Model-Based Safe Reinforcement Learning With Time-Varying Constraints Applications to Intelligent Vehicles

Zhang, Xinglong; Peng, Yaoqian; Luo, Biao; Pan, Wei; Xu, Xin; Xie, Haibin

DOI
10.1109/TIE.2023.3317853

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Industrial Electronics

Model-Based Safe Reinforcement Learning With Time-Varying Constraints: Applications to Intelligent Vehicles

Xinglong Zhang, *Member, IEEE*, Yaoqian Peng, Biao Luo, *Senior Member, IEEE*, Wei Pan, *Member, IEEE*, Xin Xu, *Senior Member, IEEE*, and Haibin Xie

***Abstract*—In recent years, safe reinforcement learning (RL) with the actor-critic structure has gained significant interest for continuous control tasks. However, achieving near-optimal control policies with safety and convergence guarantees remains challenging. Moreover, few works have focused on designing RL algorithms that handle time-varying safety constraints. This article proposes a safe RL algorithm for optimal control of nonlinear systems with time-varying state and control constraints. The algorithm's novelty lies in two key aspects. Firstly, the approach introduces a unique barrier force-based control policy structure to ensure control safety during learning. Secondly, a multistep policy evaluation mechanism is employed, enabling the prediction of policy safety risks under time-varying constraints and guiding safe updates. Theoretical results on learning convergence, stability, and robustness are proven. The proposed algorithm outperforms several state-of-the-art RL algorithms in the simulated Safety Gym environment. It is also applied to the real-world problem of integrated path following and collision avoidance for two intelligent vehicles—a differential-drive vehicle and an Ackermann-drive one. The experimental results demonstrate the impressive sim-to-real transfer capability of our approach, while showcasing satisfactory online control performance.**

***Index Terms*—Barrier force, multistep policy evaluation, safe reinforcement learning (RL), time-varying constraints.**

Manuscript received 12 June 2023; revised 15 August 2023; accepted 5 September 2023. Date of publication 4 January 2024; date of current version 19 June 2024. This work was supported by the National Natural Science Foundation of China under Grant 61825305, Grant 62003361, and Grant 62022094. *(Corresponding authors: Xinglong Zhang; Xin Xu.)*

Xinglong Zhang, Yaoqian Peng, Xin Xu, and Haibin Xie are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: zhangxinglong18@nudt.edu.cn; pengyq20@nudt.edu.cn; xinxu@nudt.edu.cn; xiehaibin@nudt.edu.cn).

Biao Luo is with the School of Automation, Central South University, Changsha, Hunan 410083, China (e-mail: biao.luo@csu.edu.cn).

Wei Pan is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: wei.pan@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2023.3317853.

Digital Object Identifier 10.1109/TIE.2023.3317853

I. Introduction

REINFORCEMENT learning (RL) is promising for solving nonlinear optimal control problems [1]. Until recently, significant progress has been made on RL with the actor-critic structure for continuous control tasks [2]. In actor-critic RL, the value function and control policy are represented by the critic and actor networks, respectively, and learned via extensive policy exploration and exploitation. However, the resulting learning-based control system might not guarantee safety for systems with state and stability constraints. It is known that safety constraint satisfaction is crucial besides optimality in many real-world robot control applications [3], [4]. For instance, autonomous driving has been viewed as a promising technology that will bring fundamental changes to everyday life. Still, one of the crucial issues concerns how to learn to drive safely under dynamic and unknown environments with unexpected obstacles [5]. For these practical reasons, many safe RL algorithms have been recently developed for safety-critical systems (see, e.g., [6], [7], [8], [9], [10], [11], and the references therein).

In general, current safe RL solutions can be categorized into the following three main approaches. 1) The first family utilizes a unique mechanism in the learning procedure for safe policy optimization using, e.g., control barrier functions [9], formal verification [12], shielding [13], and external intervention [14]. These methods are prone to safety-biased learning by sacrificing greatly on performance, and some of them rely on extra human interference [14]. 2) The second family proposes safe RL algorithms via primal-dual methods [6]. In the resulting optimization problem, the Lagrangian multiplier serves as an extra weight whose update is sensitive to the control performance [6]. 3) The third is reward/cost shaping-based RL approaches [15] where the cost functions are augmented with various safety-related parts, e.g., barrier functions. As stated in [16], such a design only informs the goal of guaranteeing safety by minimizing the reshaped cost function but fails to guide how to achieve it well through an actor-critic structure design. The weights of actor and critic networks are prone to divergence in the training process, especially when the control and safety goals are conflicting. These issues motivated our novel actor-critic structure with barrier functions and gradients. In this work, we incorporate this unique structure into a control-theory-based

RL framework, where model-based multistep policy evaluation mechanism is utilized to ensure convergence and safety in online learning scenarios. Moreover, few works have addressed the safe RL algorithm design under time-varying safety constraints.

This work proposes a model-based safe RL algorithm with theoretical guarantees for optimal control with time-varying state and control constraints. First, a new barrier force-based control policy (BCP) structure is constructed to ensure control safety during learning and enhance generalization performance. Second, the time-varying constraints are addressed by a multistep policy evaluation (MPE). The proposed safe RL approach is implemented by an online barrier force-based actor-critic learning algorithm. The closed-loop theoretical property of our approach under nominal and perturbed cases and the convergence condition of the barrier-based actor-critic (BAC) learning algorithm is derived. The effectiveness of our approach is tested on both simulations and real-world intelligent vehicles.

Our contributions are summarized as follows.

1) We proposed a safe RL algorithm for optimal control under time-varying constraints. Safety can be guaranteed in both online and offline learning scenarios. The performance and advantages of our approach are achieved by a barrier force-based policy shaping method to ensure safety and generalization performance, and a multistep evaluation mechanism to guide policy to update safely under time-varying constraints.

2) We proved that the proposed safe RL algorithm could guarantee stability and robustness in the nominal scenario and under external disturbances. Also, the convergence condition of the actor-critic learning algorithm was derived by the Lyapunov method.

3) Our approach was applied to solve a path following and collision avoidance problem of intelligent vehicles. a) Extensive simulation results illustrate that our approach outperforms other state-of-the-art safe RL methods in learning safety and performance. b) We verified our approach's offline sim-to-real transfer capability and real-world online learning performance, as well as the strengths to state-of-the-art model predictive control (MPC) algorithms.

The rest of this article is organized as follows. Section II introduces the considered control problem. Section III presents the proposed safe RL approach, while Section IV presents the main theoretical results. Section V shows the real-world experimental results. Finally, Section VI concludes the article. For space limitations, some proofs of the theoretical results and additional experimental results are given in our extended version [17].

Notation: We denote $\mathbb{N}$ and $\mathbb{N}_a^b$as the set of natural numbers and integers $a, \ldots, b$. For a vector $x \in \mathbb{R}^n$, we denote $\|x\|_Q^2$ as $x^\top Q x$ and $\|x\|$ as the Euclidean norm. For a function $f(x,u)$ with arguments x and u, we denote $\nabla_z f(x,u)$ as the partial gradient to z, $z = x$ or u. Given a matrix $A \in \mathbb{R}^{n\times n}$, we use $\lambda_{\min}(A)$ ($\lambda_{\max}(A)$) to denote the minimal (maximal) eigenvalues. We denote $\mathrm{Int}(\mathcal{Z})$ as the interior of a set $\mathcal{Z}$. For variables $z_i \in \mathbb{R}^{q_i}$, $i \in \mathbb{N}_1^M$, we define $(z_1, z_2, \ldots, z_{\mathrm{M}}) = [\, z_1^\top \; z_2^\top \; \cdots \; z_{\mathrm{M}}^\top \,]^\top \in \mathbb{R}^q$, where $q = \sum_{i=1}^M q_i$.

II. Problem Formulation

In this section, we describe the considered model and constraints, the optimal control objective, and the safe RL problem formulation using cost reconstruction with barrier functions.

A. System Model and Constraints

The considered system under control is a class of discrete-time nonlinear systems described by

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where $x_k \in \mathcal{X}_k \subseteq \mathbb{R}^n$ and $u_k \in \mathcal{U}_k \subseteq \mathbb{R}^m$ are the state and input variables, k is the discrete-time index, $\mathcal{X}_k = \{x \in \mathbb{R}^n | G_{x,k}^i(x) \le 0,\ \forall i \in \mathbb{N}_1^{p_x}\}$ and $\mathcal{U}_k = \{u \in \mathbb{R}^m | G_{u,k}^i(u) \le 0,\ \forall i \in \mathbb{N}_1^{p_u}\}$ are time-varying constraints, $\{0\} \subseteq \mathcal{U}_k, \forall\, k \in \mathbb{N}$; functions $G_{z,k}^i(z) \in \mathbb{R}$ for $z = x, u$, are assumed to be C^2; f is a smooth state transition function and $f(0,0) = 0$.

In principle, different types of state constraints can be formalized as follows. For instance, 1) $\mathcal{X}_k$ with $G_{x,k}^i(x) = E_k^i x - c_k^i$ is a linear convex set, where $E_k^i \in \mathbb{R}^{1\times n}$ and $c_k^i \in \mathbb{R}$ are time-varying parameters; 2) $\mathcal{X}_k$ with $G_{x,k}^i(x) = d_k^i - \|E_k^i x_k - c_k^i\|$ represents a dynamic obstacle avoidance constraint of a robot in a 2-D map, where $E_k^i \in \mathbb{R}^{2\times n_i}$, $c_k^i \in \mathbb{R}^2$ and $d_k^i \in \mathbb{R}$ are the center and radius of the circular dynamic obstacle respectively.

Definition 1 (Local stabilizability [18]): System (1) is stabilizable on $\mathcal{X}_k \times \mathcal{U}_k$ if, for any $x_0 \in \mathcal{X}_0$, there exists a C^1 state-feedback policy $\pi := \{u(x_k)\}_{k=0}^\infty$ with $u(x_k) \in \mathcal{U}_k\ \forall\, k \in \mathbb{N}_0^\infty$, such that $x_k \in \mathcal{X}_k$ and $x_k \to 0$ as $k \to +\infty$.

Assumption 1 (Lipschitz continuous): Model (1) is Lipschitz continuous in $\mathcal{X}_k \times \mathcal{U}_k$, for all $k \in \mathbb{N}_0^\infty$, i.e., there exists a Lipschitz constant $0 < L_f < +\infty$ such that for all $x_1, x_2 \in \mathcal{X}_k$ and C^1 control policies with $u(x_1), u(x_2) \in \mathcal{U}_k$

$$\|f(x_1, u(x_1)) - f(x_2, u(x_2))\| \le L_f \|x_1 - x_2\|. \tag{2}$$

Assumption 2 (Model): $\|\nabla_u f(x,u)\| \le g_m$ in the domain $\mathcal{X}_k \times \mathcal{U}_k$, where g_m is a positive scalar.

B. Control Objective

Given any initial condition $x_0 \in \mathcal{X}_0$, the control objective is to find an optimal control policy π^* that minimizes

$$J(x_0, u_{0:+\infty}) = \sum_{k=0}^{+\infty} \gamma^k r(x_k, u_k) \tag{3}$$

subject to model (1), $x_k \in \mathcal{X}_k$, and $u_k \in \mathcal{U}_k$, $\forall k \in \mathbb{N}$; where $r(x_k, u_k) = \|x_k\|_Q^2 + \|u_k\|_R^2$, and $Q = Q^\top \in \mathbb{R}^{n\times n}$, $R = R^\top \in \mathbb{R}^{m\times m}$, $Q, R \succ 0$, γ is a discounting factor.

Note that, many waypoint tracking problems in the robot control field can be naturally formed as the above regulation problem (3), by proper coordinate transformation. Generally, it is allowed that the time-varying state constraint might not contain the origin for some $k \in \mathbb{N}$, e.g., in a collision avoidance scenario.

It is still reasonable to introduce the following assumption for convergence guarantee.

Assumption 3 (State constraint): There exists a finite number $\bar{k} \in \mathbb{N}$ such that $\{0\} \subseteq \mathcal{X}_k$ as $k \geq \bar{k}$.

Definition 2 (Multistep safe control): For a given state $x_k \in \mathcal{X}_k$ at time instant k, a control policy $\pi_k := \{u(x_{k+l})\}_{l=0}^{\infty}$ with $u(x_{k+l}) \in \mathcal{U}_{k+l}$, is L-step safe for (1) if the resulting future state evolutions of (1) satisfy $x_{k+l} \in \mathcal{X}_{k+l}^u$, $\forall l \in \mathbb{N}_1^L$, where $\mathcal{X}_{k+l}^u$ is the resulting state constraint under π_k.

To simplify the notation, in the rest of the article, the super index in $\mathcal{X}_k^u$ is neglected, i.e., we use $\mathcal{X}_k$ to denote $\mathcal{X}_k^u$.

C. Cost Reconstruction With Barrier Functions

As policy improvement is usually performed by the gradient descent method in actor-critic RL, we have to reconstruct the cost function in (3) by incorporating continuous barrier functions of state and control constraints. To this end, we first introduce a definition of barrier functions as follows.

Definition 3 (Barrier function [19]): For a general convex set $\mathcal{Z}_k = \{z \in \mathbb{R}^l | G_{z,k}^i(z) \leq 0, \forall i \in \mathbb{N}_1^{p_z}\}$, a barrier function is defined as

$$\mathcal{B}_k^o(z) = \begin{cases} -\sum_{i=1}^{p_z} \log\left(-G_{z,k}^i(z)\right), & z \in \mathrm{Int}(\mathcal{Z}_k) \\ +\infty \ \text{otherwise.} \end{cases} \tag{4}$$

To derive a satisfactory control performance, we define a re-centered transformation of $\mathcal{B}_k^o(z)$ centered at $z_c \in \mathbb{R}^l$ is defined as $\mathcal{B}_k^c(z) = \mathcal{B}_k^o(z) - \mathcal{B}_k^o(z_c) - \nabla_z \mathcal{B}_k^o(z_c)^\top z$, where $z_c = 0$ if $\{0\} \subseteq \mathcal{Z}_k$ or z_c is selected such that $z_c \in \mathcal{Z}_k$ otherwise. This definition leads to the property that $\mathcal{B}_k^c(z) \geq 0$ and it reaches the minimum at z_c, i.e., $\mathcal{B}_k^c(z_c) = 0$, $\nabla \mathcal{B}_k^c(z_c) = 0$. For the case $\{0\} \nsubseteq \mathcal{Z}_k$, we suggest selecting z_c far from $\mathrm{Int}(\mathcal{Z}_k)$ and as the central point or its neighbor of $\mathcal{Z}_k$ (if possible).

Lemma 1 (Relaxed barrier function [19]): Define a relaxed barrier function of $\mathcal{B}_k^c(z)$ as

$$\mathcal{B}_k(z) = \begin{cases} \mathcal{B}_k^c(z) & \bar{\sigma}_k \geq \kappa_b \\ \gamma_b(z, \bar{\sigma}_k) & \bar{\sigma}_k < \kappa_b \end{cases} \tag{5}$$

where the relaxing factor $\kappa_b > 0$ is a small positive number, $\bar{\sigma}_k = \min_{i \in \mathbb{N}_1^{p_z}} -G_k^i(z)$, the function $\gamma_b(z, \bar{\sigma}_k)$ is strictly monotone and differentiable on $(-\infty, \kappa_b)$, and $\nabla_z^2 \gamma_b(z, \bar{\sigma}_k) \leq \nabla_z^2 \mathcal{B}_k(z)|_{\bar{\sigma}_k = \kappa_b}$, then there exists a matrix $H_{z_k} \geq \nabla_z^2 \mathcal{B}_k(z)|_{\bar{\sigma}_k = \kappa_b}$, such that $\|\nabla_z \mathcal{B}_k(z)\| \leq \mathcal{B}_{z_k,m}$, $\mathcal{B}_{z_k,m} = \max_{z \in \mathcal{Z}_k} \|2H_{z_k}(z - z_c)\|$.

Proof: For details please see [19].

With the aforementioned definitions of barrier functions, we reconstruct $J(x_k)$ with barrier functions defined in (4). Letting $\mu > 0$ be a tuning parameter, the resulting cost function, denoted as $\bar{J}(x_k)$, is defined as $\bar{J}(x_k) = \sum_{k=0}^{+\infty} \gamma^k \bar{r}(x_k, u_k)$, where $\bar{r}(x_k, u_k) = r(x_k, u_k) + \mu \mathcal{B}_k(u_k) + \mu \mathcal{B}_k(x_k)$. Note that, in addition to the logarithmic barrier function (4), other general types of differentiable barrier functions such as exponential, polynomial ones can be naturally used instead to construct $\bar{J}(x_k)$; however, this is beyond the scope of this work.

III. Safe RL With BCP and MPE

This section presents our safe RL approach and its implementation by an efficient actor-critic learning algorithm. Our safe RL approach has two novel designs. The first is a barrier force-based control policy structure, which has physics force interpretations to ensure safety. The second is a multistep policy evaluation mechanism, which provides the multistep safety risk prediction to guide the policy to update safely online under time-varying constraints.

A. Design of Safe RL With BCP and MPE

To solve the control problem with $\bar{J}(x_k)$, we propose a novel barrier force-based control policy inspired by the barrier method in interior-point optimization [20], i.e.,

$$u_k = v_k + \rho \nabla_v \mathcal{B}_k(v_k) + K \nabla_x \mathcal{B}_k(x_k) \tag{6}$$

where $v_k \in \mathbb{R}^m$ is a new virtual control input, $\rho \in \mathbb{R}$ and $K \in \mathbb{R}^{m \times n}$ are decision variables to be further optimized (see also Section IV); $\nabla_v \mathcal{B}_k(v_k)$ is the gradient of $\mathcal{B}_k(v_k)$ for $v_k \in \mathcal{U}_k$, $\nabla_x \mathcal{B}_k(x_k)$ is the gradient of $\mathcal{B}_k(x_k)$ for $x_k \in \mathcal{X}_k$.

Remark 1: In (6), the roles of the second and third terms are to generate the repulsive forces, respectively, as the variables x and v move toward the corresponding boundary of the constraints. As a result, (6) generates joint forces to exactly balance the forces associated with $J(x_k)$ and with the barrier functions in $\bar{J}(x_k)$. Hence, our control policy has physical force interpretations to ensure safety.

Let at any time k the control policy be $\pi_k = \{u_k(x_{k+l})\}_{l=0}^{\infty}$. One can write the difference equation for the multistep prediction of the stage cost under π_k, i.e.,

$$\begin{aligned} \bar{J}(x_k) &= \bar{r}(x_k, u(x_k)) + \gamma \bar{J}(x_{k+1}) \\ &= \sum_{l=0}^{L-1} \gamma^l \bar{r}(x_{k+l}, u(x_{k+l})) + \gamma^L \bar{J}(x_{k+L}). \end{aligned} \tag{7}$$

Under control (6), letting $\bar{J}^*(x_k)$ be the optimal value function at time instant k, a variant of the discrete-time HJB equation can be written as

$$\begin{aligned} \bar{J}^*(x_k) &= \min_{u_k \in \mathcal{U}_k} \bar{r}(x_k, u_k) + \gamma \bar{J}^*(x_{k+1}) \\ &= \min_{u_{k+l} \in \mathcal{U}_{k+l}, l \in \mathbb{N}_0^{L-1}} \sum_{l=0}^{L-1} \gamma^l \bar{r}(x_{k+l}, u_{k+l}) + \gamma^L \bar{J}^*(x_{k+L}) \end{aligned}$$

and the local optimal control policy is

$$\begin{aligned} \pi_k^* &= \operatorname*{argmin}_{u_k \in \mathcal{U}_k} \bar{r}(x_k, u_k) + \gamma \bar{J}^*(x_{k+1}) \\ &= \operatorname*{argmin}_{u_{k+l} \in \mathcal{U}_{k+l}, l \in \mathbb{N}_0^{L-1}} \sum_{l=0}^{L-1} \gamma^l \bar{r}(x_{k+l}, u_{k+l}) + \gamma^L \bar{J}^*(x_{k+L}). \end{aligned}$$

We propose a safe RL algorithm in Algorithm 1 to approximate the optimal policy and value function.

Algorithm 1: Safe RL with BCP and MPE.

require: $\bar{\epsilon} > 0$, u_k^0, $i = 0$.
for $k = 1, 2, \cdots$ **do**
while $\bar{J}^i(x_k) - \bar{J}^{i-1}(x_k) \geq \bar{\epsilon}$ **do**
1) Compute x_{k+l} with u_{k+l}^i based on model (1) for $l \in \mathbb{N}_1^L$.
2) Multistep policy evaluation:

$$\bar{J}^{i+1}(x_k) = \sum_{l=0}^{L-1} \gamma^l \bar{r}(x_{k+l}, u_{k+l}^i) + \gamma^L \bar{J}^i(x_{k+L}). \quad (8a)$$

3) Barrier force-based control policy update:

$$(v_k, \rho, K)^{i+1} = \underset{v_k, \rho, K}{\operatorname{argmin}} \bar{r}(x_k, u_k) + \gamma \bar{J}^{i+1}(x_{k+1}),$$
$$u_k^{i+1}(x_k) = v_k^{i+1} + \rho^{i+1} \nabla_v \mathcal{B}_k(v^{i+1}(x_k)) + K^{i+1} \nabla_x \mathcal{B}_k(x_k). \quad (8b)$$

4) $\pi_k^{i+1} \leftarrow \{u_k^{i+1}(x_{k+l})\}_{l=0}^{\infty}$.
5) $i \leftarrow i + 1$.
end while
end for

B. Barrier-Based Actor-Critic Learning Algorithm

In the following, Algorithm 1 is implemented with a barrier-based actor-critic (BAC) structure. We first construct a consistent type of critic network to $\bar{J}$ with barrier functions

$$\hat{\bar{J}}(x_k) = W_{c1}^\top \sigma_c(x_k) + W_{c2} \mathcal{B}_k(x_k) \quad (9)$$

where $W_{c1} \in \mathbb{R}^{N_c}$ and $W_{c2} \in \mathbb{R}$ are weighting matrices, $\sigma_c \in \mathbb{R}^{N_c}$ is a vector composed of basis functions. In a collective form, we write $\hat{\bar{J}}(x_k) = W_c^\top h_c(x_k)$, where $W_c = (W_{c1}, W_{c2})$, $h_c(x_k) = (\sigma_c(x_k), \mathcal{B}_k(x_k))$.

The ultimate goal of the critic network is to minimize the distance between $\bar{J}^*$ and $\hat{\bar{J}}$ via updating W_c. However, as $\bar{J}^*$ is not available, the following $\bar{J}^d(x_k)$ [defined according to (8a)] is used as the target to be steered by $\hat{\bar{J}}$, i.e., $\bar{J}^d(x_k) = \sum_{l=0}^{L-1} \gamma^l \bar{r}(x_{k+l}, u_{k+l}) + \gamma^L \hat{\bar{J}}(x_{k+L})$. Let $\varepsilon_{c,k} = \bar{J}^d(x_k) - \hat{\bar{J}}(x_k)$ be the approximation residual, $\delta_{c,k} = \varepsilon_{c,k}^2$, and γ_c be the learning rate, then the update rule of weight W_c according to the gradient descent is given as

$$W_{c,k+1} = W_{c,k} - \gamma_c \frac{\partial \delta_{c,k}}{\partial W_{c,k}}. \quad (10)$$

We next design the actor network for learning the control policy (6) with the following form

$$u_k = W_{a,\sigma}^\top \sigma_a(x_k) + \hat{K} \nabla_x \mathcal{B}_k(x_k)) + \hat{\rho} \nabla_v \mathcal{B}_k(v_k) \quad (11)$$

where $W_{a,\sigma} \in \mathbb{R}^{N_u \times m}$, $\hat{K} \in \mathbb{R}^{m \times n}$, and $\hat{\rho} \in \mathbb{R}$ are the weighting matrices, $\sigma_a \in \mathbb{R}^{N_u}$ is a vector of basis functions. Let $W_a^\top = [W_{a1}^\top \hat{\rho} I]$, $W_{a1}^\top = [W_{a,\sigma}^\top \hat{K}]$ and $h_a(x_k) = (h_{a1}(x_k), \nabla_v \mathcal{B}_k(v_k))$, $h_{a1}(x_k) = (\sigma_a(x_k), \nabla_x \mathcal{B}_k(x_k))$, then one can write (11) in a collective form as $u(x_k) = W_a^\top h_a(x_k)$.

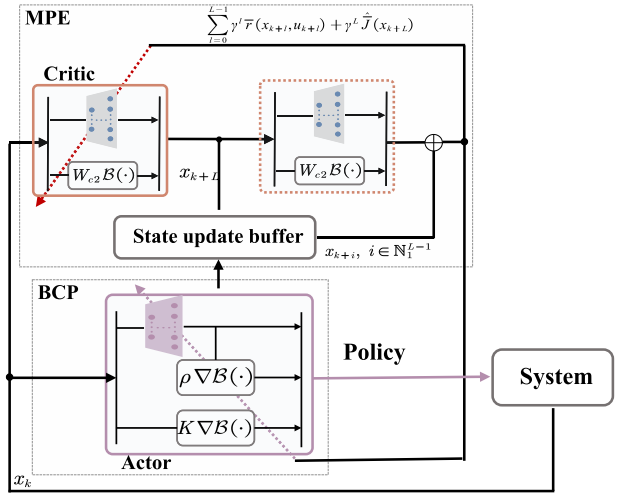


Fig. 1. Schematic diagram of the barrier-based actor-critic learning algorithm.

In view of (8b) and (11), letting $\nu_k = 2Ru_k + \mu \nabla_u \mathcal{B}_k(u_k)$, we define a desired target of ν_k, i.e., ν_k^d as $\nu_k^d = -\nabla_u f(x_k, u_k)^\top \partial \hat{\bar{J}}(x_{k+1}) / \partial x_{k+1}$. Denote $\varepsilon_{a,k} = \nu_k^d - \nu_k$ as the approximation residual, $\delta_{a,k} = \|\varepsilon_{a,k}\|^2$, and γ_a be the learning rate, then the update rule of W_{a1} and $\hat{\rho}$ according to the gradient descent is given as

$$W_{a1,k+1} = W_{a1,k} - \gamma_a \frac{\partial \delta_{a,k}}{\partial W_{a1,k}} \quad (12a)$$

$$\hat{\rho}_{k+1} = \hat{\rho}_k - \gamma_a \frac{\partial \delta_{a,k}}{\partial \hat{\rho}_k}. \quad (12b)$$

For a visual display of the barrier-based actor-critic learning algorithm, please see Fig. 1.

Remark 2: The proposed actor and critic comprise a neural network and safety-related terms multiplied by weighting parameters [see (9) and (11)]. In the critic (9), the safety-related term is to capture variations of the barrier functions in $\bar{J}(x_k)$. In the actor (11), the safety-related terms have clear physical interpretations for safety certificates. In principle, an extra neural network might be used for representing the safety-related terms in (9) and (11). However, the physical interpretations of the policy structure no longer exist. The theoretical results (deferred in the next section) might not hold directly due to possible approximation errors of neural networks. The extension to this case with theoretical guarantees is an open problem and will be left for further investigation.

IV. Theoretical Results

This section presents the theoretical results of the proposed safe RL in nominal and disturbance scenarios as well as the convergence analysis of the BAC learning algorithm.

A. Safety and Stability Guarantees in Nominal Scenario

Assumption 4 (Stabilizability): For any $x_k \in \mathcal{X}_k$, there exists a control policy $\pi := \{u_k\}_{k=0}^{\infty}$ with $u_k \in \mathcal{U}_k$ defined in (6) such that system (1) is locally stabilizable.

From Assumption 4, one promptly obtains the following L-step safe control condition: given $x_k \in \mathcal{X}_k$, there exists an L-step safe control policy such that $x_{k+l} \in \mathcal{X}_{k+l}$ $\forall l \in \mathbb{N}_1^L$. Note that this is a standard condition and can be derived from a 1-step safe control condition using mathematical induction. We highlight that the variation of the state constraints between adjacent time instants can not be arbitrarily large. Let $\tilde{\mathcal{X}}_{k+1} = \{\tilde{x}_{k+1} | \tilde{x}_{k+1} = f(x_k, u_k), \forall x_k \in \mathcal{X}_k, u_k \in \mathcal{U}_k\}$ be the maximal reachable set from $\mathcal{X}_k$ under $\mathcal{U}_k$. We require that the real state constraint at any time $k+1$ satisfies $\mathcal{X}_{k+1} \subseteq \tilde{\mathcal{X}}_{k+1}$.

Theorem 1 (Convergence): If $u_k^0(x_{k+l}) \in \mathcal{U}_{k+l}$ is such that the relaxed barrier function $\mathcal{B}_{k+l+1}(x_{k+l+1})$ $\forall l \in \mathbb{N}_0^{L-1}$, is finite, and the value function $\bar{J}^0(x_k) \geq \bar{r}(x_k, u_k^0) + \gamma \bar{J}^0(x_{k+1})$; then with (8), it holds that

1) $\bar{J}^{i+1}(x_k) \leq V^i(x_k) \leq \bar{J}^i(x_k)$, where $V^i(x_k) = \bar{r}(x_k, u_k^i) + \gamma \bar{J}^i(x_{k+1})$;
2) $\bar{J}^i(x_k) \to \bar{J}^*(x_k)$ and $\pi_k^i \to \pi_k^*$, as $i \to +\infty$.

Proof: Please refer to the extended verison [17]. □

Let π^* be the local optimal control policy via minimizing $\bar{J}(x_k)$ with (1), i.e., $\pi^* = \pi_k^*$ if the constraints are time-invariant and $\pi^* = \pi_0^*(0), \pi_1^*(0), \cdots$, otherwise; where $\pi_k^*(0) = u_k^*(x_k)$. The following proposition can be stated.

Proposition 1 (Stability): Let $\gamma = 1$, $x_0 \in \mathcal{X}_0$. Under Assumptions 3 and 4, the state x_k of model (1) using π^*, converges to the origin as $k \to +\infty$.

Proof: Please refer to the extended verison [17]. □

B. Safety and Robustness Guarantees in Disturbed Scenario

We show that our approach can guarantee safety and robustness under disturbances by properly shrinking the state constraints in the learning process. To this end, let the real model dynamics be given as

$$z_{k+1} = f(z_k, u_k) + w_k \tag{13}$$

where z_k is the real state, $w_k \in \mathcal{W}$ is an additive bounded and unknown disturbance that can represent the modeled uncertainty or measurement noise, $\mathcal{W}$ is a compact set containing origin in the interior. Note that models obtained by first principles or data-driven modeling using neural networks can be utilized in the proposed approach. For a specific data-driven modeling approach and the estimation of the associated uncertainty set $\mathcal{W}$, please refer to [18].

Let at any time instant k, $x_{k+j|k}$ be the predicted state by applying the control $u_k, \ldots, u_{k+L-1}$ using model (1). Assuming that the uncertainty set $\mathcal{W}$ is norm-bounded, i.e., $\|w_k\| \leq \varepsilon_w$, then the following lemma is stated.

Lemma 2 ([21]): The difference between the real state under $u(z)$ and the nominal one under $u(x)$ satisfies

$$\|x_{k+j|k} - z_{k+j}\| \leq \frac{L_f^j - 1}{L_f - 1} \varepsilon_w \tag{14}$$

where $x_{k|k} = z_k$.

Proof: The proof is similar to [21]. □

Let the constraint on the nominal state be shrunk, i.e., $x_{k+j|k} \in \bar{\mathcal{X}}_{k+j}$ where $\bar{\mathcal{X}}_{k+j} = \mathcal{X}_{k+j} \ominus \mathcal{D}_{\varepsilon_w}^j$, $\mathcal{D}_{\varepsilon_w}^j = \{y \in \mathbb{R}^n | \|y\| \leq \frac{L_f^j - 1}{L_f - 1} \varepsilon_w\}$. The barrier function on the state in $\bar{J}(x_k)$ is modified according to the constraint $x_{k+j|k} \in \bar{\mathcal{X}}_{k+j}$. Assume that the computed $\bar{\mathcal{X}}_{k+j}$ is nonempty and contains the origin in the interior for all $k \geq \bar{k}$.

Theorem 2 (Robustness): Under Assumptions 3-4, the state evolution of (13), by applying the learned optimal policy π^* with (1), converges to the set $\mathcal{D}_{\varepsilon_w}^{\infty}$, i.e., $\lim_{k \to +\infty} x_k \to \mathcal{D}_{\varepsilon_w}^{\infty}$.

Proof: Please refer to the extended version [17]. □

As noted in [21], to reduce the size of $\mathcal{D}_{\varepsilon_w}^j$, i.e., the Lipschitz constant L_f, two design choices are suggested: i) a different suitable norm type can be used; ii) an additional feedback term $K(z_k - x_k)$ can be added in the control input to reduce the conservativeness of the multistep prediction of (1), where $K \in \mathbb{R}^{m \times n}$ is a stabilizing gain matrix of (1).

C. Convergence Analysis of BAC Learning Algorithm

Note that, as shown in the Proposition 1 of our extended version [17], the control problem for (1) with $\bar{J}(x_k)$ is equivalent to an unconstrained problem for a time-varying model $x_{k+1} = f(x_k, u_k)$, $y_k = (x_k, \sqrt{\mathcal{B}_k(x_k)})$ with $\bar{J}_u(x_k) = \sum_{k=0}^{+\infty} \gamma^k (\|y_k\|_{Q_y}^2 + \|u_k\|_R^2 + \mu \mathcal{B}_k(u_k))$, where $Q_y = \text{diag}\{Q, \mu\}$. The convergence analysis for the BAC learning algorithm in this scenario would be much involved by Lyapunov method since the optimal weights of the actor and critic are time-dependent due to $y_k = (x_k, \sqrt{\mathcal{B}_k(x_k)})$. For the sake of simplicity, we recall that a time-varying constraint can be partitioned into several segments of time-invariant ones. Hence, in the following, we prove the convergence of the BAC learning algorithm under time-invariant state and control constraints, i.e., $\mathcal{X} = \mathcal{X}_k$ and $\mathcal{U} = \mathcal{U}_k$. That is, we prove that whenever the constraints are changed, our algorithm can eventually converge after some time steps. To this end, one first write

$$\bar{J}^*(x) = W_c^{*\top} h_c(x) + \kappa_c(x), \; u^*(x) = W_a^{*\top} h_a(x) + \kappa_a(x)$$

where W_c^* and W_a^* are constant weights, κ_c and κ_a are reconstruction errors. We introduce the following assumption.

Assumption 5 (Weights and reconstruction errors of BAC):

1) $\|W_c^*\| \leq W_{c,m}$, $\|\sigma_c(x)\| \leq \sigma_{c,m}$, $\|\nabla_x \sigma_c(x)\| \leq \bar{\sigma}_{c,m}$, $\|\kappa_c(x)\| \leq \kappa_{c,m}$;
2) $\|W_a^*\| \leq W_{a,m}$, $\|\sigma_a(x)\| \leq \sigma_{a,m}$, $\|\kappa_a(x)\| \leq \kappa_{a,m}$. ◀

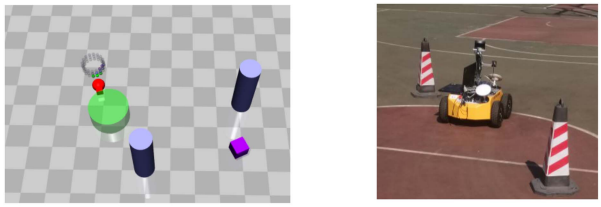
To state the following theorem in a compact form, we let $\tilde{W}_\star = W_\star^* - W_\star$, $\star = a, c$ in turns, denote $\Delta \bar{h}_{c,k} = \Delta h_{c,k}^\top \Delta h_{c,k}$, where $\Delta h_{c,k} = \gamma^L h_{c,k+L} - h_{c,k}$, and use q and q^+ to denote q_k and q_{k+1} respectively unless otherwise specified. For simplicity, we assume that $G^i(u) = E^i u$, $E^i \in \mathbb{R}^{1 \times m}$.

Theorem 3 (Convergence of BAC learning): Under Assumptions 2 and 5, if

$$R - \mu H_u \succ 0 \text{ and } I - 3 d_m (R + \mu H_u)^2 \succ 0 \tag{15a}$$

where $d_m = 4\gamma_a(\sigma_{a,m}^2 + \mathcal{B}_{v,m}^2 + \mathcal{B}_{x,m}^2)$, and

$$q_1 \leq \Delta \bar{h}_{c,k} \leq q_2 \tag{15b}$$



(a) (b)

Fig. 2. (a) Simulation scenario in Safety Gym: The objective is to move the vehicle (red) to the green region while avoiding two static obstacles (grey), the moving soft object (purple) is not considered in the controller design. (b) Experimental platform of the differential-drive vehicle and testing scenario.

where $q_1, q_2 > 0$, then it holds that $\|(\xi_{a,k}, \tilde{W}_{c,k})\| \leq \sqrt{\frac{\epsilon_m}{\lambda_{\min}(S)}}$, as $k \to +\infty$, where $\xi_{a,k} = \tilde{W}_{a,k}^\top h_a(x_k)$, ϵ_m is a bounded error and S is a positive-definite matrix, whose definitions are given in the extended version [17]. Also, $(\xi_{a,k}, \tilde{W}_{c,k}) \to 0$, as $k \to +\infty$, if $\kappa_\star(x_k) \to 0$, $\star = a, c$ in turns.

Proof: Please refer to the extended verison [17]. □

V. Simulation and Experimental Results

In this section, we focus on the applications of our approach to two real-world intelligent vehicles.

A. Application to a Differential-Drive Vehicle: Offline Learning Scenario

Consider a kinematics model

$$\dot{q} = (\dot{p}_x, \dot{p}_y, \dot{\theta}) = (v_o \cos\theta, v_o \sin\theta, \omega), \tag{16}$$

where (p_x, p_y) is the coordinate of vehicle in Cartesian frame, θ is the yaw angle, $u = [v_o, \omega]^\top$ is the input, where v_o and ω are the linear velocity and yaw rate, respectively.

Let us define the path following error as $e = q_r - q$, where q_r is the reference state. One can write the error model and discretize it with a sampling interval $\Delta t = 0.05$ s to derive the model like (1). The constraint for collision avoidance was formulated as $\mathcal{X}_k = \{(p_x, p_y) | \|(p_x, p_y) - c_k\| \geq d\}$, where d and c_k are the radius and center of the obstacle respectively. Also, the size of $\mathcal{X}_k$ was properly shrunk by increasing d to account for uncertainties. In the training, the penalty matrices were selected as $Q = I$, $R = 0.1$, $\mu = 0.001$. The discounting factor γ was $\gamma = 0.95$. The relaxing factor κ_b was $\kappa_b = 0.05$. The basis functions $\sigma_c(x)$ and $\sigma_a(x)$ were chosen as hyperbolic tangent activation functions with $N_c = N_u = 4$. The step L was chosen as $L = 10$. Weights W_c and W_a were initialized with uniformly random numbers.

Simulation results using Safety Gym environment [22]: We tested our approach in the Safety Gym environment with the MoJoCo simulator [23] (see the left panel in Fig. 2). Our method was compared with several state-of-the-art safe RL algorithms: constrained policy optimization (CPO) [24], trust region policy optimization with Lagrangian methods (TRPO-L) [22], proximal policy optimization with Lagrangian methods (PPO-L) [22], deep deterministic policy gradient [25] with cost shaping (DDPG-CS), and soft actor-critic (SAC) [26] with cost shaping (SAC-CS). In the training stage, all the parameter settings of CPO, TRPO-L, and PPO-L were consistent with that in [22]. In DDPG-CS and SAC-CS, we used the same cost function as ours. We directly deployed the offline learned control policy in implementation since we did not know the vehicle's dynamic model. All the comparative algorithms were trained and deployed using the same environment in Safety Gym. The simulation results in Table I show that our approach outperforms all the comparative algorithms in data efficiency, collision avoidance, and performance (see the video details[1] with extracting code: 9426.).

As shown in Table II, when the obstacles overlapped with the reference path between the target and vehicle, our approach offers a significant performance improvement compared with other adopted approaches[1]. In summary, our approach outperforms the comparative model-free safe RL approaches for the following two reasons. Firstly, the proposed barrier force-inspired control policy structure has a clear physical interpretation to guarantee safety online and improve the generalization ability. Secondly, our approach is model-based, facilitating multistep policy evaluation online.

Real-world experimental results with comparisons to nonlinear MPC algorithms: We also tested our proposed algorithm on a real-world differential-drive vehicle platform. The control task is to follow a predefined reference path (with $v_{o,r} = 0.7$ m/s) while passing and avoiding collision with a moving object (vehicle) that is traveling along the reference path. In such a situation, the conflict between the goals of path following and collision avoidance leads to a challenging multiobjective control problem.

In the experiment, the vehicle was equipped with a Laptop running Ubuntu in an Intel i7-8550 U CPU@1.80 GHz. The sampling interval was set as $\Delta t = 0.1$s. We directly deployed the offline learned policy of our approach to control the vehicle. At each sampling instant, the onboard laptop computed the control input in real-time using the state information, which was periodically measured by the onboard satellite inertial guidance integrated positioning system (SIGIPS). To simplify the experimental setup, another wheeled vehicle following the reference path with a lower speed profile ($v_{o,r} = 0.3$ m/s) was regarded as the obstacle to be avoided. Its position and velocity information was measured in real-time by SIGIPS and transmitted to the ego vehicle via a WIFI network.

The following MPC algorithms were adopted for comparison.

1) A nonlinear MPC algorithm with nonconvex circular constraints (NMPC-C). The vehicle obstacle was approximated by a circle, i.e., we enforce constraint $(\Delta p_x)^2 + (\Delta p_y)^2 > d_o^2$ in NMPC-C, where Δp_x and Δp_y were deviations from the robot to the obstacles in the associated coordinate axes, $d_o = 1$m.
2) A nonlinear MPC algorithm with nonconvex ellipsoidal constraints according to [27]. The vehicle obstacle was approximated by an ellipsoid, where the semimajor axis of the ellipsoid was in the direction of the reference

[1][Online]. Available: https://pan.baidu.com/s/1NxJ-zgD4ZdVvqIXQgJkCqg.

TABLE I
NUMERICAL COMPARISONS IN SAFETY GYM WITH RANDOMLY GENERATED OBSTACLE POSITIONS

Approach	Collision rate	Target reach	Average speed (m/s)	Episode	Samples	Training scenario	Deployment scenario
CPO	0.1	0.9	0.82	1000	3.3e5	Safety Gym	Safety Gym
TRPO-L	0.095	0.905	0.85	1000	3.3e5	Safety Gym	Safety Gym
PPO-L	0.095	0.905	0.84	1000	3.3e5	Safety Gym	Safety Gym
DDPG-CS	–	–	–	1000	3.3e5	with data from (1)	–
SAC-CS	–	–	–	1000	3.3e5	with data from (1)	–
Ours without MPE (Speed scenario I)	0.025	0.975	0.76	**100**	**5e3**	with model (1)	Safety Gym
Ours without MPE (Speed scenario II)	0.03	0.97	0.8	**100**	**5e3**	with model (1)	Safety Gym
Ours with MPE	**0**	**0.945**	0.76	**100**	**5e3**	with model (1)	Safety Gym

The bold values are the ones which have the best performance.

TABLE II
NUMERICAL COMPARISONS IN SAFETY GYM WITH GENERATED OBSTACLES ON THE PATH BETWEEN THE TARGET AND VEHICLE

Approach	Collision rate	Target reach	Average speed (m/s)	Episode	Samples
CPO	0.815	0.185	0.76	1000	3.3e5
TRPO-L	0.835	0.165	0.8	1000	3.3e5
PPO-L	0.8	0.2	0.78	1000	3.3e5
Ours without MPE	0.23	**0.77**	0.76	**100**	**5e3**
Ours	**0**	0.595	0.76	**100**	**5e3**

The bold values are the ones which have the best performance.

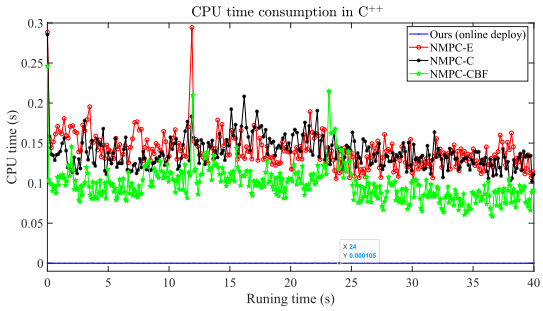


Fig. 3. CPU running time comparison in C++. In many time instants, the computational time values of the NMPC-c, NMPC-e, NMPC-cbf are greater than the adopted sampling interval, i.e., 0.1 s, which could hamper the control performance (see Table III), while the computational time of our approach is much smaller (less than 1 ms) and its influence on the control performance can be negligible.

TABLE III
EXPERIMENTAL COMPARISONS UNDER DYNAMIC OBSTACLES WITH $d_o = 1$

Methods	Scenarios Parameters	d_r(m)	Coll. avoid./overtak.	J_e (coll. avoid.)	J_e (path foll.)
Ours	–	0.07	S/S	**0.237**	**0.003**
		0.56	S/S	**0.286**	0.031
		1.12	S/S	**0.365**	**0.107**
NMPC-C	$\mu_p = 5 \cdot 10^{-4}$	0.07	S/F	–	–
		0.56	S/S	0.579	0.03
		1.12	S/S	0.96	0.199
	$\mu_p = 5 \cdot 10^{-3}$	0.07	S/F	–	–
		0.56	S/S	0.452	0.03
		1.12	S/S	0.639	0.139
NMPC-E	$\mu_p = 5 \cdot 10^{-4}$	0.07	S/F	–	–
		0.56	S/S	0.62	**0.029**
		1.12	S/S	0.782	0.158
	$\mu_p = 5 \cdot 10^{-3}$	0.07	S/F	–	–
		0.56	S/S	0.885	0.033
		1.12	S/S	0.798	0.164
NMPC-CBF	$\eta = 0.5$	0.07	S/F	–	–
		0.56	S/S	0.618	0.03
		1.12	S/S	1.12	0.118
	$\eta = 1.0$	0.07	S/F	–	–
		0.56	S/S	0.589	0.03
		1.12	S/S	0.961	0.139
	$\eta = 2.5$	0.56	S/S	0.428	0.06
		1.12	S/S	0.626	0.3

Cost $J_e = 1/M \sum_{j=1}^{M} \|e_j\|^2$.
d_r represents the distance between adjacent reference points.
"S" and "F" stand for "succeed" and "fail," respectively.
The bold values are the ones which have the best performance.

path. The semimajor radius and semiminor radius were computed as 1.517 and 1.017 m, respectively.

3) A nonlinear MPC algorithm with control barrier function [28] (NMPC-CBF). The collision avoidance constraint is formulated by a control barrier function constraint, i.e., $h(k+1) - h(k) \geq -\eta h(k)$, where $h = (\Delta p_x)^2 + (\Delta p_y)^2 - d_o^2$ is a control barrier function, $\eta > 0$ is properly tuned for fair comparisons.

The stage costs of all the comparative MPC algorithms were designed the same, and the prediction horizon was set as $N_p = 20$. According to [27], the following potential function $J_{\mathrm{p}}(k) = \sum_{j=0}^{N_p-1} \mu_p \frac{1}{(\Delta p_x(k+j))^2 + (\Delta p_y(k+j))^2 + \epsilon_p}$, was additionally adopted to improve the collision avoidance performance in NMPC-C and NMPC-E, ϵ_p was chosen as 0.0001, and μ_p was tuned for fair comparisons.

All the MPC algorithms were solved at each sampling interval based on the CasADi toolbox [29] with an `Ipopt` solver [30]. All the algorithms were tested under different reference profiles. A brief summary of experimental results under dynamic collision avoidance was illustrated in Table III (see the video details). Please see Table V and Figs. 4–9 in [17] for more experimental results with various parameter tuning conditions. The results show that the NMPC-C and NMPC-E failed in realizing overtaking and followed behind the moving obstacle when the adopted reference points were dense, while our approach can realize conflict resolution in all scenarios. Also, our approach outperforms NMPC-C and NMPC-E in terms of the planning performance and path following performance (see Table III). In addition to the unique policy design and learning mechanism of our approach, the performance improvement to the MPC algorithms is also due to the significant computational load reduction (see Fig. 3). To further show the effectiveness of our

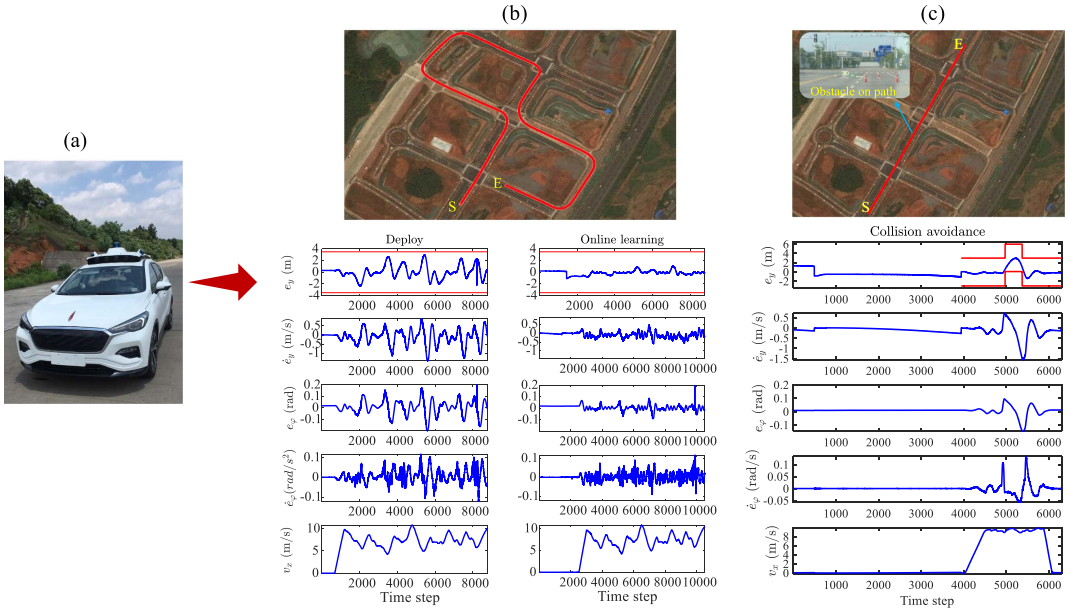


Fig. 4. (a) HongQi EHS3 autonomous driving experimental platform. (b) Upper panel presents the road map with road boundary constraints, where "S" stands for the starting point, "E" stands for the ending point, and the red line is the reference path for the path following control. The lower panel presents the corresponding state errors compared with the offline learning case, a significantly improved performance can be achieved by online policy learning. (c) Upper panel presents the road map with a collision avoidance scenario, while the lower panel gives the numerical state errors.

approach, we carried out extra tests by manually manipulating the moving obstacle to block the path of the ego vehicle when the latter reacted promptly to avoid collision successfully (see Fig. 9 in [17]).

B. Application of an Ackermann-Drive Vehicle: Online Learning Scenario

Consider the path following control of an Ackermann-drive vehicle with collision avoidance. Its simplified lateral dynamics is described by a "bicycle" model (cf. [31]), i.e,

$$\begin{aligned}
\dot{X} &= v_x \cos\varphi - v_y \sin\varphi \\
\dot{Y} &= v_x \sin\varphi + v_y \cos\varphi \\
\dot{v}_y &= -v_x\dot{\varphi} + \frac{2}{m}\left[C_f\left(\delta - \frac{v_y + l_f\dot{\varphi}}{v_x}\right) + C_r\frac{l_r\dot{\varphi} - v_y}{v_x}\right] \\
\ddot{\varphi} &= \frac{2}{I_z}\left[l_f C_f\left(\delta - \frac{v_y + l_f\dot{\varphi}}{v_x}\right) - l_r C_r\frac{l_r\dot{\varphi} - v_y}{v_x}\right] \qquad (17)
\end{aligned}$$

where X and Y are the coordinates of the vehicle center of mass in the Cartesian frame XoY, v_x and v_y are the longitudinal and lateral velocities respectively, φ is the yaw angle, $I_z = 4175\ \text{kg}\cdot m^2$ is the yaw moment of inertia, $m = 1723$ kg is the mass of the vehicle, $C_f = 66900$ N and $C_r = 62700$ N are the cornering stiffness of the front and rear tires, respectively, $l_f = 1.322$ m, $l_r = 1.468$ m, δ is the front wheel angle variable to be manipulated.

Given the path reference points (X^r, Y^r) and v_x, we aim to minimize the lateral distance from the vehicle center of mass to the nearest reference point while avoiding potential collisions with obstacles. To this end, let the nearest point be (X_p^r, Y_p^r), then one can compute the reference yaw angle φ_p^r. Define $e_y = -(X - X_p^r)\sin(\varphi_p^r) + (Y - Y_p^r)\cos(\varphi_p^r)$, $e_\varphi = \varphi - \varphi_p^r$. Let $x = (e_y, \dot{e}_y, e_\varphi, \dot{e}_\varphi)$, then one can obtain the continuous-time lateral dynamical model: $\dot{x} = F_1(x) + F_2(x)\delta + F_3(x)\varphi_p^r$, where $F_1(0) = 0$, $F_3(0) \neq 0$. Since $(x, \delta) = 0$ might not be an equilibrium point if $\varphi_p^r \neq 0$, we introduced a virtual control variable $u = \delta + \delta_f$, where δ_f was selected such that $F_2(x)\delta_f = F_3(x)\varphi_p^r$. Consequently, the lateral dynamical model was discretized with a sampling interval $\Delta t = 0.02$ s, i.e., $x_{k+1} = x_k + \Delta t F_1(x_k) + \Delta t F_2(x_k)u_k$.

In the path following control task with collision avoidance, the cost function was chosen as $\bar{J} = \sum_{k=0}^{+\infty} \|x_k\|_Q^2 + \|u_k\|_R^2 + \mu\mathcal{B}_k(e_{y,k})$, where $Q = I$, $R = 1$, $\mu = 0.02$. The basis functions $\sigma_c(x)$ and $\sigma_a(x)$ were chosen as polynomial kernel functions with $N_c = 10$ and $N_u = 14$.

Real-world experimental results[1]*:* We also tested our safe RL algorithm on the real-world intelligent vehicle platform built with a HongQi EHS3 electric car to realize the path following control (see Fig. 4). In the experiment, the states of the vehicle were measured by a SIGIPS; then, the measured states were transmitted to an industrial control computer, where the control policy was computed using our approach with a sampling interval of 0.02 s. We first applied our algorithm to follow a reference path with road boundaries [see Fig. 4(b)]. Different from that in the simulation tests, the vehicle speed was controlled by a PI controller to track a time-varying speed reference. This

caused a strong nonlinearity of the lateral dynamics, leading to extra difficulties in the control task. The experimental results displayed in Fig. 4 show that the control policy of our approach can be learned offline and deployed online safely, showing an impressive sim-to-real transfer capability. Also, one can achieve better control performance by online learning the control policy, which further demonstrates the adaptability of our approach to dynamic environments.

To show the capability of dealing with time-varying state constraints, we tested our approach to tracking a reference path that overlapped with obstacles [see Fig. 4(c)]. Similarly, the location information of obstacles was assumed to be pre-detected. In the experiment, the control policy was learned and deployed synchronously online. The initial constraints were the road boundaries. Then, the constraint on e_y was changed accordingly once the vehicle was near the obstacle. The vehicle using our approach can avoid collision successfully and converge rapidly to the reference path after completing the collision avoidance task (see again Fig. 4).

C. Implementation Issues and Discussions

Implementation issues: First, the tuning parameter μ is suggested to be chosen smaller than the entries of Q and R to obtain a satisfactory control performance. A larger choice of μ might result in a safe but conservative control policy. Second, the initial values of $W_{a,\sigma}$, $\hat{K}$, and $\hat{\rho}$ in the actor must be properly selected such that the initial control policy with (11) is L-step safe, which is a prior condition in Theorem 1. Finally, the relaxing factor κ_b in Lemma 1 must also be selected properly. A smaller choice is suggested if a less conservative control policy is expected, while a larger choice can be made to ensure absolute control safety.

Discussions: As a prominent feature, our approach can learn an explicit control policy offline and deploy it to a different control scenario even if the concerned constraints are nonlinear and nonconvex. However, in MPC, the control action must be computed online by periodically solving an optimization problem [18], which can be difficult for the on-the-fly implementation under nonlinear and nonconvex constraints, see Section V-A. As shown in the simulation, our learned policy using an inaccurate model shows an impressive sim-to-real transfer capability compared with state-of-the-art model-free RL approaches. In the experiments of differential-drive vehicles, our approach outperforms comparative MPC algorithms under measurement noises and modeling uncertainties. Indeed, our approach is a step forward in applying safe RL to the real-world intelligent vehicle control problem.

VI. Conclusion

This article proposed a safe RL algorithm with a barrier force-based control policy structure and a multistep policy evaluation mechanism for optimal control of discrete-time nonlinear systems with time-varying safety constraints. Under certain conditions, safety can be guaranteed by our approach in both online and offline learning cases. Our approach can solve continuous control tasks in the dynamic environment both online and offline. The convergence and robustness of our safe RL under nominal and disturbed scenarios were proven, respectively. The convergence condition of the barrier force-based actor-critic learning algorithm was obtained.

The simulation and real-world experiment results illustrate that our method outperforms state-of-the-art safe RL approaches in control safety, and shows an impressive sim-to-real transfer capability and a satisfactory real-world online learning performance. In general, the proposed safe RL is a step forward in applying safe RL to the optimal control of real-world nonlinear physical systems with time-varying safety constraints. Future works will consider the extension to model-free and multiagent safe RL with theoretical guarantees.

References

[1] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babusks, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[2] Y. Li, T. Yang, and S. Tong, "Adaptive neural networks finite-time optimal control for a class of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4451–4460, Nov. 2020.

[3] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.

[4] Y. Chow, O. Nachum, A. Faust, E. Dueñez-Guzman, and M. Ghavamzadeh, "Safe policy learning for continuous control," in *Proc. Conf. Robot Learn.*, Nov. 16–18, 2021, vol. 155, pp. 801–821. [Online]. Available: https://proceedings.mlr.press/v155/chow21a.html

[5] T. Kessler, K. Esterle, and A. Knoll, "Mixed-integer motion planning on German roads within the Apollo driving stack," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 851–867, Jan. 2023, doi: 10.1109/TIV.2022.3162671.

[6] T. Xu, Y. Liang, and G. Lan, "CRPO: A new approach for safe reinforcement learning with convergence guarantee," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11480–11491.

[7] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Accelerating safe reinforcement learning with constraint-mismatched baseline policies," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11795–11807.

[8] L. Zheng, Y. Shi, L. J. Ratliff, and B. Zhang, "Safe reinforcement learning of control-affine systems with vertex networks," in *Proc. Conf. Learn. Dyn. Control*, 2021, pp. 336–347.

[9] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *Int. J. Robust Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[10] B. Chen et al., "Context-aware safe reinforcement learning for non-stationary environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 10689–10695.

[11] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, no. 1, pp. 411–444, 2022.

[12] M. Turchetta, A. Kolobov, S. Shah, A. Krause, and A. Agarwal, "Safe reinforcement learning via curriculum induction, *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12151–12162, 2020.

[13] B. Thananjeyan et al., "Recovery RL: Safe reinforcement learning with learned recovery zones," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4915–4922, Jul. 2021.

[14] N. C. Wagener, B. Boots, and C.-A. Cheng, "Safe reinforcement learning using advantage-based intervention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10630–10640.

[15] Y. Hu et al., "Learning to utilize shaping rewards: A new approach of reward shaping," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 15931–15941, 2020.

[16] S. Paternain, M. Calvo-Fullana, L. F. O. Chamon, and A. Ribeiro, "Safe policies for reinforcement learning via primal-dual methods," *IEEE Trans. Autom. Control*, vol. 68, no. 3, pp. 1321–1336, 2022.

[17] X. Zhang, Y. Peng, B. Luo, W. Pan, X. Xu, and H. Xie, "Model-based safe reinforcement learning with time-varying state and control constraints: An application to intelligent vehicles," 2023, *arXiv:2112.11217*.

[18] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, "Robust tube-based model predictive control with Koopman operators," *Automatica*, vol. 137, 2022, Art. no. 110114.
[19] A. G. Wills and W. P. Heath, "Barrier function based model predictive control," *Automatica*, vol. 40, no. 8, pp. 1415–1422, 2004.
[20] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ. Press, 2004.
[21] D L. Marruedo, T. Alamo, and E. F. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proc. IEEE 41st Conf. Decis. Control*, 2002, pp. 4619–4624.
[22] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," 2019, *arXiv:1910.01708*.
[23] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.
[24] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 22–31.
[25] T. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Representation Learn. (ICRL)*, 2016.
[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[27] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4459–4466, Oct. 2019.
[28] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. Amer. Control Conf.*, 2021, pp. 3882–3889.
[29] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi–A software framework for nonlinear optimization and optimal control," *Math. Program. Computation*, vol. 11, no. 1, pp. 1–36, 2019.
[30] A. Wächter and T. L Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.
[31] R. Rajamani, *Vehicle Dynamics and Control*. Berlin, Germany: Springer, 2011.

Xinglong Zhang (Member, IEEE) received the Ph.D. degree in system and control from Politecnico di Milano, Milan, Italy, 2018.

He is currently an Associate Professor with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China. His research interests include safe reinforcement learning, model predictive control, and their applications in automotive systems.

Yaoqian Peng received the B.S. degree in control engineering from Central South University, Changsha, China, in 2020. He is currently working toward the master's degree in control engineering with the National University of Defense Technology, Changsha, China.

His research interests include safe reinforcement learning and its applications in robotics.

Biao Luo (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Beihang University, Beijing, China, in 2014.

He is currently a Professor with the School of Automation, Central South University (CSU), Changsha, China. His research interests include distributed parameter systems, intelligent control, reinforcement learning, deep learning, and computational intelligence.

Dr. Luo is an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, *Artificial Intelligence Review*, *Neurocomputing, and Journal of Industrial and Management Optimization*.

Wei Pan (Member, IEEE) received the Ph.D. degree in bioengineering from Imperial College London, London, U.K., in 2016.

He is currently an Assistant Professor with the Department of Cognitive Robotics, Delft University of Technology, Delft, Netherlands. Until 2018, he was a Project Leader with DJI, Shenzhen, China, responsible for machine learning research for DJI drones and AI accelerator. His research interests include machine learning and control theory with applications in robotics.

Dr. Pan is the recipient of Dorothy Hodgkin's Postgraduate Awards, Microsoft Research Ph.D. Scholarship, and Chinese Government Award for Outstanding Students Abroad, Shenzhen Peacock Plan Award. He is on the editorial board of *CoRL, ICRA, IROS, IEEE Robotics and Automation Letters*.

Xin Xu (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, National University of Defense Technology (NUDT), Changsha, China, in 2002.

He is currently a Professor with the College of Intelligence Science and Technology, NUDT. He has been a Visiting Professor with Hong Kong Polytechnic University, Hong Kong, the University of Alberta, Edmonton, AB, Canada, the University of Guelph, Guelph, ON, Canada, and the University of Strathclyde, Glasgow, U.K. His current research interests include intelligent control, reinforcement learning, and autonomous vehicles.

Dr. Xu has served as an Associate Editor or a Guest Editor for *Information Sciences*, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, *International Journal of Robotics and Automation*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, etc.

Haibin Xie received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, 2006.

He is currently an Associate Professor with the College of Intelligence Science and Technology, National University of Defense Technology. His research interests include machine learning and its applications to automotive systems.