \mathcal{I}(x_0)$, then $h(x(t)) > 0$, $\forall t \in \mathcal{I}(x_0)$, which implies $x(t) \in \text{Int}(\mathcal{C})$, $\forall t \in \mathcal{I}(x_0)$. $\square$

Similar to how CBFs are often used as a safety filter for a nominal control policy [1], in this paper we exploit the properties of LCBFs to endow existing control policies for (1) with strong safety guarantees. In particular, we propose the following *safeguarding controller*

$$k_b(x) := -c_b g(x)^T \nabla B(x)^T, \quad (6)$$

where $c_b \in \mathbb{R}_{>0}$ is a gain and B is a LCBF candidate. The following assumption places conditions on the dynamics (1), the safe set $\mathcal{C}$, and candidate CBF b that facilitate the development of subsequent results.

Assumption 1 *Given system (1) and a candidate safe set $\mathcal{C}$, the following conditions hold:*

(1) *There exists a positive, non-decreasing function $\ell_f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that $\|f(x)\| \leq \ell_f(\|x\|)\|x\|$ for all $x \in \mathcal{C}$ and $\lim_{x \to \partial\mathcal{C}} \ell_f(\|x\|)\|x\| < \infty$.*
(2) *There exists a positive constant $\underline{g} \in \mathbb{R}_{>0}$ such that $\underline{g} \leq \|g(x)\|$ for all $x \in \mathcal{C}$.*

(3) There exists a neighborhood of $\partial\mathcal{C}$, denoted by $\mathcal{N}(\partial\mathcal{C})$, such that $0 \notin \mathcal{N}(\partial\mathcal{C})$ and $\|\nabla b(x)g(x)\| \neq 0$ for all $x \in \mathcal{N}(\partial\mathcal{C})$.

Remark 1 *The first condition in Assumption 1, made largely for technical reasons, places growth restrictions on the system drift and ensures the drift dynamics do not "blow up" on the boundary of $\mathcal{C}$. The second assumption ensures the control directions do not vanish where control authority may be required to keep the system safe. The third condition is, in essence, a feasibility assumption that ensures the existence of control values that render the safe set forward invariant and is not restrictive provided the candidate CBF b has relative degree one with respect to (1). As discussed later in Remark 2, this last condition is stronger than necessary and can be slightly relaxed provided b is a valid CBF. Additionally, note that $\|\nabla b(x)g(x)\| \neq 0$ for all $x \in \mathcal{N}(\partial\mathcal{C})$ in conjunction with $0 \notin \mathcal{N}(\partial\mathcal{C})$ implies $\|\nabla B(x)g(x)\| \neq 0$ for all $x \in \mathcal{N}(\partial\mathcal{C})$.*

The following theorem shows that, under Assumption 1, the policy in (6) renders $\text{Int}(\mathcal{C})$ forward invariant for (1).

Theorem 1 *Consider system (1), a set $\mathcal{C} \subset \mathbb{R}^n$ as in (3) with $0 \in \text{Int}(\mathcal{C})$, and let $b : \text{Int}(\mathcal{C}) \to \mathbb{R}_{\geq 0}$ be a candidate CBF for (1) on $\mathcal{C}$. Provided Assumption 1 holds, the controller $u = k_b(x)$ with k_b as in (6) renders $\text{Int}(\mathcal{C})$ forward invariant for the closed-loop system (1).*

PROOF. Taking the derivative of B along the closed-loop vector field yields

$$\dot{B}(x) = \nabla B(x)f(x) + \nabla B(x)g(x)k_b(x)$$
$$= \nabla B(x)f(x) - c_b\|\nabla B(x)g(x)\|^2.$$

Provided Assumption 1 holds, $\dot{B}$ can be upper bounded for all $x \in \mathcal{N}(\partial\mathcal{C})$ as

$$\dot{B}(x) \leq \|\nabla B(x)\|\ell_f(\|x\|)\|x\| - c_b\|\nabla B(x)g(x)\|^2$$
$$\leq \|\nabla B(x)\|\ell_f(\|x\|)\|x\| - c_b\|\nabla B(x)\|^2\underline{g}^2 \quad (7)$$
$$= \|\nabla B(x)\|^2 \cdot \left(\frac{\ell_f(\|x\|)\|x\|}{\|\nabla B(x)\|} - c_b\underline{g}^2 \right).$$

If $\ell_f(\|x\|)\|x\|$ is finite in the limit as $x \to \partial\mathcal{C}$ then $\ell_f(\|x\|)\|x\|/\|\nabla B(x)\| = 0$ in the limit as $x \to \partial\mathcal{C}$, which implies that $(\ell_f(\|x\|)\|x\|/\|\nabla B(x)\| - c_b\underline{g}^2) = -c_b\underline{g}^2$ in the limit as $x \to \partial\mathcal{C}$. Hence, taking limits in (7) as x tends to $\partial\mathcal{C}$ yields

$$\lim_{x \to \partial\mathcal{C}} \dot{B}(x) = -\infty < 0. \quad (8)$$

The fact that $\lim_{x \to \partial\mathcal{C}} \dot{B}(x) < 0$ precludes the existence of trajectories that enter $\partial\mathcal{C}$. To see this, suppose there exists a trajectory of the closed-loop system $t \mapsto x(t)$

with $x_0 \in \text{Int}(\mathcal{C})$ defined on some maximal interval of existence $\mathcal{I}(x_0)$ such that $\lim_{t \to T} B(x(t)) = \infty$ for some finite $T \in \mathcal{I}(x_0)$. Since $B(x) \to \infty \iff b(x) \to \infty$, this implies that $\lim_{t \to T} x(t) \in \partial\mathcal{C}$. However, since $\lim_{t \to T} B(x(t)) = \infty$ for some finite $T \in \mathcal{I}(x_0)$, this also implies that $\lim_{t \to T} \dot{B}(x(t)) = \infty$, which contradicts (8). Hence, there cannot exist a trajectory of the closed-loop system for which $\lim_{t \to T} B(x(t)) = \infty$ for some finite $T \in \mathcal{I}(x_0)$, implying $B(x(t)) < \infty$ for all $t \in \mathcal{I}(x_0)$. The forward invariance of $\text{Int}(\mathcal{C})$ follows from Lemma 1. □

Having established the safety of (1) under the influence of (6), we show in the following corollary to Theorem 1 how the safeguarding controller can be used to modify an existing control policy for (1) to guarantee safety.

Corollary 1 *Let $k(x,t)$ be a nominal control policy, locally Lipschitz in x and piecewise continuous in t, satisfying $k(0,t) = 0$ for all $t \in \mathcal{I}(x_0)$. In addition to the assumptions of Theorem 1, suppose there exists a positive, non-decreasing function $\ell_{gk} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that $\|g(x)k(x,t)\| \leq \ell_{gk}(\|x\|)\|x\|$ for all $x \in \mathcal{C}$ and $t \in \mathcal{I}(x_0)$, and $\lim_{x \to \partial\mathcal{C}} \ell_{gk}(\|x\|)\|x\| < \infty$. Then, the controller*

$$u = k(x,t) + k_b(x), \quad (9)$$

where k_b is defined as in (6), renders $\text{Int}(\mathcal{C})$ forward invariant for (1) and ensures the origin is an equilibrium point for the closed-loop system (1).

PROOF. The proof of the first part follows from redefining the drift as $f(x,t) := f(x) + g(x)k(x,t)$ and invoking Theorem 1. The second part follows from noting that B is positive semi-definite on $\text{Int}(\mathcal{C})$, hence $\nabla B(0) = 0$ and $k_b(0) = 0$. It follows from the assumption that $f(0) = 0$ and $k(0,t) = 0, \forall t \in \mathcal{I}(x_0)$ that $\dot{x} = f(0) + g(0)u(0,t) = 0, \forall t \in \mathcal{I}(x_0)$. Hence, the origin is an equilibrium point for the closed-loop system. □

Remark 2 *If Assumption 1 fails to hold in the sense that there exists a set $\mathcal{G} \subset \mathcal{N}(\partial\mathcal{C})$ for which $\|\nabla b(x)g(x)\| = 0$ for all $x \in \mathcal{G}$, safety can still be guaranteed if b is a valid CBF. For Theorem 1, this follows from the observation that (4b) implies $\|\nabla b(x)g(x)\| = 0 \implies \nabla b(x)f(x) \leq \alpha_3(h(x))$, which implies the unforced dynamics exhibit a certain degree of safety when $\|\nabla b(x)g(x)\| = 0$.*

Theorem 1 and Corollary 1 illustrate how one can modify an existing policy in a minimally invasive fashion to guarantee safety [1] . Based on the bounds on $\dot{B}$ from the proof of Theorem 1, choosing lower values of c_b implies that $\|\nabla B(x(t))\|$ must reach higher values (i.e., $h(x(t))$

---

[1] Although these results have been established for one safe set, multiple sets can be considered by defining $B(x) := \sum_i B_i(x)$, where each B_i is a LCBF over a set $\mathcal{C}_i$ as in (3).

must achieve lower values) before the trajectory $x(t)$ is "pushed" away from $\partial \mathcal{C}$. We emphasize that knowledge of the bounds on the dynamics from Assumption 1 is not required for implementation of the safeguarding controller and is made to ensure the existence of control values that render the safe set forward invariant. Knowledge of such bounds, however, can be helpful in the selection of c_b. In theory, one could select an arbitrarily small positive value of c_b so that the influence of the safeguarding controller only becomes dominant on $\partial \mathcal{C}$. In practice, however, this could lead to control inputs with large magnitude that may exceed physical actuator limits and could also lead rapid changes in the magnitude of the control input. Hence, c_b is a design parameter that must be carefully selected by the user based upon the specific problem under consideration to ensure a desirable response of the closed-loop system.

4 Approximate Dynamic Programming

4.1 Infinite-Horizon Nonlinear Optimal Control

In this section, we shift our attention to the problem of finding a stabilizing control policy for (1) that can be combined with (6) to address Problem 1. Popular approaches to solving such a problem involve uniting CBFs and CLFs to achieve dual objectives of stability and safety, yet finding a CLF for a general nonlinear control affine system (1) is a non-trivial problem - especially if the dynamics (1) are unknown. A general way to search for a CLF is to find a control policy that minimizes the infinite-horizon cost functional

$$J(x, u(\cdot)) := \int_{t_0}^{\infty} \underbrace{\left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) \right)}_{r(x(\tau), u(\tau))} d\tau,$$
(10)

where $Q \in \mathbb{R}^{n \times n}$ is positive definite and $R \in \mathbb{R}^{m \times m}$ is symmetric and positive definite. Solutions to such an optimal control problem are typically characterized in terms of the *optimal value function*

$$V^*(x(t)) := \inf_{u(\cdot) \in \mathcal{U}} \int_t^{\infty} r(x(\tau), u(\tau)) d\tau,$$
(11)

where $\mathcal{U}$ is the set of admissible control signals [2]. Provided V^* is continuously differentiable, it can be shown to be the unique positive definite solution to the *Hamilton-Jacobi-Bellman* (HJB) equation

$$0 = \inf_{u \in \mathbb{R}^m} \{ \nabla V^*(x) f(x) + \nabla V^*(x) g(x) u + r(x, u) \}, \quad (12)$$

<hr/>

[2] Given an initial condition $x(t_0) \in \mathbb{R}^n$, a control signal $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^m$ is said to be admissible if it is bounded, piecewise continuous, and $J(x(t_0), u)$ is finite.

for all $x \in \mathbb{R}^n$ with a boundary condition of $V^*(0) = 0$. Provided there exists a continuously differentiable positive definite function V^* satisfying the HJB, taking the minimum on the right-hand side of (12) yields the optimal feedback control policy as

$$k^*(x) = -\frac{1}{2} R^{-1} g(x)^T \nabla V^*(x)^T.$$
(13)

Assumption 2 *There exists a continuously differentiable positive definite function $V^*(x)$ satisfying* (12). *Moreover, $\nabla V^*(x)$ is locally Lipschitz.*

Under Assumption 2, the closed-loop system (1) with $u = k^*(x)$ can be shown to be asymptotically stable with respect to the origin using V^* as a CLF [31, Ch. 8.5], and Corollary 1 can be used to endow (13) with safety guarantees. Although this approach provides a general way of constructing a CLF for (1), the practicality of it is hindered by the need to solve the HJB equation (12) for V^*, which generally does not admit a closed-form solution, and would require an accurate system model, which may be unavailable in practice. The remainder of this paper is hence dedicated to developing a MBRL framework that can be combined with the results of Sec. 3 to safely learn V^* and the dynamics (1) online.

4.2 Value Function Approximation

Since the optimal value function V^* is unknown and difficult to compute in general, we seek a parametric approximation of V^* over some compact set $\chi \subset \mathbb{R}^n$ containing the origin. To this end, we leverage state following (StaF) kernels introduced in [16,30] to generate a local approximation of the value function within a smaller compact set $\mathcal{B}_l(x) \subset \chi$ that follows the system trajectory. Using StaF kernels, the value function (11) can be represented at points $y \in \mathcal{B}_l(x)$ as [16]

$$V^*(y) = W(x)^T \phi(y, c(x)) + \varepsilon(x, y),$$
(14)

where $W : \chi \to \mathbb{R}^L$ is a continuously differentiable ideal weight function, $\phi : \chi \times \chi \to \mathbb{R}^L$ is a vector of $L \in \mathbb{N}$ bounded, positive definite, and smooth kernel functions, where $c(x) = [c_1(x) \ldots c_L(x)]^T$ and $c_i(x) \in \mathcal{B}_l(x)$, $i \in \{1, \ldots, L\}$ denotes the center of the ith kernel, and $\varepsilon : \chi \times \chi \to \mathbb{R}$ denotes the function reconstruction error. Further details regarding the selection of kernel functions can be found in [16, Footnote 7] and [30]. The ideal weight function W in (14) is generally unknown, and is therefore replaced with approximations [3] $\hat{W}_c(t), \hat{W}_a(t) \in \mathbb{R}^L$ yielding the approximate value function and approximate optimal control policy as

$$\hat{V}(y, x, \hat{W}_c) := \hat{W}_c^T \phi(y, c(x)),$$
(15a)

<hr/>

[3] The use of separate weight estimates is motivated by the fact that (18) is affine in $\hat{W}_c$ [35].

$$\hat{k}(y, x, \hat{W}_a) := -\frac{1}{2} R^{-1} g(y)^T \nabla \phi(y, c(x))^T \hat{W}_a. \quad (15b)$$

4.3 System Identification

In addition to not knowing the value function, we now assume that the system drift from (1) is unknown, but can be expressed as a linear combination of weights $\theta \in \mathbb{R}^p$ and user-defined basis functions $Y : \mathbb{R}^n \to \mathbb{R}^{n \times p}$ such that over a given compact set $\chi \subset \mathbb{R}^n$

$$f(x) = Y(x)\theta + \varepsilon_\theta(x), \quad (16)$$

where $\varepsilon_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is the bounded *unknown* function reconstruction error. The basis Y may capture physical knowledge of the dynamics or may represent user-defined basis functions, such as polynomials, radial basis functions, and pre-trained neural networks with tunable outer layer weights, that can approximate functions arbitrarily closely over compact sets. As the ideal weights [4] for the given basis are assumed to be unknown, let $\hat{\theta} \in \mathbb{R}^p$ denote an estimate of θ and let $\hat{f}(x, \hat{\theta}) := Y(x)\hat{\theta}$ denote the corresponding approximated drift. Provided $\hat{\theta}$ is updated by a specific class of parameter identifiers [5], then one can establish the existence of a Lyapunov-like function $V_\theta : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfying

$$\dot{V}_\theta(\tilde{\theta}, t) \leq -K_\theta \|\tilde{\theta}\|^2 + D_\theta \|\tilde{\theta}\|, \quad (17)$$

for all $\tilde{\theta} \in \mathbb{R}^p$ and $t \in \mathbb{R}_{\geq 0}$, where $\tilde{\theta} := \theta - \hat{\theta}$ denotes the weight estimation error and $K_\theta, D_\theta \in \mathbb{R}_{>0}$ are positive constants, the latter of which depends upon ε_θ (cf. [16, 17]). The bound in (17) implies that, under such an identification scheme, the weight estimation error exponentially decays to a ball about the origin, the size of which also depends upon ε_θ.

4.4 Bellman Error

A performance metric for learning the ideal parameters of the value function and policy can be obtained by replacing the optimal value function, optimal policy, and true drift in the HJB (12) with their corresponding approximations yielding the *Bellman error* (BE)

$$\begin{aligned} \delta(y, x, \hat{W}_c, \hat{W}_a, \hat{\theta}) := & \nabla \hat{V}(y, x, \hat{W}_c) \hat{f}(y, \hat{\theta}) \\ & + \nabla \hat{V}(y, x, \hat{W}_c) g(y) \hat{k}(y, x, \hat{W}_a) \\ & + r(y, \hat{k}(y, x, \hat{W}_a)). \end{aligned}$$
$$(18)$$

Given weight estimates $(\hat{W}_c, \hat{W}_a)$, the BE, evaluated at any point in the state-space, encodes the "distance"

of the approximations $(\hat{V}, \hat{k})$ from their true values (V^*, k^*). Thus, the objective is to select $(\hat{W}_c, \hat{W}_a)$ such that $\delta(y, x, \hat{W}_c, \hat{W}_a, \hat{\theta}) = 0$ for all $x \in \chi$ and all $y \in \mathcal{B}_l(x)$, which is accomplished by updating $(\hat{W}_c, \hat{W}_a)$ online using techniques from adaptive control [5, 21, 35].

5 Safe Exploration via Simulation of Experience

This section builds upon the approach from [16–18] to develop a safe exploration framework for learning V^* and k^* online while guaranteeing safety. To this end, define the control policy for (1) as

$$u = \hat{k}(x, x, \hat{W}_a) - \frac{c_b}{2} R^{-1} g(x)^T \nabla B(x)^T, \quad (19)$$

where $\hat{k}$ is defined as in (15b), $c_b \in \mathbb{R}_{>0}$ is a gain, and ∇B is a LCBF as in (5), which is a special case of the controller proposed in Corollary 1. Define the regressor $\omega(t) := \nabla \phi(x(t), c(x(t))) \cdot (\hat{f}(x(t), \hat{\theta}(t)) + g(x(t))u(t))$ and the resulting BE along the system trajectory as

$$\delta_t(t) := r(x(t), u(t)) + \hat{W}_c(t)^T \omega(t). \quad (20)$$

In contrast to (18), the version of the BE in (20) is a function of ∇B and therefore selecting the weight estimates to minimize (20) may not correspond to the minimization of the original BE in (18). That is, even if $(\hat{W}_c, \hat{W}_a) \to W$, the BE in (20) may be large at certain points in the state-space because of the influence of the safeguarding component of (19), making (20) a non-ideal performance metric for learning. However, given an approximate model of the system, the BE can be evaluated at any point in the state-space [17] using a *different* policy to generate data more representative of (18). To facilitate this approach, define the family of mappings $\{x_i : \chi \times \mathbb{R}_{\geq t_0} \to \chi\}_{i=1}^N$ such that each $x_i(x(t), t) \in \mathcal{B}_l(x(t))$ maps the current state $x(t)$ to some unexplored point in $\mathcal{B}_l(x(t))$. For each extrapolated trajectory [6], $t \mapsto x_i(x(t), t), i \in \{1, \dots, N\}$, we define an *exploratory policy* as $u_i(t) := \hat{k}(x_i(x(t), t), x(t), \hat{W}_a(t))$ with $\hat{k}$ as in (15b), which yields the BE along the extrapolated system trajectories as [7]

$$\delta_i(t) := r(x_i(x(t), t), u_i(t)) + \hat{W}_c(t)^T \omega_i(t). \quad (21)$$

Note that the exploratory policy u_i is not augmented with the safeguarding controller, thereby allowing for maximum exploration of the state-space by the extrapolated system trajectories without risking safety violation of the original system. Consequently, the BE in (21) is representative of the original BE from (18) and is used

---

[4] For a given basis Y, the ideal weight vector is defined as $\theta := \arg\min_{\hat{\theta} \in \mathbb{R}^p} \sup_{x \in \chi} \|f(x) - Y(x)\hat{\theta}\|$ (see [14, Ch. 8.7]).
[5] See [17, Assumption 2] for conditions that such an identifier must satisfy and [5, 7, 8, 29] for examples of identifiers.

[6] See [8, Rem. 5] for details on generating trajectories.
[7] Mappings with the subscript i denote evaluations along the extrapolated trajectories.

to update the weight estimates using a recursive least squares update law as

$$\dot{\hat{W}}_c(t) = -\Gamma(t)\left(k_{c1}\frac{\omega(t)}{\rho^2(t)}\delta_t(t) + \frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i(t)}{\rho_i^2(t)}\delta_i(t)\right) \tag{22}$$

$$\dot{\Gamma}(t) = \beta_c\Gamma(t) - \Gamma(t)\left(k_{c1}\Lambda(t) + \frac{k_{c2}}{N}\sum_{i=1}^{N}\Lambda_i(t)\right)\Gamma(t), \tag{23}$$

where $\Lambda(t) := \frac{\omega(t)\omega(t)^T}{\rho^2(t)}$, $\rho(t) := 1 + \gamma_c\omega(t)^T\omega(t)$ is a normalizing signal with a gain of $\gamma_c \in \mathbb{R}_{>0}$, $k_{c1}, k_{c2} \in \mathbb{R}_{>0}$ are learning gains, and $\beta_c \in \mathbb{R}_{>0}$ is a forgetting factor. In (22)-(23) the terms $\omega_i, \rho_i, \Lambda_i$ are defined in a similar manner to ω, ρ, Λ. The weight update law for $\hat{W}_a$ is then selected as

$$\begin{aligned}\dot{\hat{W}}_a(t) = \text{proj}\Big\{ &- k_{a1}(\hat{W}_a(t) - \hat{W}_c(t)) - k_{a2}\hat{W}_a(t) \\ &+ \frac{k_{c1}}{4\rho^2(t)}G_\phi(t)^T\hat{W}_a(t)\omega(t)^T\hat{W}_c(t) \\ &+ \sum_{i=1}^{N}\frac{k_{c2}}{4N\rho_i^2(t)}G_{\phi,i}(t)^T\hat{W}_a(t)\omega_i(t)^T\hat{W}_c(t)\Big\},\end{aligned} \tag{24}$$

where $G_\phi := \nabla\phi(x, c(x))G_R(x)\nabla\phi(x, c(x))^T$, $G_R := g(x)R^{-1}g(x)^T$, $k_{a1}, k_{a2} \in \mathbb{R}_{>0}$ are learning gains, and proj$\{\cdot\}$ is a smooth projection operator, standard in the adaptive control literature [21, Appendix E], that ensures the weight estimates remain bounded. The following proposition shows that the policy in (19) renders Int$(\mathcal{C})$ forward invariant for the closed-loop system (1).

Proposition 1 *Consider system* (1), *a set* $\mathcal{C} \subset \mathbb{R}^n$ *as in* (3) *with* $0 \in Int(\mathcal{C})$, *and let* $b : Int(\mathcal{C}) \to \mathbb{R}_{\geq 0}$ *be a candidate CBF. Provided Assumptions 1-2 hold and there exists a positive, non-decreasing function* $\ell_{g\phi} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *such that* $\|g(x)^T\nabla\phi(x, c(x))^T\| \leq \ell_{g\phi}(\|x\|)\|x\|$ *for all* $x \in \mathcal{C}$ *and* $\lim_{x\to\partial\mathcal{C}}\ell_{g\phi}(\|x\|)\|x\| < \infty$, *then the control policy in* (19) *and weight update law in* (24) *ensure* $Int(\mathcal{C})$ *is forward invariant for* (1) *and ensure the origin is an equilibrium point for the closed-loop system* (1).

PROOF. Under the assumptions of the proposition, the nominal policy $\hat{k}(x, x, \hat{W}_a)$ can be bounded as $\|\hat{k}(x, x, \hat{W}_a)\| \leq \frac{1}{2}\lambda_{\max}(R^{-1})\bar{W}_a\ell_{g\phi}(\|x\|)\|x\|$ for all $x \in \mathcal{C}$, where $\|\hat{W}_a\| \leq \bar{W}_a$ for some $\bar{W}_a \in \mathbb{R}_{>0}$ follows from the use of the projection operator in (24). The proof then follows from letting $k(x, t) = \hat{k}(x, x, \hat{W}_a(t))$ and invoking Corollary 1. $\square$

In contrast to related approaches [6, 11, 26, 27, 39], the above proposition does not make use of any Lyapunov-

based arguments that require the value function to decrease along the system trajectory, only that $\hat{W}_a$ remains bounded, which is guaranteed by the use of the projection operator in (24). The following assumption outlines the exploration conditions required for the approximations to converge to a neighborhood of their ideal values.

Assumption 3 ([7, 16]) *There exist constants* $T \in \mathbb{R}_{>0}$ *and* $\underline{c}_1, \underline{c}_2, \underline{c}_3 \in \mathbb{R}_{\geq 0}$ *such that for all* $t \geq t_0$ *we have* $\underline{c}_1 I_L \leq \int_t^{t+T}\Lambda(\tau)d\tau$, $\underline{c}_2 I_L \leq \int_t^{t+T}\frac{1}{N}\sum_{i=1}^{N}\Lambda_i(\tau)d\tau$, *and* $\underline{c}_3 I_L \leq \inf_{t\geq t_0}(\frac{1}{N}\sum_{i=1}^{N}\Lambda_i(t))$, *where at least one of* $\underline{c}_1, \underline{c}_2, \underline{c}_3$ *is strictly greater than zero* [8].

5.1 Stability Analysis

To facilitate the analysis of the closed-loop system under the influence of the controller in (19), define the weight estimation errors $\tilde{W}_c(t) := W(x(t)) - \hat{W}_c(t)$, $\tilde{W}_a(t) := W(x(t)) - \hat{W}_a(t)$, and a composite state vector $Z := [x^T, \tilde{W}_c^T, \tilde{W}_a^T, \tilde{\theta}^T]^T$. Now consider the following Lyapunov function candidate

$$V_L(Z, t) := V^*(x) + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}(t)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a + V_\theta(\tilde{\theta}, t),$$

where V_θ is from (17). If $\lambda_{\min}(\Gamma^{-1}(t_0)) > 0$ and Assumption 3 is satisfied, then $\Gamma(t)$ can be shown to satisfy $\underline{\Gamma}I_L \leq \Gamma(t) \leq \overline{\Gamma}I_L$ for all $t \geq t_0$, where $\underline{\Gamma}, \overline{\Gamma} \in \mathbb{R}_{>0}$ [16, Lemma 1], which implies V_L is positive definite and hence satisfies $\eta_1(\|Z\|) \leq V_L(Z, t) \leq \eta_2(\|Z\|)$ for all $t \geq t_0$ with $\eta_1, \eta_2 \in \mathcal{K}$ [19, Lemma 4.3]. The following theorem illustrates that the safe exploration scheme ensures the system state and weight estimation errors remain uniformly ultimately bounded.

Theorem 2 *Consider system* (1), *the cost functional in* (10), *a set* $\mathcal{C} \subset \mathbb{R}^n$ *as in* (3) *with* $0 \in Int(\mathcal{C})$, *and let* $b : Int(\mathcal{C}) \to \mathbb{R}_{\geq 0}$ *be a candidate CBF for* (1) *on* $\mathcal{C}$. *Let the optimal value function and corresponding optimal control policy be approximated over a compact set* $\chi \subset \mathbb{R}^n$ *as detailed in Sec. 4.2, and let* $\mathcal{B}_\zeta(0) \in \chi \times \mathbb{R}^{2L+p}$ *be a closed ball of radius* $\zeta \in \mathbb{R}_{>0}$ *centered at* $Z = 0$. *Provided Assumptions 1-3, the conditions of Proposition 1, and*

---

[8] The first condition in Assumption 3 is the persistence of excitation (PE) condition, whose satisfaction cannot be verified in general, but can be achieved heuristically by including an exploration/probing signal in the control input. The second condition is the same PE condition, but is placed on the *extrapolated trajectories*. Since the extrapolated trajectories are user-defined through the mappings $\{x_i(x(t), t)\}_{i=1}^{N}$, one can construct these mappings in an attempt to satisfy such a condition [8, Rem. 5] without having to excite the original system. The third condition can be heuristically satisfied by selecting many extrapolation points N; however, this scales poorly with the state dimension (cf. [16]).

the inequaility in (17) hold, $x(t_0) \in Int(\mathcal{C})$, and

$$\lambda_{\min}(M) > 0, \quad \sqrt{\tfrac{2\iota}{\kappa}} < \eta_1^{-1}\left(\eta_2(\zeta)\right), \qquad (25)$$

where

$$M := \begin{bmatrix} \frac{\underline{c}k_{c2}}{4} & -\frac{\varphi_{ac}}{2} & -\frac{\varphi_{c\theta}}{2} \\ -\frac{\varphi_{ac}}{2} & \left(\frac{k_{a1}+k_{a2}}{4} - \varphi_a\right) & 0 \\ -\frac{\varphi_{c\theta}}{2} & 0 & \frac{K_\theta}{4} \end{bmatrix}, \qquad (26)$$

with $\underline{c} := (\frac{c_3}{2} + \frac{\beta_c}{2k_{c2}\overline{\Gamma}})$ and $\kappa, \iota, \varphi_a, \varphi_{ac}, \varphi_{c\theta} \in \mathbb{R}_{>0}$ defined in the Appendix, then the control policy in (19) and weight update laws in (22), (23), (24) guarantee that $Int(\mathcal{C})$ is forward invariant for the closed-loop system (1) and that $Z(t)$ is uniformly ultimately bounded such that

$$\limsup_{t\to\infty} \|Z(t)\| \leq \eta_1^{-1} \circ \eta_2(\sqrt{2\iota/\kappa}). \qquad (27)$$

Remark 3 *It is difficult to verify when the sufficient conditions in (25) are satisfied because M and ι depend on terms that are either unknown, such as ε, or completely determined by the system trajectory, such as $\underline{c}$, and therefore one cannot formally guarantee the satisfaction of (25) a priori. Although the stability guarantees depend upon these conditions, the safety guarantees of the proposed framework have already been established in Proposition 1, which is in contrast to related approaches that use the value function as a safety certificate. The definitions provided in the Appendix imply that (25) can be satisfied by selecting a sufficient number of basis functions L, which decreases the approximation error ε, by choosing R such that $\lambda_{\min}(R)$ is large, and by ensuring that $\underline{c}$ is large, which can be achieved using the methods mentioned in Footnote 8. The ultimate bound is also a function of $\|\nabla B(x(t))\|_\infty$ and therefore selecting c_b to be small can aid in satisfying the sufficient conditions.*

PROOF. The forward invariance of $Int(\mathcal{C})$ follows from Proposition 1. Expressing (20) and (21) in terms of the weight estimation errors and then subtracting the right-hand side of the HJB equation (12) using the StaF representation of V^* from (14) yields an alternative form of the BEs as $\delta_t = -\tilde{W}_c^T\omega + \frac{1}{4}\tilde{W}_a^T G_\phi \tilde{W}_a - W^T\nabla\phi Y\tilde{\theta} - \frac{1}{2}c_b\nabla B G_R \nabla\phi^T \tilde{W}_a + \frac{1}{4}c_b^2\nabla B G_R \nabla B^T + \Delta$ and $\delta_i = -\tilde{W}_c^T\omega_i + \frac{1}{4}\tilde{W}_a^T G_{\phi,i}\tilde{W}_a - W_i^T\nabla\phi_i Y_i\tilde{\theta} + \Delta_i$, where functional dependencies have been suppressed for ease of presentation and $\Delta(x), \Delta_i(x_i)$ consist of terms that are uniformly bounded over χ. Taking the derivative of V_L along $Z(t)$ yields

$$\dot{V}_L = \nabla V^* f + \nabla V^* g u + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}(\dot{W} - \dot{\hat{W}}_c)$$
$$\quad - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T(\dot{W} - \dot{\hat{W}}_a) + \dot{V}_\theta.$$

Using $\nabla V^* f = -\nabla V^* g k^* - k^{*T}Rk^* - x^T Q x$ from the HJB (12), $\dot{W} = \nabla W(f + gu)$, substituting in the weight update laws (22), (23), (24) using the alternate form of the BEs, expressing everything in terms of the weight estimation errors, upper bounding and then completing squares yields $\dot{V}_L \leq -\kappa\|Z\|^2 - \underline{Z}^T M\underline{Z} + \iota$, where $\underline{Z} := [\|\tilde{W}_c\|\ \|\tilde{W}_a\|\ \|\tilde{\theta}\|]^T$. Provided (25) holds, further bounding yields $\dot{V}_L \leq -\frac{\kappa}{2}\|Z\|^2 \ \forall \zeta \geq \|Z\| \geq \sqrt{\frac{2\iota}{\kappa}}$. Finally, invoking [19, Thm. 4.18] implies $Z(t)$ is uniformly ultimately bounded such that (27) holds. □

6 Numerical Examples

Nonlinear System To demonstrate the efficacy of the developed approach, we apply our method to the scenario from [15]. Consider a system as in (1) with $x \in \mathbb{R}^2$, $u \in \mathbb{R}$, $f(x) = [-0.6x_1 - x_2, x_1^3]^T$ and $g(x) = [0, x_2]^T$. The drift is assumed to be unknown but linear in the unknown parameters (i.e., $\varepsilon_\theta \equiv 0$) and is represented as $f(x) = Y(x)\theta$, where $Y(x) = [x_1, x_2, 0; 0, 0, x_1^3]$ and $\theta = [-0.6, -1, 1]^T$. The objective is to drive $x(t)$ to the origin while ensuring $x(t)$ remains in a set $\mathcal{C}$ as in (3) defined by $h(x) = px_2^2 - x_1 + 1$, where $p \in \{-1, 1\}$ determines if $\mathcal{C}$ is convex ($p = -1$) or not ($p = 1$). To the best of our knowledge, such a set cannot be rendered safe using the approaches from [11, 26, 39] as the constraints are not of the form $\underline{a}_i < x_i < \overline{a}_i \ \forall i \in \{1, \ldots, n\}$ with $\underline{a}_i, \overline{a}_i \in \mathbb{R}$. For each simulation, a safeguarding controller is obtained by constructing the CBF $b(x) = 1/h(x)$ and then constructing an LCBF as in (5). To obtain a stabilizing control policy, we define an optimal control problem as in (10) with $Q = I_2$ and $R = 1$. The resulting value function is approximated using a basis of $L = 3$ StaF kernels $\phi(x, c(x)) = [\phi_1(x, c_1(x))\ \phi_2(x, c_2(x))\ \phi_3(x, c_3(x))]^T$, where each kernel is selected as the polynomial kernel $\phi_i(x, c_i(x)) = x^T c_i(x)$. The centers of each kernel are placed on the vertices of an equilateral triangle centered at the current state as $c_i(x) = x + \nu(x)d_i$, where $d_i \in \mathbb{R}^2$ corresponds to vertices of the triangle and $\nu(x) := \frac{x^T x}{x^T x + 1}$. The learning gains are selected as $k_{c1} = 0.1$, $k_{c2} = 1$, $k_{a1} = 1$, $k_{a2} = 0.1$, $\gamma_c = 1$, $\beta_c = 0.001$ and the weights are initialized as $\hat{W}_c(t_0) = \hat{W}_a(t_0) = [0.5, 0.5, 0.5]^T$ and $\Gamma(t_0) = 100I_3$. The learning procedure outlined in Sec. 5 is carried out by extrapolating the BE to one point from a uniform distribution over a $\nu(x) \times \nu(x)$ square centered at x every time-step, where the drift parameters are identified online using integral concurrent learning [8, 29].

To illustrate efficacy of the developed approach, the system is simulated with the safeguarding controller, without the safeguarding controller, and without the safeguarding controller where the LCBF is incorporated into the cost function in a similar fasion to [6, 27]. For the first simulation, we consider the convex safe set defined by $p = -1$ and the policy in (19) with $c_b = 1$, the results

of which are provided in Fig. 1 (left) and Fig. 2 (left). As shown in Fig. 1, the RL policy in (19) augmented with the safeguarding controller stabilizes $x(t)$ to the origin without leaving the safe set. Without the safeguarding component, the RL policy stabilizes the system, but violates the safety constraints multiple times. When the CBF term is included in the problem's cost function, the system initially violates the safety constraints, but eventually converges to a safe policy. For this particular example, the system rapidly approaches the boundary of the safe set (see the top left plot in Fig. 2) and crosses into the unsafe region before enough time has passed for convergence to a suitable policy without additional help from the safeguarding controller. This phenomenon underscores the crucial distinction between the approach taken herein and those in, e.g., [6, 27], where safety is enforced by including a CBF in the problem's cost function. In essence, these aforementioned approaches aim to *learn a safe policy*, whereas the approach presented herein aims to *safely learn a performance policy*. Ultimately, this allows for safe exploration while learning an uncertain system model (Fig. 2, bottom) and an approximately optimal policy. Moreover, note that the safeguarding controller is minimally invasive in the sense that it intervenes only when absolutely necessary to prevent safety violation. In fact, the trajectories of the controlled system with and without the safeguarding controller are almost identical up until the point at which $x(t)$ approaches $\partial \mathcal{C}$. To demonstrate the ability of the RL policy to safely stabilize the system within a nonconvex safe set, a second simulation under the RL policy with and without the safeguarding controller is run with $p = 1$, the results of which are shown in Fig. 1 (right) and Fig. 2 (right). All parameters remain the same as before with the exception of $c_b = 0.001$ and $\Gamma(t_0) = 10I_3$. Once again, the policy in (19) renders the system safe and the trajectory under the nominal and safe policy only diverge from each other if intervention from the safeguarding controller is required to ensure safety.

Collision Avoidance We now examine a simple scenario to demonstrate some interesting properties of the proposed method. Consider a mobile robot modeled as a two-dimensional single integrator $\dot{x} = u$ tasked with navigating to the origin while avoiding a circular obstacle centered at $x_o \in \mathbb{R}^2$ with a radius of $r_o \in \mathbb{R}_{>0}$. The safety objective can be addressed by considering a set $\mathcal{C}$ defined by $h(x) = \|x - x_o\|^2 - r_o^2$. Similar to the previous example, the safe set is not a hyper-rectangle and therefore the techniques in [11, 26, 39] cannot be applied. To obtain a stabilizing control policy we associate with the single integrator an optimal control problem as in (10) with $Q = I_2$ and $R = I_2$. Since the system is linear and the cost is quadratic, one could solve the algebraic Riccati equation (ARE) to obtain the optimal policy and then augment it with the safeguarding controller as in Corollary 1. However, as demonstrated in the subsequent
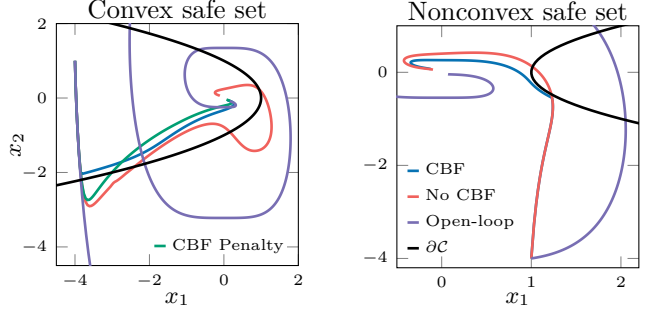

Fig. 1. Trajectories of the nonlinear system evolving in the convex (left) and nonconvex (right) safe sets. In each plot the blue curve denotes the trajectory under the RL policy augmented with the safeguarding controller, the orange curve denotes the trajectory under only the RL policy, the green curve denotes the trajectory under the RL policy with a cost function of $r(x, u) = \|x\|^2 + u^2 + 20B(x)$, the purple curve denotes the open-loop trajectory, and the black curve denotes the boundary of the safe set.
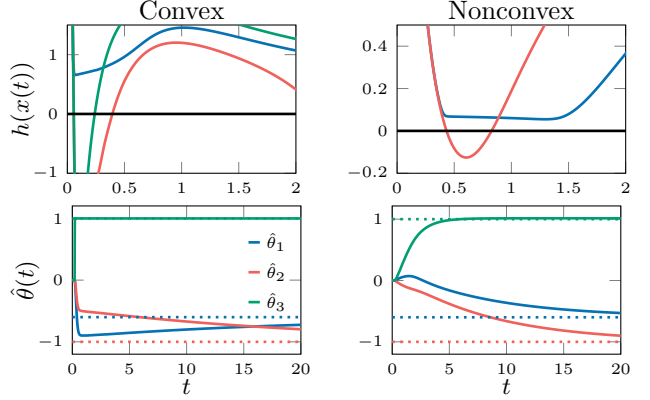

Fig. 2. Evolution of the CBF h along the system trajectory (top) and trajectory of the estimated drift dynamics weights along the trajectory of the nonlinear system under the RL policy augmented with the safeguarding controller (bottom). In the top plots, the curves of various color have the same interpretation as those in Fig. 1. In the bottom plots, the dotted lines of corresponding color denote the true values of the uncertain parameters.

numerical results, such an approach presents certain limitations. To compare the analytical solution with the learning-based solution, the corresponding value function is approximated using the same parameters as in the previous example with all weights initialized to 1 and $\Gamma(t_0) = 10I_3$. Since the dynamics for this example are trivial, no system identification is performed.

Simulations are performed to compare the policy from (19) with that obtained from solving the ARE, both of which are augmented with the safeguarding controller with $c_b = 0.1$. The resulting system trajectories are provided in Fig. 3, where the trajectory under the RL policy navigates around the obstacle and converges to the origin, whereas the trajectory under the linear quadratic

regulation (LQR) policy gets stuck behind the obstacle. Note that LQR policy augmented with the safeguarding controller is a continuous *time-invariant* feedback controller and can therefore not achieve dual objectives of obstacle avoidance and stability from certain initial conditions [9]. On the other hand, the RL policy is continuous, but is also *time-varying* because the policy explicitly depends on the evolution of $\hat{W}_a$. Indeed, adaptive control methods in general can be seen as a form of nonlinear *dynamic* feedback [21]. As illustrated in Fig. 3 (top), both trajectories quickly approach the obstacle and initially get stuck, failing to make progress towards the origin. However, unlike the static LQR policy, the weights of the learning-based controller dynamically evolve, as shown in Fig. 3 (bottom), and eventually converge to a new policy that successfully navigates the system around the obstacle and to the origin.
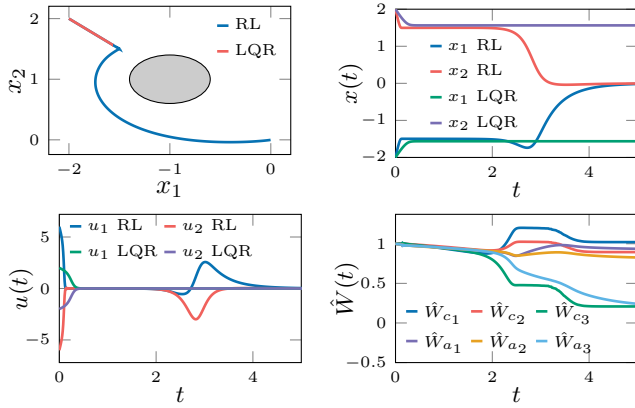

Fig. 3. State (top), control (bottom left), and weight (bottom right) trajectory for the single integrator simulation. In the top left plot the gray disk denotes the obstacle.

7 Conclusions

In this paper, we developed a safe MBRL framework that allows one to learn online the value function of an optimal control problem and the drift dynamics of an uncertain control affine system while satisfying safety constraints given as CBFs. Our approach was facilitated by the introduction of a new class of CBFs, termed LCBFs, that were used to augment a learning-based control policy to guarantee stability and safety. The benefits of the proposed method were illustrated by introducing numerical examples that, to the best of our knowledge, cannot be handled by related approaches. Directions for future research include integrating zeroing CBFs into the developed framework using approaches such as in [25,32], which may address the limitations mentioned after Corollary 1, and may strengthen the proposed framework in the context of uncertain systems.

---

[9] The unfamiliar reader is referred to [24, Ch. 4.1] for an intuitive discussion on this topic.

References

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: theory and applications. In *Proc. Eur. Control Conf.*, pages 3420–3431, 2019.

[2] 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*, 62(8):3861–3876, 2017.

[3] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[4] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proc. Conf. on Artificial Intel.*, volume 33, pages 3387–3395, 2019.

[5] G. Chowdhary. *Concurrent learning for convergence in adaptive control without persistency of excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.

[6] M. H. Cohen and C. Belta. Approximate optimal control for safety-critical systems with control barrier functions. In *Proc. Conf. Decis. Control*, pages 2062–2067, 2020.

[7] P. Deptula, Z. I. Bell, E. A. Doucette, J. W. Curtis, and W. E. Dixon. Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with control effectiveness faults. *Automatica*, 116:1–10, 2020.

[8] P. Deptula, Z. I. Bell, F. M. Zegers, R. A. Licitra, and W. E. Dixon. Approximate optimal influence over an agent through an uncertain interaction dynamic. *Automatica*, 134:1–13, 2021.

[9] P. Deptula, H.Y. Chen, R. Licitra, J. A. Rosenfeld, and W. E. Dixon. Approximate optimal motion planning to avoid unknown moving avoidance regions. *IEEE Trans. Robot*, 32(2):414–430, 2020.

[10] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. Autom. Control*, 64(7):2737–2752, 2019.

[11] M. L. Greene, P. Deptula, S. Nivison, and W. E. Dixon. Sparse learning-based approximate dynamic programming with barrier constraints. *IEEE Contr. Syst. Lett.*, 4(3):743–748, 2020.

[12] S. Gros and M. Zanon. Data-driven economic nmpc using reinforcement learning. *IEEE Trans. Autom. Control*, 65(2):636–648, 2020.

[13] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control Robot. Auton. Syst.*, 3:269–296, 2020.

[14] P. Ioannou and B. Fidan. *Adaptive control tutorial*. SIAM, 2006.

[15] M. Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, 96:359–367, 2018.

[16] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon. Efficient model–based reinforcement learning for approximate online optimal control. *Automatica*, 74:247–258, 2016.

[17] R. Kamalapurkar, P. Walters, and W. E. Dixon. Model–based reinforcement learning for approximate optimal regulation. *Automatica*, 64:94–104, 2016.

[18] R. Kamalapurkar, P. Walters, J. A. Rosenfeld, and W. E. Dixon. *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach*. Springer, 2018.

[19] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.

[20] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. Neural Net. Learning Syst.*, 29(6):2042–2062, 2017.

[21] M. Krstić, I. Kanellakopoulos, and P. Kokotović. *Nonlinear and adaptive control design*. John Wiley & Sons, 1995.

[22] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Syst.*, 32(6):76–105, 2012.

[23] Z. Li, U. Kalabić, and T. Chu. Safe reinforcement learning: Learning with supervision using a constraint-admissible set. In *Proc. Amer. Control Conf.*, pages 6390–6395, 2018.

[24] D. Liberzon. *Switching in systems and control*. Birkhäuser, Boston, MA, 2003.

[25] B. T. Lopez, J. J. Slotine, and J. P. How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Contr. Syst. Lett.*, 5(3):1031–1036, 2021.

[26] S. M. N. Mahmud, K. Hareland, S. A. Nivison, Z. I. Bell, and R. Kamalapurkar. A safety aware model-based reinforcement learning framework for systems with uncertainties. In *Proc. Amer. Control Conf.*, pages 1979–1984, 2021.

[27] Z. Marvi and B. Kiumarsi. Safe reinforcement learning: A control barrier function optimization approach. *Int. J. Robust Nonlin. Control*, 31(6):1923–1940, 2021.

[28] D. Panagou, D. M. Stipanovic, and P. G. Voulgaris. Distributed coordination control for multi-robot networks using lyapunov-like barrier functions. *IEEE Trans. Autom. Control*, 61(3):617–632, 2016.

[29] A. Parikh, R. Kamalapurkar, and W. E. Dixon. Integral concurrent learning: Adaptive control with parameter convergence using finite excitation. *Int. J. Adapt. Control Signal Process.*, 33(12):1775–1787, 2019.

[30] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon. The state following (staf) approximation method. *IEEE Trans. Neural Net. Learning Syst.*, 30(6):1716–1730, 2019.

[31] E. D. Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Springer Science & Business Media, 2013.

[32] A. J. Taylor and A. D. Ames. Adaptive safety with control barrier functions. In *Proc. Amer. Control Conf.*, pages 1399–1405, 2020.

[33] A. J. Taylor, A. Singletary, Y. Yue, and A. Ames. Learning for safety-critical control with control barrier functions. In *Proc. Conf. Learning for Dyn. and Control*, volume 120 of *PMLR*, pages 708–717, 2020.

[34] K. P. Tee, S. S. Ge, and E. H. Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, 2009.

[35] K. G. Vamvoudakis and F. L. Lewis. Online actor–critic algorithm to solve the continuous–time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

[36] K. G. Vamvoudakis, M. F. Miranda, and J. P. Hespanha. Asymptotically stable adaptive–optimal control algorithm with saturating actuators and relaxed persistence of excitation. *IEEE Trans. Neural Net. Learning Syst.*, 27(11):2386–2398, 2015.

[37] P. Wieland and F. Allgöwer. Constructive safety using control barrier functions. In *Proc. IFAC Symp. Nonlin. Control Syst.*, 2007.

[38] A. G. Willis and W. P. Heath. Barrier function based model predictive control. *Automatica*, 40(8):1415–1422, 2004.

[39] Y. Yang, K. G. Vamvoudakis, and H. Modares. Safe reinforcement learning for dynamical games. *Int. J. Robust Nonlin. Control*, 30(9):3706–3726, 2020.

A   Supporting Constants

Given the compact set $\chi \subset \mathbb{R}^n$ and a continuous mapping $(\cdot) : \chi \to \mathbb{R}^N$, with $N \in \mathbb{N}$, define $\overline{\|(\cdot)\|} := \sup_{x \in \chi} \|(\cdot)\|$. The constants used from (25) are defined as $\kappa := \min\{\lambda_{\min}(Q), \frac{ck_{c2}}{4}, \frac{k_{a1}+k_{a2}}{4}, \frac{K_\theta}{4}\}$, $\iota := \frac{1}{2}\overline{\|\nabla V^* G_R\|}(\overline{\|\phi^T \nabla W + \nabla \varepsilon^T\|} + c_b \|\nabla B^T\|_\infty) + \frac{\iota_c^2}{2c k_{c2}} + \frac{\iota_a^2}{2(k_{a1}+k_{a2})} + \frac{D_\theta^2}{2K_\theta}$, where $\iota_c := \frac{k_{c1}3\sqrt{3}}{64\sqrt{\gamma_c}}\overline{\|G_R\|}\|\nabla B\|_\infty^2 + \frac{(k_{c1}+k_{c2})3\sqrt{3}}{16\sqrt{\gamma_c}}\overline{\|\Delta\|} + \frac{\iota_x}{\underline{\Gamma}}$, $\iota_a := \frac{1}{2}\overline{\|\nabla V^* G_R \nabla \phi^T\|} + k_{a2}\overline{\|W\|} + \frac{(k_{c1}+k_{c2})3\sqrt{3}}{64\sqrt{\gamma_c}}\overline{\|G_\phi\|\|W\|}^2 + \iota_x$, $\iota_x := \overline{\|\nabla W f\|} + \frac{1}{2}\overline{\|\nabla W G_R \nabla \phi^T W\|} + \frac{c_b}{2}\overline{\|\nabla W G_R\|}\|\nabla B^T\|_\infty$. The entries in the matrix M from (25) are defined as $\varphi_a := \frac{(k_{c1}+k_{c2}3\sqrt{3})}{64\sqrt{\gamma_c}}\overline{\|G_\phi\|\|W\|} + \frac{1}{2}\overline{\|\nabla W G_R \nabla \phi^T\|}$, $\varphi_{ac} := k_{a1} + \frac{(k_{c1}+k_{c2})3\sqrt{3}}{64\sqrt{\gamma_c}}\overline{\|G_\phi\|\|W\|} + \frac{c_b k_{c1}3\sqrt{3}}{32\sqrt{\gamma_c}}\overline{\|G_R \nabla \phi^T\|}\|\nabla B\|_\infty + \frac{1}{2\underline{\Gamma}}\overline{\|\nabla W G_R \nabla \phi^T\|}$, $\varphi_{c\theta} := \frac{(k_{c1}+k_{c2})3\sqrt{3}}{16\sqrt{\gamma_c}}\overline{\|W \nabla \phi Y\|}$.