2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Approximate Optimal Control for Safety-Critical Systems with Control Barrier Functions

Max H. Cohen and Calin Belta

Abstract— **Control Barrier Functions (CBFs) have become a popular tool for enforcing set invariance in safety-critical control systems. While guaranteeing safety, most CBF approaches are myopic in the sense that they solve an optimization problem at each time step rather than over a long time horizon. This approach may allow a system to get too close to the unsafe set where the optimization problem can become infeasible. Some of these issues can be mitigated by introducing relaxation variables into the optimization problem; however, this compromises convergence to the desired equilibrium point. To address these challenges, we develop an approximate optimal approach to the safety-critical control problem in which the cost of violating safety constraints is directly embedded within the value function. We show that our method is capable of guaranteeing both safety and convergence to a desired equilibrium. Finally, we compare the performance of our method with that of the traditional quadratic programming approach through numerical examples.**

I. INTRODUCTION

The concept of safety has received much attention in the fields of robotics and controls over the past few years. One of the prime reasons for this is the rise of autonomy for safety-critical systems such as self-driving cars. This has led to the question: how does one formally define what it means to be safe? Informally speaking, one could define safety as something bad never happens; however, more formal definitions of safety have been linked to the concept of set invariance [1]. A popular technique for enforcing set invariance in safety-critical systems is the Control Barrier Function (CBF) approach [2], [3]. These methods typically involve synthesizing a safe controller by embedding set invariance conditions within an optimal control problem. Rather than solving a general constrained optimal control problem however, most papers propose to discretize time and assume a piecewise constant control. If the control system is affine in controls and the cost is quadratic, the problem reduces to solving a quadratic program (QP) at each time step to obtain the optimal control [3], [4].

One issue with the QP-based approach is that it operates myopically, that is, the safe control is only a function of the current state [5]. While this approach can guarantee local safety at each time step, the satisfaction of the safety constraint is dependent on how frequently the QP is solved [6]. A step size too small can induce unnecessary computation whereas a step size too large can result in unsafe behavior. Additionally, the QP may allow trajectories to approach the boundary of the safe set very closely before intervening. Consequently, when the system approaches the boundary of the safe set the QP may become infeasible and the approach fails [7]. The feasibility of the QP can be increased by introducing relaxation variables; however, this compromises convergence to the desired equilibrium point which may no longer be guaranteed [8]. Moreover, one must take care when simply merging stabilizing conditions with safety conditions as this can shift the desired equilibrium point of the closed-loop system [9].

If the CBF problem is not framed in terms of a QP then one is faced with the task of solving a general constrained optimal control problem. One way to obtain a solution to an optimal control problem is to solve the Hamilton-Jacobi-Bellman (HJB) equation; however, for many systems this involves solving a nonlinear partial differential equation (PDE) which typically does not have a closed-form solution [10]. The approach commonly taken is to numerically solve the HJB equation offline to generate a control policy which is then implemented on the system in real time. Along these lines, recent work has proposed using density functions, which are the dual to the value function in optimal control, to enforce safety [11]. It was shown in [5] that CBF constraints can be embedded within the density function and the resulting optimal control problem can be solved with a primal-dual algorithm. This approach addresses the myopic nature of the QP method; however, the solution is obtained by discretizing the state space and solving the HJB PDE offline which is computationally demanding. Other authors proposed using neural networks (NNs) to learn safe control policies subject to CBF constraints [12]; however, these results don't present stability guarantees and the solution is obtained offline. One issue with offline solutions is that they can become computationally demanding as the complexity of the system increases. Additionally, offline solutions are poorly-suited for safety-critical tasks as they are not robust to uncertainties in the system and environment. Therefore, there is a need for online solutions to the safety-critical optimal control problem.

Recently, reinforcement learning (RL) inspired methods such as approximate dynamic programming (ADP) have been proposed to approximately solve optimal control problems online (see [13], [14] for a survey). These methods utilize an actor-critic structure where the critic learns the optimal value function and the actor learns the optimal control input. This

This work was partially supported by the National Science Foundation under grant numbers DGE-1840990 and IIS-1723995. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The authors are with the Department of Mechanical Engineering, Boston University, 110 Cummington Mall, Boston, MA 02215, United States {maxcohen, cbelta}@bu.edu

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

method was used to solve infinite-horizon optimal regulation problems for nonlinear continuous-time systems online in [15] and more recent work has focused on various extensions [16], [17], [18], [19], [20], [21], [22]. In most literature the actor and critic are parameratized as NNs and although the solution is obtained online, the computational demands of the NNs may inhibit real-time implementation on physical systems. Because of this, other works have focused on developing computationally efficient approximation methods which are able to approximate functions in a local neighborhood of the current state [20]. These computationally efficient ADP methods have been successfully used in some safety-critical applications such as robot motion planning [21]; however, designing provably safe ADP controllers for general safety-critical systems is still an open area of research [22].

In this paper we present an ADP method to solve the safety-critical optimal control problem online in which safety-invariants are expressed as barrier functions. In Sec. II we introduce formal notions of safety used in the current literature and formulate the general problem under consideration. In Sec. III we reformulate the traditional problem as an unconstrained optimal control problem and show that the solution to this new problem guarantees satisfaction of the original constraints. Sec. IV provides an ADP solution to the reformulated problem from Sec. III and Sec. V presents a Lyapunov-based analysis in which the ADP method is shown to guarantee both convergence and safety of this solution. Finally, we provide numerical examples in Sec. VI and finish with concluding remarks in Sec. VII. In comparison to the current QP approach our method: 1) shows improved convergence to a stable equilibrium, 2) has increased feasibility, and 3) is not dependent on discretizing the time. To the best of our knowledge this is also the first attempt to use CBFs to design provably safe ADP controllers.

II. PRELIMINARIES AND PROBLEM FORMULATION

Throughout this paper we consider affine control systems of the form

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad x(0) = x_0, \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the system state, $f : \mathbb{R}^n \to \mathbb{R}^n$ models the system drift, the columns of $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ capture the control directions, $u(t) \in U \subset \mathbb{R}^m$ is the control input, and U denotes the control constraint set. Note that the explicit dependence on time will be dropped unless needed for clarity. We assume the functions f, g are locally Lipschitz continuous, $f(0) = 0$, f is sufficiently smooth and $0 < ||g(x)|| \leq \bar{g}$ with $\bar{g} \in \mathbb{R}_{>0}$ where $||\cdot||$ denotes the 2-norm. To formalize the concept of safety we introduce the following:

Definition 1 (Forward Invariance)**.** Consider a set $C \subseteq \mathbb{R}^n$ and initial condition $x(0) = x_0$. The set C is *forward invariant* for system (1) if $x_0 \in C \implies x(t) \in C, \forall t \geq 0$.

In the current literature [2], [3], [4], if a set C can be rendered forward invariant, then system (1) is said to be *safe* with respect to C. In this paper, we assume that the set[1] C is described by the superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ [2] such that

$$C = \{x \in \mathbb{R}^n \,|\, h(x) \geq 0\}, \tag{2a}$$
$$\partial C = \{x \in \mathbb{R}^n \,|\, h(x) = 0\}, \tag{2b}$$
$$\text{Int}(C) = \{x \in \mathbb{R}^n \,|\, h(x) > 0\}. \tag{2c}$$

Definition 2 (Control barrier function [4])**.** The function h in (2) is a *control barrier function* (CBF) for system (1) if there exists a class $\mathcal{K}$ function α such that

$$L_f h(x) + L_g h(x)u + \alpha\left(h(x)\right) \geq 0, \quad \forall x \in C, \tag{3}$$

where $L_f h(x) = \frac{dh}{dx} f(x)$ is the Lie derivative of h along f.

Theorem 1 ([2])**.** *Let C be defined as in* (2)*. If h is a CBF on C and $\frac{\partial h}{\partial x}(x) \neq 0$, $\forall x \in \partial C$ then any Lipschitz continuous controller $u(x) \in K_{cbf}(x)$ for* (1)*, where*

$$K_{cbf}(x) \triangleq \{u \in U \,|\, L_f h(x) + L_g h(x)u + \alpha\left(h(x)\right) \geq 0\}, \tag{4}$$

renders C forward invariant.

The above theorem illustrates that the existence of a CBF implies the safety of (1). However, given certain assumptions on C, it has been shown that CBFs provide necessary and sufficient conditions for safety, which is formalized through the following theorem:

Theorem 2 ([2])**.** *Let C be a compact set defined by* (2) *with the property that $\frac{\partial h}{\partial x}(x) \neq 0$, $\forall x \in \partial C$. If there exists a control law u that renders C forward invariant, then $h : C \to \mathbb{R}$ is a CBF on C.*

Definition 3 (Control Lyapunov Function [2])**.** A continuously differentiable function $V_{clf} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a *control Lyapunov function* (CLF) for (1) if it is positive definite and satisfies

$$\inf_{u\in U} \left[L_f V_{clf}(x) + L_g V_{clf}(x)u + \gamma(V_{clf}(x))\right] \leq 0, \tag{5}$$

where $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class $\mathcal{K}$ function.

Theorem 3 ([2])**.** *Given system* (1)*, if there exists a CLF $V_{clf}(x) \geq 0$ satisfying* (5)*, then any Lipschitz continuous feedback controller $u(x) \in K_{clf}(x)$ where*

$$K_{clf}(x) \triangleq \{u \in U \,|\, L_f V_{clf}(x) + L_g V_{clf}(x)u + \gamma(V_{clf}(x)) \leq 0\}, \tag{6}$$

asymptotically stabilizes the system to $x = 0$.

Now consider the cost functional

$$J(x, u) \triangleq \int_0^\infty r(x(\tau), u(\tau))d\tau, \tag{7}$$

where $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is an instantaneous positive definite cost. Consider the following problem:

[1]For a set C, the notation ∂C denotes the boundary of C and $\text{Int}(C)$ denotes its interior.

Problem 1. Consider system (1) with initial condition $x_0 \in \text{Int}(C)$. Find a control $u \in U$ that drives the system from x_0 to the origin, while minimizing (7) and keeping the system safe.

To solve Problem 1 existing works [3], [2], [4] propose to view (4) and (6) as constraints in an optimal control problem. Time is then discretized and the system state is assumed to be fixed at the start of each time interval. Consequently the constraints become linear in the control and, if r is quadratic in u, the problem reduces to solving a QP at each time step. This constant control is then applied to the continuous system (1) over the entire time interval and the procedure is repeated at each time step. Specifically, the QP solved is of the form

$$\min_{u \in U} \quad u^T R u + p\varphi^2 \tag{8a}$$

$$\text{s.t.} \quad L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0, \tag{8b}$$

$$L_f V_{clf}(x) + L_g V_{clf}(x) u \leq -\gamma(V_{clf}(x)) + \varphi, \tag{8c}$$

where $\varphi \in \mathbb{R}$ is a relaxation variable which is penalized by $p \in \mathbb{R}_{>0}$, $R \in \mathbb{R}^{m \times m}$ is the control penalty, and α, γ are the class $\mathcal{K}$ functions from (3) and (5), respectively. The relaxation variable is added to increase the feasibility of the QP which can easily become infeasible in the presence of conflicting control, stability, and safety constraints [7]. While increasing feasibility, this relaxation no longer guarantees convergence to the desired equilibrium point [8]. To address these issues we seek a solution to Problem 1 by formulating an optimal control problem whose solution satisfies the control, stability, and safety constraints without relying on the discretization of time. To this end we propose to augment the instantaneous cost r with additional terms whose minimization imply satisfaction of the original constraints.

III. PROBLEM REFORMULATION AND APPROACH

Consider Problem 1 with the cost functional in (7). Rather than dealing with a constrained problem we seek to reformulate Problem 1 as an unconstrained optimal control problem. To this end we redefine the instantaneous cost as

$$r(x,\, u) \triangleq x^T Q x + R_u(u) + B(x), \tag{9}$$

where $Q \in \mathbb{R}^{n \times n}$ is a positive definite matrix which penalizes the state, $R_u : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is a positive definite function which penalizes and ensures boundness of the control, and $B : \text{Int}(C) \to \mathbb{R}_{\geq 0}$ is a barrier-like function that satisfies

$$\inf_{x \in \text{Int}(C)} B(x) \geq 0, \quad \lim_{x \to \partial C} B(x) = \infty, \qquad B(0) = 0. \tag{10}$$

Based on (2), a choice of B which satisfies (10) is $B(x) = \frac{s(x)}{h(x)}$ where $s : \mathbb{R}^n \to [0,1]$ is a user-defined smooth scheduling function[2] that ensures trajectories are only penalized near ∂C. The state penalty matrix Q from (9) is positive defnite and hence satisfies $\underline{q}\|x\|^2 \leq x^T Q x \leq \overline{q}\|x\|^2$ with $\underline{q}, \overline{q} \in \mathbb{R}_{>0}$ for all $x \in \mathbb{R}^n$. Moreover, we assume the control constraint set U is defined by symmetric input constraints such that $U = \{u \in \mathbb{R}^m \mid -\overline{u} \leq u_i \leq \overline{u},\, i = 1, ..., m\}$, where u_i is the ith component of u and $\overline{u} \in \mathbb{R}_{>0}$ is the maximum allowable control. A popular approach to enforcing such control constraints is to use a non-quadratic control cost of the form [18], [23]

$$R_u(u) \triangleq 2 \sum_{i=1}^{m} \int_0^{u_i} \bar{u} r_i \tanh^{-1}(\zeta_i / \bar{u}) d\zeta_i, \tag{11}$$

where $r_i \in \mathbb{R}_{>0}$ are components that form a diagonal positive definite matrix $R \in \mathbb{R}^{m \times m}$ as $R \triangleq \text{diag}\{\bar{r}\}$ and $\bar{r} \triangleq [r_1, \ldots, r_m]^T$. If the time-horizon is infinite and the system and cost are time-invariant then the optimal value function $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is also time-invariant and can be expressed as $V^*(x) = \inf_{u(\tau) \in U} \int_t^\infty r(x(\tau), u(\tau)) d\tau$. The associated Hamiltonian is $H(x, u, \nabla V^*) = L_f V^*(x) + L_g V^*(x) u + r(x, u)$ which can be used with the stationary condition $\partial H / \partial u = 0$ to derive the optimal controller as

$$u^*(x) = -\bar{u} \text{Tanh}\left(\frac{R^{-1} g(x)^T}{2\bar{u}} \nabla V^*(x)^T \right), \tag{12}$$

where $\nabla(\cdot)$ denotes the derivative of $(\cdot)$ with respect to its first argument and $\text{Tanh}(\zeta) \triangleq [\tanh(\zeta_i), ..., \tanh(\zeta_m)]^T$, $\forall \zeta \in \mathbb{R}^m$. The optimal value function and controller satisfy the HJB equation

$$0 = \min_{u \in U} H = L_f V^*(x) + L_g V^*(x) u^* + r(x, u^*), \tag{13}$$

with a boundary condition of $V^*(0) = 0$.

Proposition 1. *Let the origin be contained in C and let $x_0 \in$ Int(C). Further, assume that there exists a smooth function $V^*(x) \geq 0$ which satisfies* (13)*. Then, the closed-loop system composed of* (1) *and controller* (12) *solves Problem 1.*

Proof. By definition V^* is positive definite and satisfies $V^*(0) = 0$, making it a suitable Lyapunov function candidate. Taking the derivative of V^* along the trajectories of (1) yields

$$\dot{V}^*(x) = L_f V^*(x) + L_g V^*(x) u^*(x) \leq -\underline{q}\|x\|^2, \tag{14}$$

where $L_f V^*(x) = -L_g V^*(x) u^*(x) - r(x, u^*)$ from (13) and $-x^T Q x - R_u(u^*) - B(x) \leq \underline{q}\|x\|^2$ from (10), (11) were used. Since V^* was used as a Lyapunov function candidate, it follows from (14) and [24, Theorem 4.1] that the origin is asymptotically stable for (1). Additionally, u^* maps from $\mathbb{R}^n \to (-\bar{u}, \bar{u})$ thus, the input constraints are satisfied. Now suppose $x_0 \in C$ and C is not forward invariant. Then $\exists\, \bar{t} \geq 0$ such that $x(\bar{t}) \to \partial C \implies B(x(\bar{t})) \to \infty \implies V^*(x(\bar{t})) \to \infty$ which contradicts (14). Thus, $\nexists\, t \geq 0$ for which $x \to \partial C$ so $x_0 \in C \implies x \in C, \forall t \geq 0$ and by Def. 1 C is forward invariant and (1) is safe. Moreover, if C is compact it follows from Theorem 2 that $h : C \to \mathbb{R}$ is a CBF for (1) over C and $u^*(x) \in K_{cbf}(x)$. □

Proposition 1 illustrates that the solution to the unconstrained infinite-horizon optimal control problem with a cost defined by (9) solves Problem 1; however, this is conditioned

[2]It is assumed that the smooth scheduling function is designed such that $s(0) = 0$. See [21] for examples of scheduling functions.

on solving the HJB equation (13) for V^*. Generally speaking, (13) is a nonlinear PDE which cannot be solved analytically. To address this issue, we propose an ADP approach in which the optimal value function is learned online.

IV. APPROXIMATE DYNAMIC PROGRAMMING

In the following, we develop a local approximation scheme and online update laws to learn the solution to the HJB equation online.

A. Value Function Approximation

Consider the compact set $\chi \subset \mathbb{R}^n$ with x in the interior of χ and let $\Omega(x)$ denote a small compact set centered at the current state x. The value function can be represented at points $y \in \Omega(x)$ using state following (StaF) kernels [20], [25] as

$$V^*(y) = W(x)^T \sigma\left(y,\, c(x)\right) + \epsilon(x, y), \tag{15}$$

where $W : \chi \to \mathbb{R}^L$ is the continuously differentiable ideal weight function, $\sigma : \chi \times \chi \to \mathbb{R}^L$ is a vector of $L \in \mathbb{N}$ continuously differentiable bounded positive definite kernel functions, and $c_i(x) \in \chi, i = 1, \ldots, L$, are the distinct centers of each kernel. The function $\epsilon : \chi \times \chi \to \mathbb{R}$ is the function approximation reconstruction error which is assumed to be bounded over χ. Adding and subtracting a bounded version of the barrier-like function (10), denoted as $\bar{B} : \text{Int}(C) \to \mathbb{R}$ where C is the set defined in (2), from (15), taking the gradient, and substituting into (12) yields an expression for the optimal policy as

$$u^*(y) = -\bar{u}\text{Tanh}\left(\frac{R^{-1}g(y)^T}{2\bar{u}} D^*(y)\right), \tag{16}$$

where $D^*(y) \triangleq \nabla\sigma\left(y,\, c\left(x\right)\right)^T W\left(x\right) + \nabla W(x)^T \sigma(y,\, c(x)) + \nabla\epsilon\left(x, y\right)^T + \nabla\bar{B}(y)^T$. The addition and subtraction of $\bar{B}$ is made to facilitate the analysis in Sec. V. If B is chosen as $B(x) = \frac{s(x)}{h(x)}$ then $\bar{B}$ can always be constructed as $\bar{B}(x) = \frac{s(x)}{h(x)+a}$ where $a \in \mathbb{R}_{>0}$ is a positive constant.

In general, the ideal weight function W is unknown a priori and must be replaced with an estimated weight function $\hat{W}(t) \in \mathbb{R}^L$. Similar to most ADP approaches, we maintain separate weight estimates for the value function and optimal policy, denoted as $\hat{W}_c(t), \hat{W}_a(t) \in \mathbb{R}^L$, respectively. Using these estimated weights in the StaF parameterizations of the value function (15) and optimal policy (16) results in the approximate value function

$$\hat{V}(y, x, \hat{W}_c) \triangleq \hat{W}_c^T \sigma(y,\, c(x)) + \bar{B}(y), \tag{17}$$

and approximate optimal policy

$$\hat{u}(y, x, \hat{W}_a) \triangleq -\bar{u}\text{Tanh}\left(\frac{R^{-1}g(y)^T}{2\bar{u}} \hat{D}(y, x, \hat{W}_a)\right), \tag{18}$$

where $\hat{D}(y, x, \hat{W}_a) \triangleq \left(\nabla\sigma(y,\, c(x))^T \hat{W}_a + \nabla\bar{B}(y)^T\right)$. The notation $\hat{V}(y, x, \hat{W}_c)$ denotes the approximate value function evaluated at y, using a kernel centered at x, with a weight estimate of $\hat{W}_c$. The expressions for the approximate optimal value function and policy in (17) and (18) can then be substituted into (13) to obtain an expression for the approximate HJB equation as

$$\hat{H}(y, x, \hat{W}_c, \hat{W}_a) = r(y, \hat{u}(y, x, \hat{W}_a)) + L_f\hat{V}(y, x, \hat{W}_c) + L_g\hat{V}(y, x, \hat{W}_c)\hat{u}(y, x, \hat{W}_a), \tag{19}$$

where $\hat{H} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$ is the approximate Hamiltonian. Taking the difference between the approximate and optimal Hamiltonian as $\delta(y, x, \hat{W}_c, \hat{W}_a) \triangleq \hat{H}(y, x, \hat{W}_c, \hat{W}_a) - H\left(x, u^*, \nabla V^*\right)$ yields the residual approximation error δ, referred to as the Bellman error (BE). From (13), $H\left(x, u^*, \nabla V^*\right) = 0$, thus the BE is just the approximate Hamiltonian. From Proposition 1, if $\hat{V} \to V^*$ and $\hat{u} \to u^*$, then implementing $\hat{u}$ on (1) will solve Problem 1. Thus, we are faced with the problem of developing estimates of the ideal weights $\hat{W}_c, \hat{W}_a$ that minimize the BE.

B. Online Learning

In this section we develop online update laws for the estimated weights that ensure convergence to their ideal values. In traditional ADP approaches [15], [16] a persistence of excitation (PE) condition is required to ensure convergence of the weight estimates; however, this typically involves adding an exploration signal into the system. In addition to degrading performance, the introduction of an exploration signal could compromise safety. More recent works [19] have leveraged techniques from concurrent learning adaptive control [26] in the form of BE extrapolation which allows the BE to be evaluated at unexplored regions of the state-space. This extrapolation results in a virtual excitation of the system which facilitates weight estimate convergence [19]. To this end, at each time step the BE is extrapolated to a set of points $\{x_k(t) \in \Omega(x(t)) \,|\, k = 1, ..., N\}$ about the current state $x(t)$. In the following, let $\delta(t) \triangleq \delta(x(t), x(t), \hat{W}_c(t), \hat{W}_a(t))$ and let the subscript k denote that a function is evaluated at the extrapolated state $x_k(t)$, i.e. $\delta_k(t) \triangleq \delta(x_k(t), x(t), \hat{W}_c(t), \hat{W}_a(t))$. Additionally, let the control $u(t) \triangleq \hat{u}(x(t), x(t), \hat{W}_a(t))$ be the input that drives (1). For notational brevity, the BE can be expressed more compactly as $\delta(t) = \hat{W}_c(t)^T\omega(t) + r(x(t), u(t)) + \omega_B(t)$ where $\omega(t) \triangleq \nabla\sigma(x(t), c(x(t))(f(x(t)) + g(x(t))u(t)$, $\omega_B(t) \triangleq \nabla\bar{B}(x(t))(f(x(t)) + g(x(t))u(t))$. To derive an update law for $\hat{W}_c$ consider a squared, normalized version of the BE as $E(t) \triangleq \frac{1}{2}\left(\frac{k_{c1}\delta^2(t)}{\rho^2(t)} + \sum_{k=1}^{N} \frac{k_{c2}\delta_k^2(t)}{N\rho_k^2(t)}\right)$ where $k_{c1}, k_{c2} \in \mathbb{R}_{>0}$ are gains and ρ is a normalization term which is defined as $\rho(t) \triangleq 1 + \nu\omega(t)^T\omega(t)$, where $\nu \in \mathbb{R}_{>0}$ is a gain. An update law is obtained using a gradient descent approach as $\dot{\hat{W}}_c(t) = -\Gamma(t)\frac{\partial E}{\partial \hat{W}_c}(t)$, which yields

$$\dot{\hat{W}}_c(t) = -\Gamma(t)\left(k_{c1}\frac{\omega(t)}{\rho^2(t)}\delta(t) + \frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{\omega_k(t)}{\rho_k^2(t)}\delta_k(t)\right), \tag{20}$$

where $\Gamma(t) \in \mathbb{R}^{L\times L}$ is a gain matrix that is updated according to

$$\dot{\Gamma}(t) = \beta\Gamma(t) - \Gamma(t)\left(k_{c1}\Lambda(t) + \frac{k_{c2}}{N}\sum_{k=1}^{N}\Lambda_k(t)\right)\Gamma(t), \tag{21}$$

where $\Lambda(t) \triangleq \frac{\omega(t)\omega(t)^T}{\rho^2(t)}$, $\Lambda_k(t) \triangleq \frac{\omega_k(t)\omega_k(t)^T}{\rho_k^2(t)}$, and $\beta \in \mathbb{R}_{>0}$ is a gain. Based on the analysis in Sec. V, the update law for $\hat{W}_a$ is selected as

$$\dot{\hat{W}}_a(t) = \text{proj}\{-k_{a1}(\hat{W}_a(t) - \hat{W}_c(t))\}, \tag{22}$$

where $k_{a1} \in \mathbb{R}_{>0}$ is a learning gain and proj$\{\cdot\}$ is a smooth operator[3] which bounds the weight estimates. We make the following assumption to ensure weight estimate convergence:

Assumption 1 ([20])**.** There exists constants $\underline{c}_1, \underline{c}_2, \underline{c}_3 \in \mathbb{R}_{\geq 0}$, $T \in \mathbb{R}_{>0}$ such that 1) $\underline{c}_1 I_L \leq \frac{1}{N}\sum_{k=1}^{N}\Lambda_k(t)$, 2) $\underline{c}_2 I_L \leq \int_t^{t+T}\left(\frac{1}{N}\sum_{k=1}^{N}\Lambda_k(\tau)\right)d\tau$, $\forall t \in \mathbb{R}_{\geq 0}$, 3) $\underline{c}_3 I_L \int_t^{t+T}(\Lambda(\tau))\,d\tau$, $\forall t \in \mathbb{R}_{\geq 0}$ where at least one of $\underline{c}_i$, $i = 1,2,3$ is strictly positive[4].

If $\lambda_{\min}\{\Gamma^{-1}(0)\} > 0$ and Assumption 1 holds, (21) can be used to show that Γ satisfies $\underline{\Gamma} I_L \leq \Gamma(t) \leq \overline{\Gamma} I_L$, $\forall t \in \mathbb{R}_{\geq 0}$ where $\underline{\Gamma}, \overline{\Gamma} \in \mathbb{R}_{>0}$ and $\lambda_{\min}\{(\cdot)\}$ denotes the minimum eigenvalue of $(\cdot)$ [20, Lemma 1].

V. ANALYSIS

To aid in the analysis, we define the ideal weight estimate errors as $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$. Now consider the Lyapunov function candidate

$$V_L(Z,t) \triangleq V^* + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a \tag{23}$$

and let $Z \triangleq [x^T\ \tilde{W}_c^T\ \tilde{W}_a^T]^T$. Note that the value function is positive definite, thus the Lyapunov function candidate is positive definite and can be bounded as $\eta_1(||Z||) \leq V_L(Z,t) \leq \eta_2(||Z||)$ where $\eta_1, \eta_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions [24, Lemma 4.3]. The sufficient conditions for the following theorem are $\psi k_{c2} > k_{a1}$, $\eta^{-1}(\kappa) < \eta_2^{-1}(\eta_1(\xi))$, $x_0 \in \text{Int}(C)$ where $\xi \in \mathbb{R}_{>0}$ is the radius of the compact set used for value function approximation, $\psi \triangleq \left(\frac{\beta}{2k_{c2}\overline{\Gamma}} + \frac{c_1}{2}\right)$, $\kappa \in \mathbb{R}_{>0}$ is a known positive constant that depends on the gains, and $\eta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class $\mathcal{K}$ function that satisfies

$$\eta(\|Z\|) \leq \frac{q}{2}\|x\|^2 + \frac{k_{a1}}{8}\left\|\tilde{W}_a\right\|^2 + \frac{k_{c2}}{4}\psi\left\|\tilde{W}_c\right\|^2.$$

Theorem 4 (Convergence and Safety)**.** *Given system* (1) *under controller* (18) *with update laws* (20)*,* (21)*,* (22)*, if Assumption 1 holds and the sufficient conditions are satisfied then C is forward invariant and the state x and weight estimation errors $\tilde{W}_c$, $\tilde{W}_a$ are uniformly ultimately bounded.*

Proof. Omitted due to space constraints. Available upon request. □

[3]Details on the projection operator can be found in [27].

[4]The notation I_L denotes an $L \times L$ identity matrix.

VI. NUMERICAL EXAMPLES

In this section we present simulation results which were performed to assess the efficacy of our method and to compare it with the traditional QP approach. In the following, the system is simulated for 25 seconds under the influence of each controller. All differential equations are solved using Matlab's `ode45` function and (8) is solved using Matlab's `quadprog` function. Consider a two dimensional single integrator which can be represented as (1) with $x \in \mathbb{R}^2$, $f = 0_{2\times 1}$, and $g = I_2$. The safe set is defined by (2) with

$$h(x) \triangleq \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2} - r_h, \tag{24}$$

where $z = [z_1\ z_2]^T$ denotes the center of the circular set, and $r_h \in \mathbb{R}_{>0}$ is its radius. For the approximate optimal controller we select the gains as $k_{c1} = 0.05$, $k_{c2} = 0.75$, $k_{a1} = 0.75$, $\nu = 1$, $\beta = 0.001$. The cost function parameters are set to $Q = I_2$, $R = 10I_2$ and the controller saturation is $\bar{u} = 0.5$. The initial weights for the update laws are selected randomly from a uniform distribution between 0 and 4. The kernel function is defined by $\sigma(x, c(x)) = \left[x^T c_1(x)\ x^T c_2(x)\ x^T c_3(x)\right]^T$ and we select the centers to be at the vertices of an equilateral triangle such that $c_i(x) = x + \vartheta(x)d_i$ where ϑ is a scaling factor defined as $\vartheta = \frac{0.5x^Tx}{1+x^Tx}$ and $d_1 = [0\ -1]^T$, $d_2 = [0.866\ -0.5]^T$, $d_3 = [-0.866\ -0.5]^T$ are the center offsets. To facilitate the finite excitation condition for weight convergence the BE is extrapolated to 1 random point from a $0.1\vartheta(x) \times 0.1\vartheta(x)$ uniform distribution centered about x at each time step. We select the barrier-like function as $B(x) = \frac{k_p s(x)}{h(x)}$ where s is a smooth scheduling function and $k_p \in \mathbb{R}_{>0}$ is a gain. For the QP in (8) we define the CLF as $V_{clf}(x) = x^TQx$. The functions α, γ, and p act as tuning parameters and are selected as $\alpha(h(x)) = h(x)$, $\gamma(V_{clf}(x)) = 10V_{clf}(x)$, and $p = 2$. To ensure results are comparable between methods we select Q, R, and $\bar{u}$ to be the same as in the ADP case. The results from applying each controller are shown in Fig. 1-2. Fig. 1 illustrates each controller's ability to remain in C; however, the QP controller is incapable of converging to the origin. This behavior is further illustrated in Fig. 2 which is a result of introducing the relaxation variable φ to ensure solvability of the QP. The gains on the CLF and relaxation variable can be tuned in an attempt to achieve better convergence; however, for no finite value of relaxation penalty p can one ensure convergence to the equilibrium point [8]. Fig. 2 illustrates each controller's ability to satisfy the input and safety constraints.

VII. CONCLUSION

We presented an alternative to the QP-based CBF approach to synthesizing optimal controllers for safety-critical systems. Instead, our method is based on ADP where we incorporate the cost of safety violation directly into the value function of an optimal control problem. We showed that the ADP method is able to guarantee both safety and stability of the resulting closed-loop system. We further illustrated this result with numerical examples in which the ADP controller outperformed

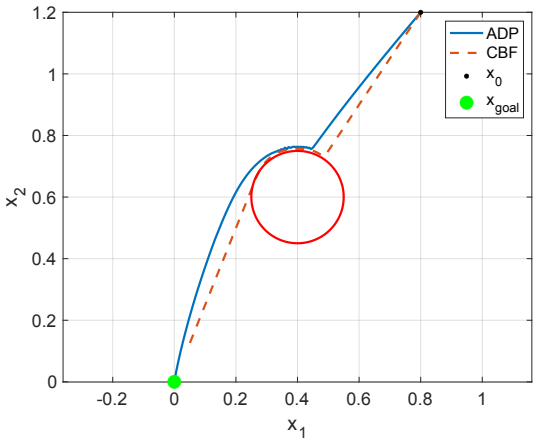

Fig. 1. Trajectory of the system under ADP controller and QP controller. The boundary of C is represented by the orange circle, and the origin is represented by a green dot.

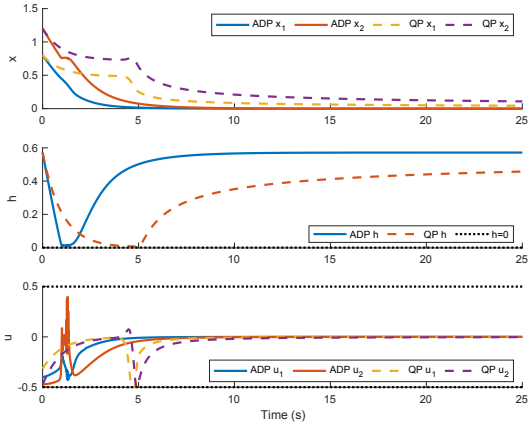

Fig. 2. System states under each controller (top). Evolution of the barrier function under each controller (middle). Control trajectory over the course of the simulation (bottom).

the traditional QP controller in terms of convergence and feasibility. Future work will explore extending our approach to uncertain systems and differential games.

REFERENCES

[1] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[2] 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.*, pp. 3420–3431, 2019.

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

[4] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. Conf. Decis. Control*, pp. 474–479, 2019.

[5] Y. Chen, M. Ahmadi, and A. D. Ames, "Optimal safe controller synthesis: a density function approach," in *Proc. Amer. Control Conf.*, pp. 5407–5412, 2020.

[6] G. Yang, C. Belta, and R. Tron, "Self-triggered control for safety critical systems using control barrier functions," in *Proc. Amer. Control Conf.*, pp. 4454–4459, 2019.

[7] W. Xiao, C. Belta, and C. G. Cassandras, "Feasibility-guided learning for robust control in constrained optimal control problems," in *Proc. Conf. Decis. Control (to appear)*, 2020. preprint available at arXiv:1912.04066.

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

[9] M. F. Reis, A. P. Aguiar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Contr. Syst. Lett.*, vol. 5, no. 2, pp. 731–736, 2020.

[10] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2011.

[11] Y. Chen and A. D. Ames, "Duality between density function and value function with applications in constrained optimal control and markov decision process," *arXiv preprint arXiv:1902.09583*, 2019.

[12] J. V. Deshmukh, J. P. Kapinski, T. Yamaguchi, and D. Prokhorov, "Learning deep neural network controllers for dynamical systems with safety guarantees," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, 2019.

[13] 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 Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[14] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, 2017.

[15] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous–time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[16] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, 2013.

[17] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous–time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, 2015.

[18] 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 Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2386–2398, 2015.

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

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

[21] P. Deptula, H. Chen, R. Licitra, J. A. Rosenfeld, and W. E. Dixon, "Approximate optimal motion planning to avoid unknown moving avoidance regions," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 414–430, 2020.

[22] Y. Yang, K. G. Vamvoudakis, H. Modares, W. He, Y. Yin, , and D. Wunsch, "Safety-aware reinforcement learning framework with an actor-critic-barrier structure," in *Proc. Amer. Control Conf.*, pp. 2352–2358, 2019.

[23] S. E. Lyshevski, "Optimal control of nonlinear continuous-time systems: Design of bounded controllers via generalized nonquadratic cost functionals," in *Proc. Amer. Control Conf.*, pp. 205–209, 1998.

[24] H. K. Khalil, *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, 2002.

[25] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "The state following (staf) approximation method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1716–1730, 2019.

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

[27] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.