International Journal of Robust and Nonlinear Control

WILEY

RESEARCH ARTICLE

Safe Reinforcement Learning for Optimal Tracking of Continuous-Time Nonlinear Systems

Soha Kanso | Mayank Shekhar Jha | Didier Theilliol

CRAN, UMR 7039, CNRS, Université de Lorraine, Vandoeuvre-lès-Nancy Cedex, France

Correspondence: Mayank Shekhar Jha (mayank-shekhar.jha@univ-lorraine.fr)

Received: 23 January 2024 | **Revised:** 4 January 2026 | **Accepted:** 9 March 2026

Keywords: control barrier function | model-based control | reinforcement learning | safety | tracking

ABSTRACT

This work develops a novel off-policy safe reinforcement learning (RL) approach for optimal tracking of continuous-time nonlinear systems, affine in control input. The main contribution consists of the synthesis of an optimal tracker under safety guarantees. A novel formulation is developed enabling optimal tracking of references while satisfying state-based safety constraints. The tracking error and the state dynamics are considered to form an augmented system, facilitating this dual objective with the primary goal being to guarantee the safety without compromising the system performance. To this end, the safety is achieved during the exploration phase, by dynamically adjusting control inputs that are solutions of quadratic programming (QP) problem that incorporates zeroing control barrier function (ZCBF) conditions. Additionally, the safety during exploitation (operational phase) of the learned policy is strengthened by integrating a reciprocal control barrier function (RCBF) into the cost function, leading to an effective trade-off between safety and system performance. Neural networks are employed to approximate the optimal control law, and novel mathematically rigorous proofs are developed to guarantee the safety, the stability, and the convergence towards optimality. Finally, the effectiveness of the approach is assessed using a simulation example.

1 | Introduction

Safety is an essential and crucial factor in modern control applications [1]. This is becoming particularly relevant in fields experiencing rapid growth, such as industrial robotics, autonomous driving, and aerospace. Safety requirements can arise due to state as well as input-based constraints, that restrict the behavior of the system and prevent it from exceeding its boundaries and causing damages [2].

Within this context, there has been an increasing interest in developing optimal controllers that ensure system safety while taking into consideration predefined performance criteria. One common optimal control problem in this regard is the tracking problem, where the goal is to design controllers that guide a system to follow a desired trajectory or reference signal while also satisfying safety constraints. This task involves considerable difficulties, especially since system dynamics are uncertain and subject to change over time [3, 4] which can lead to deviations from the desired trajectory, compromising the safety and performance.

Reinforcement learning (RL) has proven to be a powerful learning technique that allows designing optimal controllers for uncertain systems by iteratively interacting with the environment in real-time [5]. Using RL methods, optimal regulation problems have been successfully solved by learning the solution of the Hamilton-Jacobi-Bellman (HJB) equations. Initially,

Abbreviations: CLF, control Lyapunov function; CT, continuous-time; DT, discrete-time; HJB, Hamilton-Jacobi-Bellman; HJI, Hamilton-Jacobi-Isaac; IRL, integral reinforcement learning; PI, policy iteration; QP, quadratic programming; RCBF, reciprocal control barrier function; RL, reinforcement learning; STH, safe tracking Hamiltonian; ZCBF, zeroing control barrier function.

RL methods were primarily developed for systems operating in discrete-time (DT) for partially [5] and fully unknown [6] dynamical systems. Subsequently, RL algorithms were further adapted and developed to handle systems operating in continuous-time (CT), two notable algorithms in this context are integral reinforcement learning (IRL) [7] for partially known dynamical systems and off-policy [8] algorithm for fully unknown dynamical systems.

The adaptation of RL-based adaptive optimal regulators into adaptive optimal trackers has captured significant interest due to its wider range of practical applications. In [9], a Q-learning approach is introduced to solve the linear quadratic tracking problem in DT for linear systems, without prior knowledge of system dynamics. For nonlinear systems, [10] presents an adaptive optimal control approach for solving tracking problems for partially known dynamical systems with input constraints. In CT, several works have addressed the problem of tracking. For instance, [11] presents a novel approach for solving optimal tracking control for nonlinear systems using the IRL approach, addressing the optimal tracking problem without prior knowledge of system drift dynamics, while incorporating input constraints. Chen et al. [12] applies an off-policy method and uses experience replay to achieve zero tracking error with data efficiency for linear systems with unknown dynamics. Modares et al. [13] addresses the design of an H_∞ tracking controller for nonlinear systems with completely unknown dynamics. However, it is important to note that while these approaches show promise in addressing tracking control problems, they do not explicitly incorporate safety guarantees related to state constraints in their formulations.

Many works have addressed the challenge of ensuring safety in regulatory problems. In [14] and [15], Gaussian processes were employed to reduce the conservativity of reachability analysis by capturing state-dependent disturbances. The operational region of the system is expanded by solving modified terminal value Hamilton-Jacobi-Isaac (HJI) equations and integrating safety metrics for controller switching. In [16], an action projection-based safety shield uses parameterized reachability analysis directly on the original nonlinear system model. [17] incorporates policy-gradient RL algorithm and control barrier functions (CBF) to achieve safety while relying on nominal models. Yang et al. [18] introduces a barrier function (BF)-based system transformation technique that guarantees full-state constraints. By transforming the full-state constrained system to an equivalent system without state constraints, this method enables the use of standard control and optimization techniques. Cohen and Belta [19] combines model-based RL with CBF for a safe exploration scheme. Using Lyapunov-like BF [20], it forms a partially model-free safeguarding controller that ensures safety alongside learning-based control policies. This controller enables online value function learning through experience simulation. Recently, [21] uses off-policy RL to learn an optimal safe policy that minimizes a cost augmented by CBF, while data collection uses a safe and potentially conservative policy. Lastly and very recently, [22] proposes an end-to-end safe learning approach based on CBF and control Lyapunov function (CLF) conditions, that assures safety during initialization, exploration, and exploitation for the optimal regulation problem.

Most of the existing works focus on the safety issue associated with the optimal regulation problem, and the optimal tracking problem remains relatively unexplored within the safe paradigm. It is important to highlight that only very few research efforts have been made to address the critical safety aspect of tracking problems. Notable contributions in this area include the works of [23]. Moreover, in all the existing works centered on nonlinear systems, typically an augmented system is formed comprising the error dynamics and the reference trajectory. While such a formulation ensures that all states follow their respective references, it does not address the cases where some states (more than one) might not require tracking and obliges all the states to track reference(s) necessarily.

To bridge this existing scientific gap, this paper proposes a novel approach by considering system states and tracking errors as an augmented system (against error dynamics-reference trajectory based augmented system) to address the problem of safe tracking while ensuring safety during the exploration phase as well as the operational (exploitation) phase. The contributions in this work can be summarized as follows:

- Developing a new tracking formulation with an augmented system composed of system states and tracking error for effective state tracking;
- Ensuring safety and state constraint satisfaction during the exploration phase using CBFs, which involves formulating a Quadratic Programming (QP) problem;
- Assuring the safety and optimality of the learned policy (operational policy) by augmenting the reward function with a CBF.
- Developing novel rigorous mathematical proofs to demonstrate stability, optimality, and safety guarantees under the proposed algorithm.

The paper is organized as follows. In Section 2, the safe tracking problem is formulated. Section 3 presents the proposed approach to solve the safe and optimal tracking problem, where a safe policy iteration (PI) algorithm is proposed. In Section 4, QP problem is used to ensure safety during exploration and data collection, and an off-policy algorithm is developed. Section 5 examines the effectiveness of the proposed approach on an academic application. Finally, the conclusion summarizes the significant advances and presents the future perspectives.

Notations. The interior of set $\mathscr{C}$ is denoted as $Int\mathscr{C}$ and $\partial\mathscr{C}$ stands for its boundary. For a differentiable function $V(x)$ and a vector $f(x)$, the notation $L_f V(x)$ corresponds to $\frac{\partial V}{\partial x} f(x)$. The symbol $\otimes$ denotes the Kronecker product.

2 | Safe Optimal Tracking Control Problem

In this section, the formulation of a nonlinear optimal tracker is presented. This paper is primarily dedicated to the analysis and control of nonlinear systems affine in control input in continuous time:

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

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ represents the state of the system, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input. $f(.)$: $\mathcal{X} \to \mathbb{R}^n$ and $g(.)$: $\mathcal{X} \to \mathbb{R}^{n\times m}$ are Lipschitz continuous and $f(0) = 0$. All the states of the system are supposed to be measurable. The sets $\mathcal{X}$ and $\mathcal{U}$ are compact. $\mathcal{U}$ denotes the set of all admissible inputs that ensure stability of the system. $\mathcal{C} \subseteq \mathcal{X}$ represents the set of safe feasible states, thus the set inside which the system's state must evolve to assure a safe operation. The mathematical definition of $\mathcal{C}$ is as follows:

$$\mathcal{C} = \{x \in \mathcal{X} \quad | \quad h(x) \geq 0\} \tag{2}$$

for a smooth (continuously differentiable) function h: $\mathcal{X} \to \mathbb{R}$.

In this work, we make this important assumption.

Assumption 1. The equilibrium $x = 0$ lies in the interior of the safe set, that is, $0 \in Int\ \mathcal{C}$ (equivalently, $h(0) > 0$).

The objective in this work is to design a safe infinite-horizon tracker for the system (1). The controller must force the state $x(t)$ to optimally follow the reference trajectory $x_r(t)$ while adhering to safety boundaries and constraints.

Assumption 2. The command generator model of the reference trajectory [11] is defined by:

$$\dot{x}_r = z(x_r) \tag{3}$$

where $z(.)$ is a Lipschitz continuous function with $z(0) = 0$ and $x_r \in \mathbb{R}^p$ is bounded, with $p \leq n$.

The error is formulated as:

$$e_r(t) = Cx(t) - x_r(t) \tag{4}$$

where $C \in \mathbb{R}^{p\times n}$ refers to the states to be tracked, since the target in this paper is to track specific states. The dynamics of the tracking error can be expressed in terms of the control input u as follows:

$$\dot{e}_r = C(f(x) + g(x)u) - z(Cx - e_r) \tag{5}$$

Based on (1) and (5), the augmented system is defined in terms of the system states x and the tracking error e_r as:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{e}_r \end{bmatrix} = F(X) + G(X)u \tag{6}$$

with $F(X) = \begin{bmatrix} f(x) \\ Cf(x) - z(Cx - e_r) \end{bmatrix}$ and $G(X) = \begin{bmatrix} g(x) \\ Cg(x) \end{bmatrix}$.

The general reward function for the tracking problem is usually considered in the following manner:

$$r = e_r^T Q e_r + u^T R u \tag{7}$$

where Q and R are symmetric and positive definite. However, this reward function does not take into account any safety considerations. To this end, in the following section, this problem will be addressed.

3 | Safe Reinforcement Learning Algorithm for Solving the Tracking Problem

3.1 | Safe Value Function

Now, based on the augmented system (6), a modified reward function that is sensitive to the system safety is introduced as

$$\tilde{r}(X, u) = X^T \tilde{Q} X + u^T R u + \overline{B}_\vartheta(\rho X) \tag{8}$$

with $\tilde{Q} = \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix}$, $\rho = \begin{bmatrix} I_n & 0_{n\times p} \end{bmatrix} \in \mathbb{R}^{n\times(n+p)}$(i.e., $\rho X = x$), and $\overline{B}_\vartheta$ a normalized reciprocal control barrier function (RCBF) term defined by

$$\overline{B}_\vartheta(\rho X) \triangleq B_\vartheta(\rho X) - B_\vartheta(0), \tag{9}$$

where B_ϑ is defined in (10) as follows

$$B_\vartheta(\rho X) = -\log\left(\frac{\vartheta h(\rho X)}{\vartheta h(\rho X) + 1}\right) \tag{10}$$

with $\vartheta > 0$.

Definition 1. A function B: $Int\mathcal{C} \to \mathbb{R}$ is a RCBF for the set $\mathcal{C}$ if there exists class κ functions α_1, α_2 and α_3 such that

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))} \tag{11}$$

$$\inf_{u\in\mathcal{U}} \left(L_f B(x) + L_g B(x)u - \alpha_3(h(x))\right) \leq 0 \qquad \forall x \in Int\mathcal{C} \tag{12}$$

Based on the safe reward function (8), the following discounted performance function is introduced

$$V(X(t)) = \int_t^\infty e^{-\gamma(\tau - t)} \tilde{r}(X, u) d\tau \tag{13}$$

where $\gamma > 0$ is the discount factor. Using a discounted performance function is crucial in the proposed formulation since, for the tracking problems, the system trajectory does not reach zero. As such, without the discount factor, the performance function becomes infinite because the control input incorporates a feed-forward component, preventing it from converging to zero as $t \to \infty$, as indicated in [10].

Remark 1. Since the equilibrium satisfies $0 \in Int\ \mathcal{C}$ (i.e., $h(0) > 0$) (see Assumption 1), the barrier value $B_\vartheta(0)$ is finite. With the normalized barrier term in (9), we have $\tilde{r}(0, 0) = 0$ and therefore the discounted value function (13) satisfies $V(0) = 0$ exactly for any $\vartheta > 0$.

Moreover, subtracting the constant $B_\vartheta(0)$ only shifts the value function by a constant and does not modify the optimal policy in (21), since the policy depends on ∇V.

Remark 2. The quantity $B_\vartheta(0)$ is a known scalar constant obtained by evaluating the known safety function $h(\cdot)$ at $x = 0$ and substituting into (10), that is,

$$B_\vartheta(0) = -\log\left(\frac{\vartheta\ h(0)}{\vartheta\ h(0) + 1}\right).$$

For multiple constraints $h_j(x) \geq 0$ $(j = 1, \ldots, q)$ implemented via a summed barrier $B_\vartheta(x) = \sum_{j=1}^{q} B_{j,\vartheta}(x)$ leads to $B_\vartheta(0) = \sum_{j=1}^{q} B_{j,\vartheta}(0)$.

The objective then becomes to find a safe and optimal control policy that minimizes (13). Before addressing the problem of safe tracking, the following definition and assumption are made.

Definition 2. A control policy is defined as a function from the state space to the control space, typically denoted as π: $\mathcal{X} \times \mathbb{R}^p \to \mathcal{U}$. In other words, given a state X, the policy determines a control input by

$$u = \pi(X) \tag{14}$$

Definition 3. The set of safe and admissible inputs $\mathcal{U}_s$ for the current state x is defined as

$$\mathcal{U}_s = \{u \in \mathcal{U} \quad | \quad x_u \in \mathcal{C}\} \tag{15}$$

x_u is the state of the system (1) evolved by the input u.

Assumption 3. There exists a safe feedback control policy $\pi_0(.)$: $\mathcal{C} \times \mathbb{R}^p \to \mathcal{U}_s$ that asymptotically stabilizes the tracking error (5) and stabilizes the system (1) and the cost defined in (13) is finite.

The primary objective of this work is to minimize the value function as defined in (13), which serves as a key measure of performance and effectiveness in achieving safe and optimal tracking control. In the following section, the solution that enables tackling this objective effectively will be developed.

3.2 | Safe Bellman and Safe HJB Equations for Tracking Problem

Applying Leibniz's rule, the following safe tracking Bellman equation is defined by

$$\dot{V} = -\tilde{r}(X,u) + \int_t^\infty \frac{\partial}{\partial t} e^{-\gamma(\tau - t)} \tilde{r}(X(\tau), \pi(X(\tau)))\mathrm{d}\tau \tag{16}$$

Since the second term on the right-hand side of (16) is equal to $\gamma V(X)$, it gives

$$\dot{V}(X) = -\tilde{r}(X,u) + \gamma V(X) \tag{17}$$

Next, we construct a Hamiltonian for the safe tracking problem.

Definition 4. For any $V \in C^1$ and any $u \in \mathcal{U}$, a Safe Tracking Hamiltonian (STH) is defined as

$$\mathcal{H}(X, \nabla V, u) \triangleq \nabla V^T(F(X) + G(X)u) + \tilde{r}(X,u) - \gamma V(X). \tag{18}$$

For a fixed admissible policy $u = \pi(X)$, the corresponding value function V^π satisfies the policy evaluation equation $\mathcal{H}(X, \nabla V^\pi, \pi(X)) = 0$. The optimal safe value function satisfies the safe-THJB equation

$$\min_{u \in \mathcal{U}} \mathcal{H}(X, \nabla V, u) = 0. \tag{19}$$

For the quadratic input penalty term in $\tilde{r}(X,u)$, minimizing STH $\mathcal{H}(X, \nabla V, u)$ with respect to u yields the first-order optimality condition $\frac{\partial \mathcal{H}}{\partial u} = 0$, which leads to the optimal policy

$$\pi^*(X) = -\frac{1}{2} R^{-1} G^T(X) \nabla V^*(X). \tag{20}$$

To derive the equation for the optimal policy, the STH in Equation (18) is minimized with respect to u. Taking the derivative of $STH(V,u)$ with respect to u and setting it to zero, and assuming a quadratic reward $\tilde{r}(X,u) = X^T \tilde{Q} X + u^T R u$, yields to the optimal policy π^*

$$\pi^*(X) = -\frac{1}{2} R^{-1} G^T(X) \nabla V^*(X) \tag{21}$$

where $V^*(X)$ is the optimal cost function defined by

$$V^*(X) = \min_{\pi(.)} \int_t^\infty e^{-\gamma(\tau - t)} \tilde{r}(X(\tau), \pi(X(\tau)))\mathrm{d}\tau. \tag{22}$$

By substituting the optimal control (21) in (18), the STH becomes a safe tracking Hamilton-Jacobi-Bellman (safe-THJB) equation

$$H_{safe}(V^*(X)) \triangleq \nabla V^{*T}(X) F(X) + X^T \tilde{Q} X + B_\vartheta(\rho X) - \gamma V^*(X) - \frac{1}{4} \nabla V^{*T}(X) G(X) R^{-1} G^T(X) \nabla V^*(X) = 0 \tag{23}$$

Assuming that there exists an optimal safe control policy, it implies the existence of an optimal safe value function satisfying the safe-THJB equation

$$H_{safe}(V^*(X)) = 0 \tag{24}$$

where $V^*(X)$ is a safe Lyapunov function (22) for the closed-loop augmented system (6).

Assumption 4. There exists $V^* \in \mathcal{P}$, where $\mathcal{P}$ is the set of all functions in C^1 that are also positive definite and radially unbounded, such that the safe-THJB Equation (24) holds.

Lemma 1 establishes the uniqueness of solution to the safe-THJB Equation (24).

Lemma 1. *The Safe-THJB Equation (24) has a unique solution* $V^* \in \mathcal{P}$.

Proof. Considering another solution $\overline{V} \in \mathcal{P}$ to (24), along the solutions of the closed-loop augmented system composed of (6) and denote the policy associated by $\overline{\pi}(X)$. The policy $\overline{\pi}(X)$ is given by:

$$\overline{\pi}(X) = -\frac{1}{2} R^{-1} G^T(X) \nabla \overline{V}(X) \tag{25}$$

Then, along the solutions of the closed-loop system composed of (6) and the control policy $\bar{u} = \overline{\pi}(X)$, the following TSLE holds:

$$\nabla \overline{V}^T(F(X) + G(X)\bar{u}) + \tilde{r}(X,\bar{u}) - \gamma \overline{V}(X) = 0 \tag{26}$$

Similarly, the TSLE for the optimal value function V^*, with the associated optimal policy $u^* = \pi^*(X)$, is:

$$\nabla V^{*T}(F(X) + G(X)u^*) + \tilde{r}(X,u^*) - \gamma V^*(X) = 0 \tag{27}$$

By subtracting (27) from (26), it yields

$$\begin{aligned}
&\left(\nabla\overline{V}(X)-\nabla V^*(X)\right)^T F(X)+\nabla\overline{V}^T(X)G(X)\bar{u}\\
&\quad-\nabla V^{*T}(X)G(X)u^*+\nabla V^{*T}(X)G(X)\bar{u}\\
&\quad-\nabla V^{*T}(X)G(X)\bar{u}+\bar{u}^T R\bar{u}-u^{*T}Ru^*-\gamma\overline{V}(X)+\gamma V^*(X)\\
&=\left(\nabla\overline{V}(X)-\nabla V^*(X)\right)^T(F(X)+G(X)\bar{u})\\
&\quad+\nabla V^{*T}(X)G(X)(\bar{u}-u^*)+\bar{u}^T R\bar{u}-u^{*T}Ru^*\\
&\quad+\gamma\left(V^*(X)-\overline{V}(X)\right)\\
&=\left(\nabla\overline{V}(X)-\nabla V^*(X)\right)^T(F(X)+G(X)\bar{u})\\
&\quad-2u^{*T}R(\bar{u}-u^*)+\bar{u}^T R\bar{u}-u^{*T}Ru^*+\gamma\left(V^*(X)-\overline{V}(X)\right)\\
&=\left(\nabla\overline{V}(X)-\nabla V^*(X)\right)^T(F(X)+G(X)\bar{u})\\
&\quad+(\bar{u}-u^*)^T R(\bar{u}-u^*)+\gamma\left(V^*(X)-\overline{V}(X)\right)=0
\end{aligned}\tag{28}$$

Multiplying both sides by $e^{-\gamma t}$ gives

$$\begin{aligned}
&e^{-\gamma t}\left(\frac{d\overline{V}(X)}{dt}-\gamma\overline{V}(X)-\frac{dV^*(X)}{dt}+\gamma V^*(X)\right)\\
&\quad=-e^{-\gamma t}(\bar{u}-u^*)^T R(\bar{u}-u^*)
\end{aligned}\tag{29}$$

For all X_0, along the trajectories of the augmented system (6) with $u=u^*$,

$$e^{-\gamma t_0}\left(\overline{V}(X_0)-V^*(X_0)\right)=-\int_0^T e^{-\gamma t}(\bar{u}-u^*)^T R(\bar{u}-u^*)dt \tag{30}$$

Thus $\overline{V}(X)\geq V^*(X)$. Moreover, by subtracting (26) from (27), it gives

$$\begin{aligned}
&\left(\nabla V^*(X)-\nabla\overline{V}(X)\right)^T(F(X)+G(X)u^*)\\
&\quad+(\bar{u}-u^*)^T R(\bar{u}-u^*)\\
&\quad+\gamma\left(\overline{V}(X)-V^*(X)\right)=0
\end{aligned}\tag{31}$$

By following the same steps, we obtain $\overline{V}(X)\leq V^*(X)$.

Thus, it can be concluded that $\overline{V}(X)=V^*(X)$. □

The optimal safe tracking problem is solved by finding the solution of the safe-THJB Equation (24) for the value function. Then, the optimal policy is obtained by replacing the solution in (21). However, it is difficult to find the solution to (24) due to its nonlinear nature. In this regard, RL method has emerged as a powerful technique that allows addressing optimal control problems by using iterative methods. In the next section, PI algorithm will be introduced as one of the approaches to address these challenges.

3.3 | Safe Tracking Policy Iteration (PI) Algorithm

Given the analytical challenges to solve the nonlinear safe-THJB Equation (24), the problem can be tackled by using the following safe tracking PI Algorithm 1 bellow:

ALGORITHM 1 | Safe tracking policy iteration algorithm.

Initialization. Initialize $\pi_0(.)$ with a safe and admissible policy such as $\pi_0\in\mathscr{U}_s$.

Policy Evaluation. Update the value using

$$\nabla V_i^T(X)\left(F(X)+G(X)\pi_i(X)\right)+\tilde{r}\left(X,\pi_i(X)\right)-\gamma V_i(X)=0 \tag{32}$$

Policy Improvement. The control policy is improved by

$$\pi_{i+1}(X)=-\frac{1}{2}R^{-1}G^T(X)\nabla V_i(X) \tag{33}$$

The following theorem establishes the convergence property of the proposed safe tracking PI algorithm. To simplify the notation, let $u^*=\pi^*(X)$, $u_i=\pi_i(X)$ and $u_{i+1}=\pi_{i+1}(X)$.

Theorem 1. *Suppose Assumptions 3 and 4 hold, and the solution $V_i(X)\in C^1$ satisfying (32) exists for $i=0,1,\ldots$. Then, the following properties hold $\forall i=0,1,\ldots$.*

1. $V^*(X)\leq V_{i+1}(X)\leq V_i(X),\ \forall X\in\mathscr{C}\times\mathbb{R}^p$.
2. *Let* $\lim_{i\to\infty}V_i(X_0)=V(X_0)$ *and* $\lim_{i\to\infty}\pi_i(X_0)=\pi(X_0)$, $\forall X_0\in\mathscr{C}\times\mathbb{R}^p$. *Then* $X^*=X$ *and* $u^*=u$, *if* $V\in C^1$.
3. *u_i asymptotically stabilizes the error dynamics.*
4. *u_i is a safe policy, $u_i\in\mathscr{U}_s$.*

The proof is provided in Appendix.

Safe tracking PI algorithm is an effective method that allows to learn an optimal policy while ensuring safety. However, it does present some challenges that need to be addressed. First, solving Equation (32) directly at each iteration is challenging due to its complexity, as it involves partial derivatives of the value function and nonlinear terms.

To overcome this, neural networks (or more generally, linearly-parameterized basis-function approximators) are used to approximate the value function and the policy iteratively; see Section 4.2, Equations (42–60) for the approximation structure and the least-squares implementation.

Moreover, one of the main challenges arises during the stage of data collection, where the state space is explored by adding exploration noise to the policy. While probing noise can provide important information about the system, it can also lead to violation of safety constraints. In the following section, this problem is addressed.

4 | Safe Off-Policy for Tracking Problem

This section focuses on the implementation of the proposed safe tracking PI algorithm. It is important to note that generally, the PI algorithm can be implemented using two different approaches: on-policy and off-policy. However, owing to the advantages of the off-policy approach for safe learning [22] this paper only develops off-policy based approach. Off-policy based approaches

are relevant to safe learning problems and remain efficient as they update multiple value functions associated with distinct learning policies by reusing a stream of experiences obtained by applying a certain behavior policy [24]. However, in these methods, the incorporation of probing noise is fundamental for exploration purposes. Nevertheless, introducing such noise during the exploration phase entails inherent risks, as it can trigger exploratory actions that potentially lead to undesired or unsafe system states.

4.1 | Safe Exploration

For systems subject to probing noise ϵ, consider

$$\dot{X} = F(X) + G(X)u_{noisy} \tag{34}$$

with $u_{noisy} = u_0 + \epsilon$.

The probing noise is assumed not to destabilize the system as denoted in the following assumption.

Assumption 5. The closed-loop system (34) is input-to-state stable (ISS) when ϵ is considered as input.

The first family of CBF, referred as RCBF, was introduced in Section 2. Although it can be employed to ensure safety of exploration, zeroing control barrier function (ZCBF) [25] will be adopted in this section.

Definition 5. A function $h: \mathcal{X} \to \mathbb{R}$ is a ZCBF defined on set $\mathcal{X}$ with $\mathscr{C} \subseteq \mathcal{X} \subseteq \mathbb{R}^n$ if there exists an extended class κ functions α such that

$$\sup_{u \in \mathcal{U}_s} \left(L_f h(x) + L_g h(x)u + \alpha(h(x)) \right) \geq 0 \qquad \forall x \in \mathcal{X} \tag{35}$$

In the context of exploration, ZCBF h is favored over RCBF B_ϑ because ZCBF allows taking into consideration the effects of model disturbances since it is defined on a set $\mathcal{X}$ larger than $\mathscr{C}$ [26].

By satisfying the condition of the ZCBF, the safety of the policy can be assured by marginally modifying the unsafe policy. Thus, the exploration policy can be adjusted by adding the solution of the following QP problem.

QP Problem: Find the additional control input u_{safe} that satisfies

$$\begin{aligned} \min_{u_{safe}} \quad & \frac{1}{2} u_{safe}^T u_{safe} \\ s.t. \quad & L_F h(\rho X) + L_G h(\rho X)(u_{noisy} + u_{safe}) + \alpha(h(\rho X)) \geq 0 \end{aligned} \tag{36}$$

The solution of the QP problem u_{safe} is crucial for ensuring policy safety within the off-policy algorithm. It allows data collection not only within the safe set but also near the boundaries, improving the performance of the algorithm. Moreover, it is essential to note that the functions f and g must be explicitly known for the solution of the QP problem which render the approach model-based.

4.2 | Safe Learning

Safe off-policy RL is an iterative algorithm for finding an optimal and safe controller. It operates through two distinct policies: the behavior policy, which is a safe policy used for data collection during exploration, and the target policy, which evolves toward optimality using the accumulated data. Once the learning process converges, the optimal safe policy is implemented in the system. This section presents the details of the proposed safe tracking off-policy algorithm. The following system is considered

$$\dot{X} = F(X) + G(X)u_s \tag{37}$$

where $u_s = u_{noisy} + u_{safe}$. Then, (37) can be expressed as

$$\dot{X} = F(X) + G(X)u_i + G(X)v_i \tag{38}$$

with $v_i = u_s - u_i$.

From (33), one has

$$\nabla V_i^T(X) G(X) = -2u_{i+1}^T R \tag{39}$$

Thus, for all $i \geq 0$, the time derivative of $V_i(X)$ along the solutions of (37) is obtained by

$$\begin{aligned} \dot{V}_i &= \nabla V_i^T(X)(F(X) + G(X)u_i + G(X)v_i) \\ &= -X^T \tilde{Q} X - u_i^T R u_i - B_\vartheta(\rho X) \\ &\quad + \gamma V_i(X) - 2u_{i+1}^T R v_i \end{aligned} \tag{40}$$

Integrating both sides of (40) over any time interval $[t, t+T]$ yields to

$$\begin{aligned} V_i(X(t+T)) - V_i(X(t)) = -\int_t^{t+T} \big(& X^T \tilde{Q} X + u_i^T R u_i \\ & + B_\vartheta(\rho X) - \gamma V_i(X) \\ & + 2u_{i+1}^T R v_i \big) dt \end{aligned} \tag{41}$$

Considering $\Omega \subseteq \mathbb{R}^{n+p}$ as a compact set. Then, according to the Weierstrass higher order approximation Theorem [27], the value function V_i (corresponding to the critic) and the control policy u_{i+1} (corresponding to the actor) can be approximated using a dense basis function [5, 28]:

$$\tilde{V}_i(X) = W_i \tilde{\Phi}(X) \tag{42}$$

$$\tilde{u}_{i+1}(X) = U_i \tilde{\Psi}(X) \tag{43}$$

with $\tilde{\Phi} = [\tilde{\phi}_1, \tilde{\phi}_2 \dots \tilde{\phi}_{N_1}]^T$ and $\tilde{\Psi} = [\tilde{\psi}_1, \tilde{\psi}_2 \dots \tilde{\psi}_{N_2}]^T$, are the vectors of linearly independent smooth basis functions on Ω. N_1 and N_2 refer respectively to the number of critic and actor basis functions. $W_i \in \mathbb{R}^{1 \times N_1}$ and $U_i \in \mathbb{R}^{m \times N_2}$ are the matrices weights to be determined.

Assumption 6. (Approximation capability on a compact set) On the compact set $\Omega \subset \mathbb{R}^{n+p}$, there exist ideal weights $W_i^\star$ and $U_i^\star$ and bounded approximation errors $\varepsilon_{V,i}(X)$ and $\varepsilon_{u,i}(X)$ such that

$$V_i(X) = W_i^\star \tilde{\Phi}(X) + \varepsilon_{V,i}(X), \qquad \pi_{i+1}(X) = U_i^\star \tilde{\Psi}(X) + \varepsilon_{u,i}(X), \tag{44}$$

with $\sup_{X\in\Omega}|\varepsilon_{V,i}(X)| \le \bar{\varepsilon}_V$ and $\sup_{X\in\Omega}\|\varepsilon_{u,i}(X)\| \le \bar{\varepsilon}_u$. Moreover, the approximation errors converge uniformly to zero, that is, $\varepsilon_V \to 0$ and $\varepsilon_u \to 0$ as the number of basis terms retained tend to infinity, that is, $N_1 \to \infty$ and $N_2 \to \infty$ respectively. Readers are referred to [5, 28] where this approach has been described and deployed to approximate the exact value function and optimal control policy. It is noted that, in the ideal case $\bar{\varepsilon}_V = \bar{\varepsilon}_u = 0$, the parameterizations can represent the exact PI iterates on Ω.

Lemma 2. *The weights W_i and U_i can be obtained by solving the following least-squares (LS) equation*

$$\tilde{\Theta}_i^N \begin{bmatrix} \mathrm{vec}(W_i) \\ \mathrm{vec}(U_i^T) \end{bmatrix} = \tilde{E}_i^N \tag{45}$$

for $N > N_1 + mN_2$ and

$$\begin{aligned} \tilde{\Theta}_i^N &= [\tilde{\Theta}_i(t_1), \ldots, \tilde{\Theta}_i(t_N)]^T \\ \tilde{E}_i^N &= [\tilde{E}_i(t_1), \ldots, \tilde{E}_i(t_N)]^T \end{aligned} \tag{46}$$

where

$$\begin{aligned} \tilde{E}_i(t) = &-I_{\tilde{\Psi}\tilde{\Psi}}\big(U_{i-1}^T \otimes U_{i-1}^T\big)\mathrm{vec}(R) \\ &- \int_t^{t+T} \big(X^T \tilde{Q} X + B_\vartheta(\rho X)\big)dt \end{aligned} \tag{47}$$

$$\tilde{\Theta}_i(t) = \begin{bmatrix} (\tilde{\Phi}(X(t+T)) - \tilde{\Phi}(X(t)))^T - \gamma I_{\tilde{\Phi}} \\ 2\big(I_{u\tilde{\Psi}}(R \otimes I_{N_2}) - I_{\tilde{\Psi}\tilde{\Psi}}(U_{i-1}^T R \otimes I_{N_2})\big) \end{bmatrix}^T \tag{48}$$

with

$$\begin{aligned} I_{\tilde{\Psi}\tilde{\Psi}} &= \int_t^{t+\mathcal{T}} \Big(\tilde{\Psi}^T(X) \otimes \tilde{\Psi}^T(X)\Big)\mathrm{d}t \\ I_{u\tilde{\Psi}} &= \int_t^{t+\mathcal{T}} \Big(u_s^T \otimes \tilde{\Psi}^T(X)\Big)\mathrm{d}t \\ I_{\tilde{\Phi}} &= \int_t^{t+\mathcal{T}} (\tilde{\Phi}^T(X) \otimes I_{N_1})\mathrm{d}t \end{aligned}$$

Proof. Substituting the approximations (42) and (43) for W_i, u_i, and u_{i+1} into (41) results in

$$\begin{aligned} W_i(\tilde{\Phi}(X(t)) - \tilde{\Phi}(X(t+T))) = &-\int_t^{t+T} 2\tilde{\Psi}^T(X)U_i^T R\tilde{v}_i dt \\ &- \int_t^{t+T} \big(X^T \tilde{Q} X + \tilde{u}_i^T R \tilde{u}_i + B_\vartheta(\rho X)\big)dt \\ &+ \gamma \int_t^{t+T} W_i \tilde{\Phi}(X) dt \end{aligned} \tag{49}$$

where $\tilde{u}_0 = u_0$, $\tilde{v}_i = u_s - \tilde{u}_i$. Thus,

$$\int_t^{t+T} \tilde{\Psi}^T(X)U_i^T R\tilde{v}_i dt = \int_t^{t+T} \tilde{\Psi}^T(X)U_i^T R(u_s - \tilde{u}_i)dt \tag{50}$$

One can derive the following equations

$$\begin{aligned} \int_t^{t+T} \tilde{\Psi}^T(X)U_i^T R u dt &= \int_t^{t+T} \Big(u_s^T R \otimes \tilde{\Psi}^T(X)\Big)\mathrm{vec}(U_i^T)dt \\ &= \int_t^{t+T} \Big(u_s^T \otimes \tilde{\Psi}^T(X)\Big)(R^T \otimes I_{N_2})\mathrm{vec}(U_i^T)dt \\ &= I_{u\tilde{\Psi}}\big(R \otimes I_{N_2}\big)\mathrm{vec}(U_i^T) \end{aligned} \tag{51}$$

$$\begin{aligned} \int_t^{t+T} \tilde{\Psi}^T(X)U_i^T R\tilde{u}_i dt &= \int_t^{t+T} \tilde{\Psi}^T(X)U_i^T R U_{i-1}\tilde{\Psi}(X)dt \\ &= \int_t^{t+T} \Big(\tilde{\Psi}^T(X)U_{i-1}^T R \otimes \tilde{\Psi}^T(X)\Big)\mathrm{vec}(U_i^T)dt \\ &= \int_t^{t+T} \Big(\tilde{\Psi}^T(X) \otimes \tilde{\Psi}^T(X)\Big)dt \\ &\quad \big(U_{i-1}^T R \otimes I_{N_2}\big)\mathrm{vec}(U_i^T) \\ &= I_{\tilde{\Psi}\tilde{\Psi}}\big(U_{i-1}^T R \otimes I_{N_2}\big)\mathrm{vec}(U_i^T) \end{aligned} \tag{52}$$

By substituting (51) and (52) into (50), it gives:

$$\begin{aligned} \int_t^{t+T} \tilde{\Psi}^T(X)U_i^T R\tilde{v}_i dt = &\, I_{u\tilde{\Psi}}(R \otimes I_{N_2})\mathrm{vec}(U_i^T) \\ &- I_{\tilde{\Psi}\tilde{\Psi}}\big(U_{i-1}^T R \otimes I_{N_2}\big)\mathrm{vec}(U_i^T) \end{aligned} \tag{53}$$

Furthermore, it can be noted that

$$\begin{aligned} \int_t^{t+T} \tilde{u}_i^T R\tilde{u}_i dt &= \int_t^{t+T} \tilde{\Psi}^T(X)U_{i-1}^T R U_{i-1}\tilde{\Psi}(X)dt \\ &= I_{\tilde{\Psi}\tilde{\Psi}}\big(U_{i-1}^T \otimes U_{i-1}^T\big)\mathrm{vec}(R) \end{aligned} \tag{54}$$

and

$$\begin{aligned} \gamma \int_t^{t+T} W_i \tilde{\Phi}(X) &= \gamma \int_t^{t+T} (\tilde{\Phi}^T(X) \otimes I_{N_1})dt \quad \mathrm{vec}(W_i) \\ &= \gamma I_{\tilde{\Phi}}\mathrm{vec}(W_i) \end{aligned} \tag{55}$$

Finally, by substituting (54) and (55) into (49), the following expression is derived

$$\begin{aligned} &\big((\tilde{\Phi}(X(t+T)) - \tilde{\Phi}(X(t)))^T - \gamma I_{\tilde{\Phi}}\big)W_i^T \\ &\quad + 2\big(I_{u\tilde{\Psi}}(R \otimes I_{N_2}) - I_{\tilde{\Psi}\tilde{\Psi}}(U_{i-1}^T R \otimes I_{N_2})\big)\mathrm{vec}(U_i^T) \\ &= -I_{\tilde{\Psi}\tilde{\Psi}}\big(U_{i-1}^T \otimes U_{i-1}^T\big)\mathrm{vec}(R) - \int_t^{t+T} X^T \tilde{Q} X + B_\vartheta(\rho X)dt \end{aligned} \tag{56}$$

(56) can be reformulated in regression form as follows

$$\tilde{\Theta}_i(t) \begin{bmatrix} \mathrm{vec}(W_i) \\ \mathrm{vec}(U_i^T) \end{bmatrix} = \tilde{E}_i(t) \tag{57}$$

with

$$\begin{aligned} \tilde{E}_i(t) = &-I_{\Psi\Psi}\big(U_{i-1}^T \otimes U_{i-1}^T\big)\mathrm{vec}(R) \\ &- \int_t^{t+T} \big(X^T \tilde{Q} X + B_\vartheta(\rho X)\big)dt \end{aligned} \tag{58}$$

$$\tilde{\Theta}_i(t) = \begin{bmatrix} (\tilde{\Phi}(X(t+T)) - \tilde{\Phi}(X(t)))^T - \gamma I_{\tilde{\Phi}} \\ 2\big(I_{u\tilde{\Psi}}(R \otimes I_{N_2}) - I_{\tilde{\Psi}\tilde{\Psi}}(U_{i-1}^T R \otimes I_{N_2})\big) \end{bmatrix}^T \tag{59}$$

(57) involves $N_1 + mN_2$ unknown parameters which can be determined using LS method. However, it is imperative to ensure a sufficiently rich data set of state and input information to guarantee an adequate number of equations for solving these unknown parameters. The collected data are saved in matrices $\tilde{\Theta}_i^N$ and $\tilde{E}_i^N$ defined as

$$\tilde{\Theta}_i^N = [\tilde{\Theta}_i(t_1), \ldots, \tilde{\Theta}_i(t_N)]^T, \quad \tilde{E}_i^N = [\tilde{E}_i(t_1), \ldots, \tilde{E}_i(t_N)]^T$$

As a result, the LS equation takes the form:

$$\tilde{\Theta}_i^N \begin{bmatrix} \text{vec}(W_i) \\ \text{vec}(U_i^T) \end{bmatrix} = \tilde{E}_i^N \tag{60}$$

with $N > N_1 + mN_2$. □

Before solving the pair (W_i, U_i) from (45), it is crucial to verify the uniqueness of the solution. To this end, Assumption 7 is introduced.

Assumption 7. For each $i = 0, 1, 2, \ldots$, there exists a sufficiently large integer $N > 0$, such that the following rank condition holds.

$$\text{rank}\left(\tilde{\Theta}_i^N\right) = N_1 + mN_2 \tag{61}$$

It is essential to collect sufficient sampled data (i.e., ensuring N is large enough for each iteration step i). The selection of exploration noise is crucial in this process. Typically, the rank condition can be verified computationally, although it cannot be determined analytically.

Remark 3. The rank condition in (61) is derived from the persistent excitation condition commonly used in adaptive control [29].

Figure 1, presents the structure of the developed algorithm and highlights its four distinct steps:

- **Initialization**: An initial safe and admissible policy is chosen such that $\tilde{u}_0 = U_0\tilde{\Psi}(X)$.
- **Safe Exploration**: Probing noise is added to the initial policy to explore the state space and collect rich data. The exploration policy u_{noisy} is adjusted by adding the solution of the QP problem in order to assure the safety of the system.
- **Safe PI**: After collecting data, the LS equation is solved, and the policy evaluation and improvement steps are computed iteratively until the convergence of the critic weights.
- **Safe Operation** The learned safe and optimal policy is used to generate the control input of the augmented system (6).

Remark 4. The convergence and safety properties in Theorem 1 are established for the exact PI iterates (V_i, π_i). This section provides an implementable off-policy realization using approximations. Under Assumption 6 and the rank condition in Assumption 7, the least-squares regression yields unique weights in the chosen function class, and the learned policy approaches the exact PI solution up to the approximation residual.

To assess the effectiveness of the algorithm, a simulation study over an academic system has been done.

5 | Simulation Results

In this section, a simulation example is presented to verify the efficiency of the proposed scheme and to demonstrate the design procedures.

Consider a nonlinear system described by the following differential equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1^3 - 0.5x_2 + u \end{aligned} \tag{62}$$

A sinusoidal reference trajectory is generated by

$$\dot{x}_r = 0.5\sqrt{5}cos(\sqrt{5}t) \tag{63}$$

Now, considering that only the states x_1 will be tracked, the error e, is defined as

$$e = x_1 - x_r \tag{64}$$

The safe set is described by $\mathscr{C} = \{x| \quad -2 < x_2 < 2\}$. The reward function (8) is considered with $Q = 8$, $R = 0.00001$ and $\gamma = 0.1$. The CBF B_ϑ is given by

$$B_\vartheta(\rho X) = B_{1,\vartheta}(\rho X) + B_{2,\vartheta}(\rho X) \tag{65}$$

with

$$\begin{aligned} B_{1,\vartheta}(\rho X) &= -\log\left(\frac{\vartheta h_1(\rho X)}{\vartheta h_1(\rho X) + 1}\right), \\ B_{2,\vartheta}(\rho X) &= -\log\left(\frac{\vartheta h_2(\rho X)}{\vartheta h_2(\rho X) + 1}\right) \end{aligned}$$

where $\vartheta = 400$, $h_1(\rho X) = -X_2^{min} + X_2$ and $h_2(\rho X) = X_2^{max} - X_2$ based on (2). The values of Q, R, and ϑ are selected to ensure that when the state x_2 approaches the boundaries of the safe set, a larger penalty is assigned to $B_\vartheta(\rho X)$, emphasizing safety. Conversely, when the states are within the safe set, the focus shifts to performance, giving more weight to the terms $x^T Qx$ and $u^T Ru$, while the role of $B_\vartheta(\rho X)$ becomes less significant.

From $t = 0$s to $t = 5$s, the exploration noise $\epsilon(t)$ is injected into the initial policy, with $\epsilon(t)$ being set to

$$\epsilon(t) = \sum_{l=1}^{12} 10\sin((2l-1)t) \tag{66}$$

The activation functions are considered, respectively, as

$$\tilde{\Phi}(X) = [X_1^2,\ X_2^2,\ X_3^2,\ X_1X_2,\ X_1X_3,\ X_2X_3,\ X_1^4,\ X_2^4,\ X_3^4,\ X_1^2X_2^2,\ X_1^2X_3^2,\ X_2^2X_3^2,\ X_1^3X_2,\ X_1^3X_3,\ X_2^3X_1,\ X_2^3X_3,\ X_3^3X_1,\ X_3^3X_2]^T$$

$$\tilde{\Psi}(X) = \left[X_1, X_2, X_3\right]^T.$$

These weights of the critic and actor are trained by finding the solution of (45) for $N = 250$. The input and state data are collected over each interval of $T = 0.02s$. The initial weights of the actor are set to $U_0 = [-2 \quad -16 \quad -60]$.

Based on Equation (36), the CBF criteria is formulated as

$$\begin{aligned} &L_f h_1(\rho X) + L_g h_1(\rho X)\epsilon(t) + \alpha_1(h_1(\rho X)) \\ &\qquad + L_g h_1(\rho X)(u + u_{safe}) \geq 0 \\ &L_f h_2(\rho X) + L_g h_2(\rho X)\epsilon(t) + \alpha_2(h_2(\rho X)) \\ &\qquad + L_g h_2(\rho X)(u + u_{safe}) \geq 0 \end{aligned} \tag{67}$$

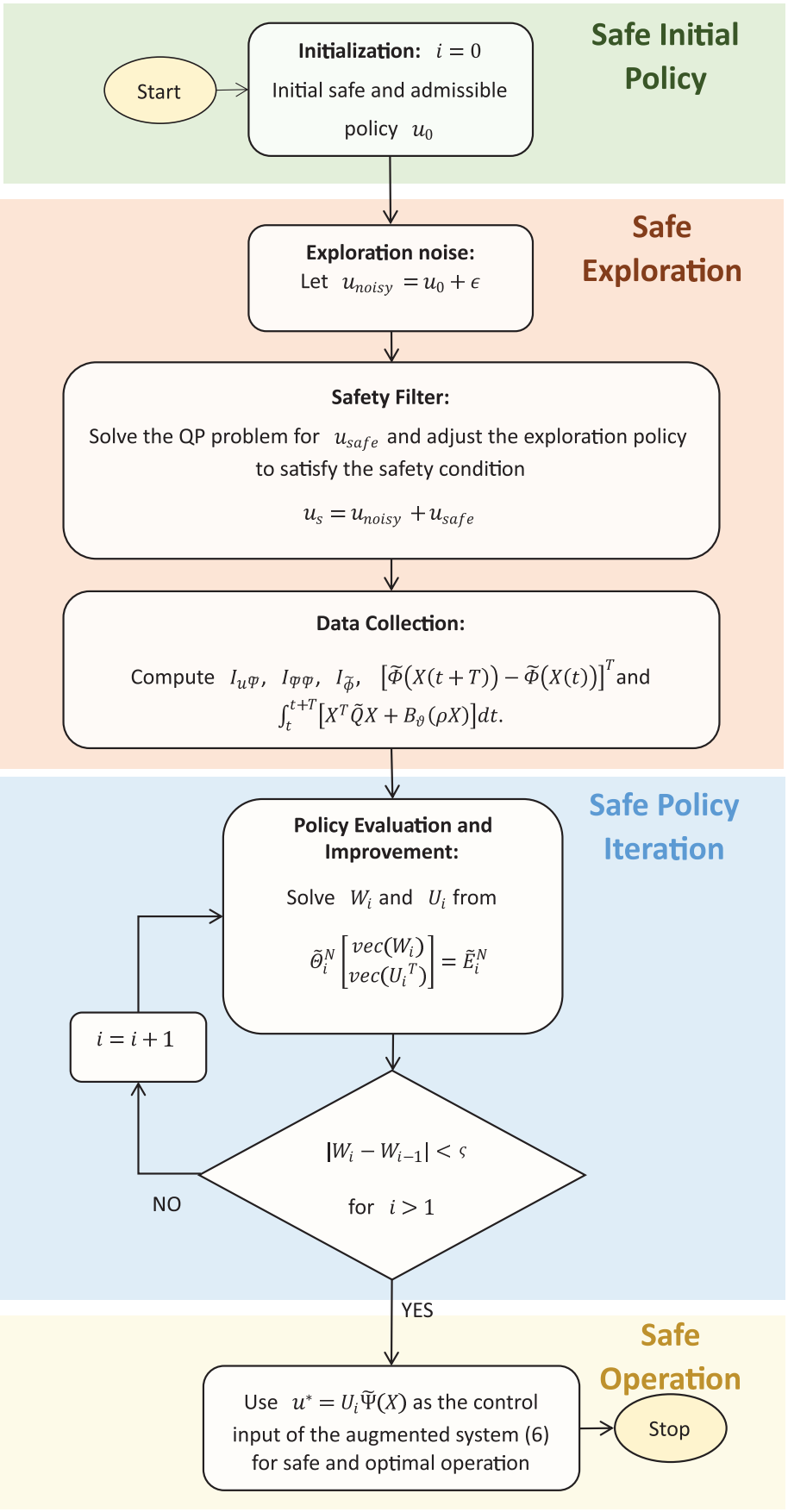

FIGURE 1 | Flowchart of Safe Off-Policy algorithm for tracking problem.

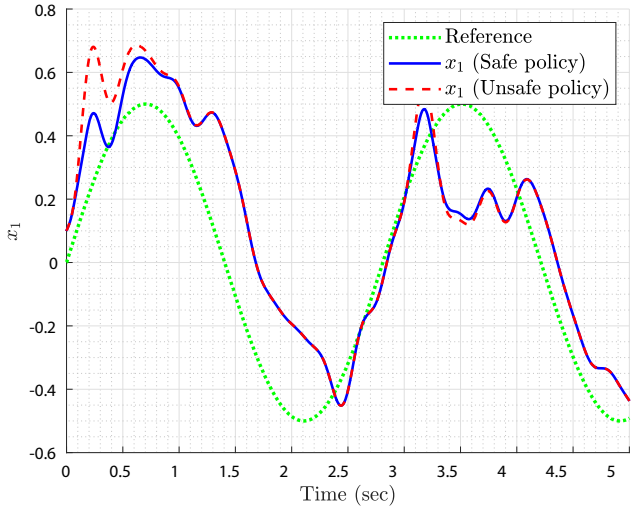

FIGURE 2 | Trajectory of x_1 during exploration.

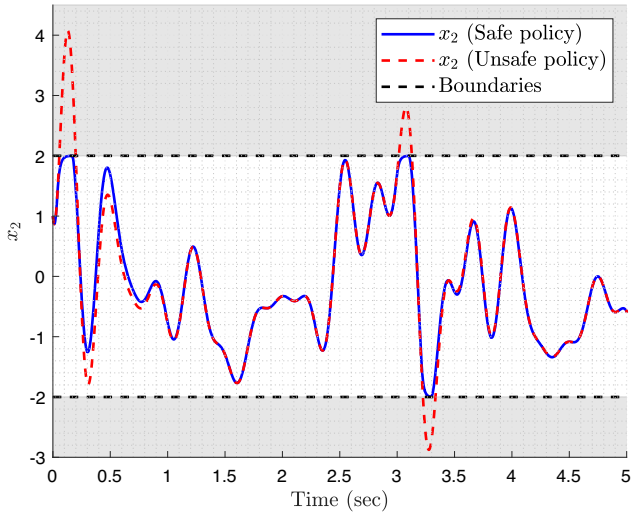

FIGURE 3 | Trajectory of x_2 during exploration.

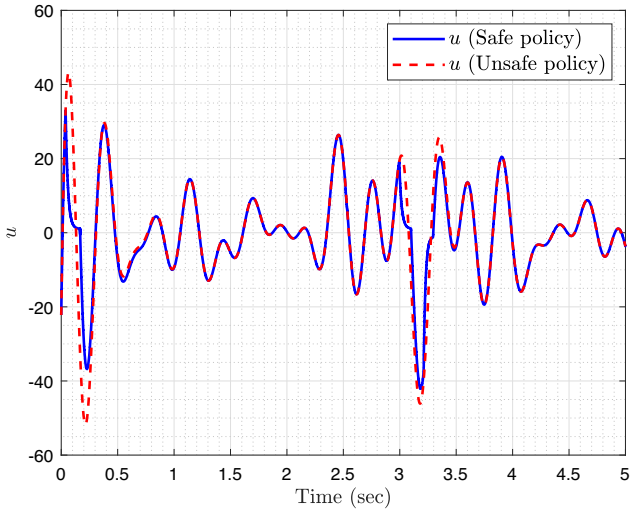

FIGURE 4 | Exploration policy under probing noise.

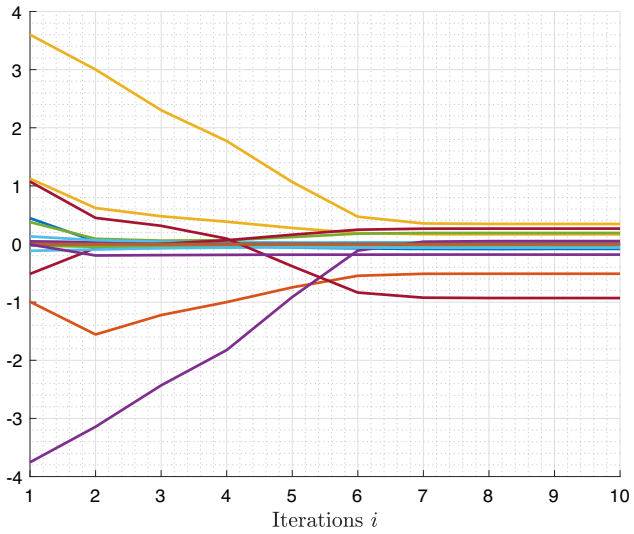

FIGURE 5 | Convergence of critic weights.

with $\alpha_1 = 60h_1(\rho X)$, $\alpha_2 = 50h_2(\rho X)$. These values are fixed to enable less conservative exploration. This selection allows the system to approach the boundaries more closely, allowing the collection of data not only from within the safe region but also from its vicinity.

In the exploration phase, Figure 2 presents the trajectories of the state x_1, showcasing both the safe case (blue curve) and the unsafe case (red curve). There is no significant difference between the two curves since there are no constraints on the state x_1. For the state x_2, it is clear that the system maintains safety during the exploration phase. However, when the QP problem is deactivated, it can be observed from the red curve that the state x_2 violates the safety boundary (dashed black line) (Figure 3). Figure 4 displays the noisy input applied for exploration purposes. It can be seen that, in order to ensure safety, slight adjustments are made to the unsafe policy (red curve), ultimately resulting in a secure policy (blue curve).

After collecting data, the safe tracking PI algorithm is iteratively computed until the critic weights converge. In this example, the algorithm has converged after 10 iterations as shown in Figure 5.

Figures 6 and 7 display the trajectory of the state variable x_1, during both phases: exploration and operational (exploitation). During the exploration phase, which lasts from t = 0 s to t = 5 s, the exploration input is applied to the system to collect data. Then the trained actor is used to control the system during the operational phase. Figure 6 displays the trajectory of the state x_1. In both the safe and unsafe scenarios, it is evident that, after applying the optimal learned policy, x_1 effectively tracks the reference, demonstrating the effectiveness of the policy in guiding the state variable toward the desired reference trajectory. In Figure 7, the trajectory of the state x_1 is presented. During the operational phase, it is noteworthy that both curves exhibit striking similarity. To validate the safety of the learned policy, it will be employed to control the system behavior from the initial states $X_0 = [0.1, 1, 0.1]^T$ and not the states where exploration was interrupted. Figure 8 displays three curves:

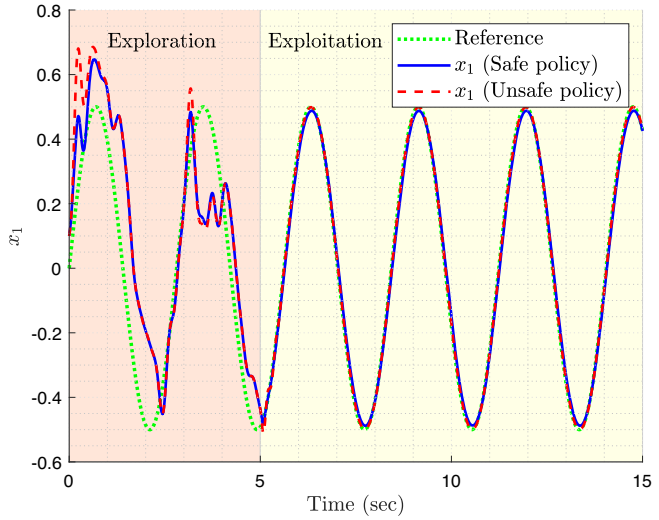

FIGURE 6 | Trajectory of x_1 during exploration and exploitation.

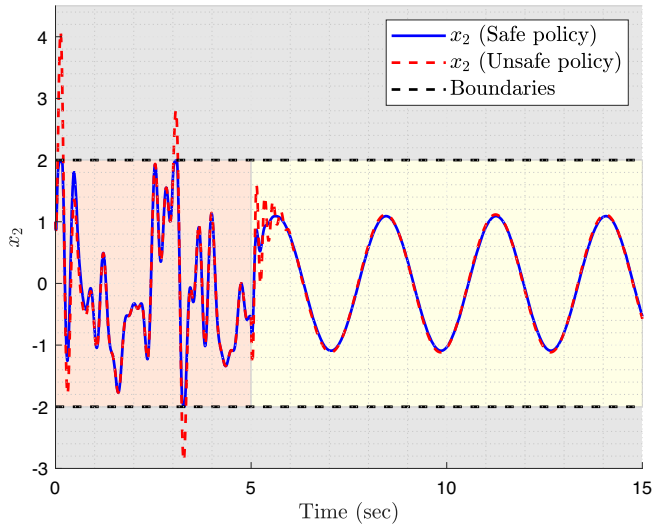

FIGURE 7 | Trajectory of x_2 during exploration and exploitation.

- the grey curve represents the trajectory of x_1 where the initial policy was applied to the system;
- the red curve illustrates the trajectory of x_1 for the classical off-policy algorithm, where there are no safety considerations during the exploration and the exploitation;
- the blue curve represents the trajectory of x_1 under the learned policy, where the exploration was done in a safe manner and the reward function was augmented with CBFs.

It can be deduced that the learned policy, whether in the safe or unsafe cases, does indeed ensure that x_1 follows the reference trajectory x_r. In contrast, when using the initial policy, a delay is observed between x_1 and the reference, underscoring the improved performance achieved through the learned policy.

In Figure 9, the two curves represent the same cases as in Figure 8, specifically for the state x_2. Notably, under the proposed algorithm, x_2 consistently remains within the safe region, demonstrating the efficiency of the approach in satisfying safety

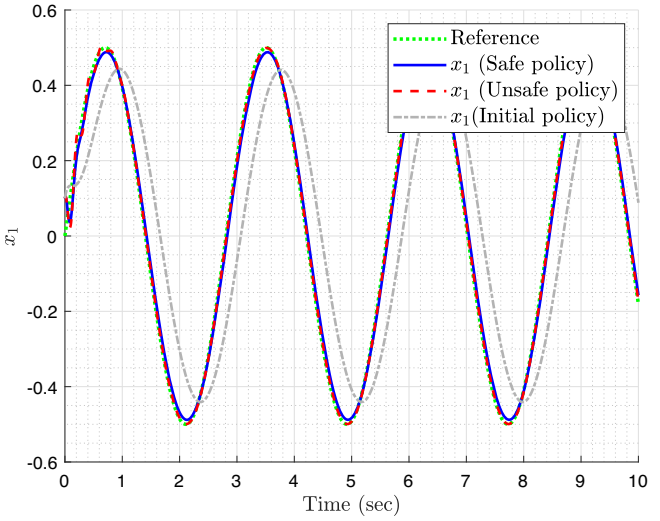

FIGURE 8 | Trajectory of x_1 during the operational phase under the learned policy.

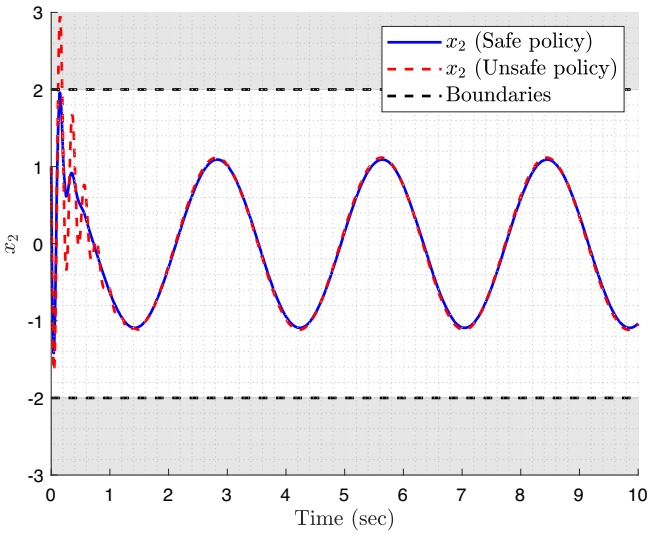

FIGURE 9 | Trajectory of x_2 during the operational phase under the learned policy.

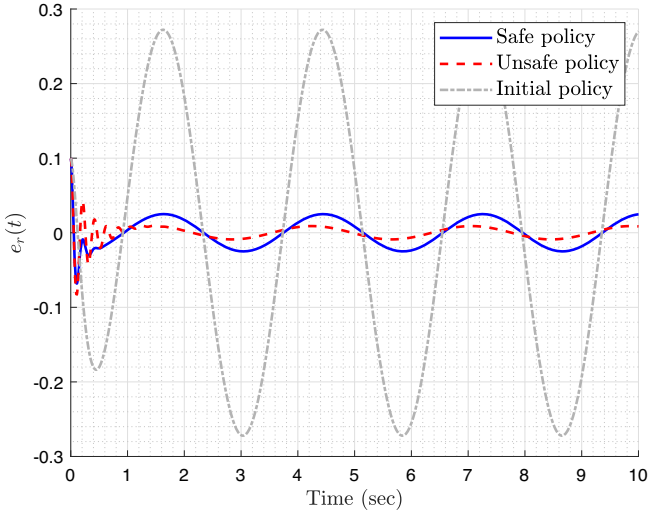

FIGURE 10 | Trajectory of e during the operational phase under the learned policy.

constraints. While applying the classical off-policy algorithm, x_2 would surpass the predefined safety boundaries, showing the advantages of the proposed method in ensuring safe operation. In Figure 10, the graph illustrates the evolution of the error between the state x_1 and the reference x_r for the three scenarios previously mentioned in Figure 8. The largest error is notably observed when employing the initial policy, indicating that it is not optimal. With the learned policy, the error significantly decreases, showing its effectiveness in trajectory tracking. However, it is worth noting that in the safe case, the error remains larger compared to the unsafe case. This is because the safe controller balances safety and performance, while the unsafe case prioritizes performance without taking into consideration any safety guarantees.

6 | Conclusion

This paper proposes a novel approach that ensures safe and optimal tracking control learning using off-policy based approach. It introduces guarantees of safety throughout both the exploration and exploitation phases. The exploration phase introduces probing noise to collect diverse and informative data, while simultaneously employing the QP problem to enforce safety constraints. Safe tracking PI is then iteratively computed to achieve a policy that balances safety and optimality. Simulation results assert the effectiveness of the algorithm in generating safe policies, even in the presence of probing noise, while significantly reducing the error between the reference and the state. This demonstrates that the learned policy successfully strikes a balance between safety and the optimality of performance.

It is important to note that while the proposed algorithm has demonstrated promising results in ensuring safety for tracking problems, future work will focus on addressing the challenge of relying on a system model to solve the QP problem. This limitation can be addressed by assuming only a nominal model is available and approximating model uncertainty using machine learning techniques such as neural networks or Gaussian processes.

Funding

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data supporting the findings of this study were generated through simulations using MATLAB within an academic system. These simulation data are not publicly available due to proprietary algorithms and ongoing related research. However, detailed descriptions of the simulation setup, parameters, and methodologies are provided within the manuscript to ensure the reproducibility of the results.

References

1. A. Alleyne, F. Allgöwer, A. Ames, et al., "Control for Societal-Scale Challenges: Road Map 2030," in *Proceedings of the 2022 IEEE CSS Workshop on Control for Societal-Scale Challenges* (IEEE Control Systems Society, 2023).

2. L. Brunke, M. Greeff, A. W. Hall, et al., "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning," *Annual Review of Control, Robotics, and Autonomous Systems* 5 (2022): 411–444.

3. K. Zhou and J. C. Doyle, *Essentials of Robust Control*, vol. 104 (Prentice Hall, 1998).

4. S. Zhang, D. H. Zhai, Y. Xiong, J. Lin, and Y. Xia, "Safety-Critical Control for Robotic Systems With Uncertain Model via Control Barrier Function," *International Journal of Robust and Nonlinear Control* 33, no. 6 (2023): 3661–3676.

5. F. L. Lewis and D. Vrabie, "Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control," *IEEE Circuits and Systems Magazine* 9, no. 3 (2009): 32–50, https://doi.org/10.1109/MCAS.2009.933854.

6. C. J. C. H. Watkins, "Learning From Delayed Rewards," 1989.

7. D. Vrabie and F. Lewis, "Neural Network Approach to Continuous-Time Direct Adaptive Optimal Control for Partially Unknown Nonlinear Systems," *Neural Networks* 22, no. 3 (2009): 237–246.

8. B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and Autonomous Control Using Reinforcement Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems* 29, no. 6 (2017): 2042–2062.

9. B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M. B. Naghibi-Sistani, "Reinforcement Q-Learning for Optimal Tracking Control of Linear Discrete-Time Systems With Unknown Dynamics," *Automatica* 50, no. 4 (2014): 1167–1175.

10. B. Kiumarsi and F. L. Lewis, "Actor–Critic-Based Optimal Tracking for Partially Unknown Nonlinear Discrete-Time Systems," *IEEE Transactions on Neural Networks and Learning Systems* 26, no. 1 (2014): 140–151.

11. H. Modares and F. L. Lewis, "Optimal Tracking Control of Nonlinear Partially-Unknown Constrained-Input Systems Using Integral Reinforcement Learning," *Automatica* 50, no. 7 (2014): 1780–1792.

12. C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan, and S. Xie, "Reinforcement Learning-Based Adaptive Optimal Exponential Tracking Control of Linear Systems With Unknown Dynamics," *IEEE Transactions on Automatic Control* 64, no. 11 (2019): 4423–4438.

13. H. Modares, F. L. Lewis, and Z. P. Jiang, "$H\infty$ Tracking Control of Completely Unknown Continuous-Time Systems via Off-Policy Reinforcement Learning," *IEEE Transactions on Neural Networks and Learning Systems* 26, no. 10 (2015): 2550–2562.

14. A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-Based Safe Learning With Gaussian Processes," in *Proceedings of the 53rd IEEE Conference on Decision and Control* (IEEE, 2014), 1424–1431.

15. J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems," *IEEE Transactions on Automatic Control* 64, no. 7 (2018): 2737–2752.

16. N. Kochdumper, H. Krasowski, X. Wang, S. Bak, and M. Althoff, "Provably Safe Reinforcement Learning via Action Projection Using Reachability Analysis and Polynomial Zonotopes," *IEEE Open Journal of Control Systems* 2 (2023): 79–92.

17. R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-To-End Safe Reinforcement Learning Through Barrier Functions for Safety-Critical Continuous Control Tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33 (2019), 3387–3395.

18. Y. Yang, Y. Yin, W. He, K. G. Vamvoudakis, H. Modares, and D. C. Wunsch, "Safety-Aware Reinforcement Learning Framework With an Actor-Critic-Barrier Structure," in *Proceedings of the 2019 American Control Conference (ACC)* (IEEE, 2019), 2352–2358.

19. M. H. Cohen and C. Belta, "Safe Exploration in Model-Based Reinforcement Learning Using Control Barrier Functions," *Automatica* 147 (2023): 110684.

20. D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed Coordination Control for Multi-Robot Networks Using Lyapunov-Like Barrier Functions," *IEEE Transactions on Automatic Control* 61, no. 3 (2015): 617–632.

21. Z. Marvi and B. Kiumarsi, "Safe Reinforcement Learning: A Control Barrier Function Optimization Approach," *International Journal of Robust and Nonlinear Control* 31, no. 6 (2021): 1923–1940.

22. S. Kanso, M. S. Jha, and D. Theilliol, "Off-Policy Model-Based End-To-End Safe Reinforcement Learning," *International Journal of Robust and Nonlinear Control*: 2806–2831, https://doi.org/10.1002/rnc.7109.

23. Y. Hu, J. Fu, and G. Wen, "Safe Reinforcement Learning for Model-Reference Trajectory Tracking of Uncertain Autonomous Vehicles With Model-Based Acceleration," *IEEE Transactions on Intelligent Vehicles* 8 (2023): 2332–2344.

24. Y. Jiang and Z. P. Jiang, *Robust Adaptive Dynamic Programming* (John Wiley & Sons, 2017).

25. A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control* 62, no. 8 (2016): 3861–3876.

26. X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of Control Barrier Functions for Safety Critical Control," *IFAC-PapersOnLine* 48, no. 27 (2015): 54–61.

27. 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 Magazine* 32, no. 6 (2012): 76–105.

28. M. Abu-Khalaf and F. L. Lewis, "Nearly Optimal Control Laws for Nonlinear Systems With Saturating Actuators Using a Neural Network HJB Approach," *Automatica* 41, no. 5 (2005): 779–791.

29. P. A. Ioannou and J. Sun, *Robust Adaptive Control* (Courier Corporation, 2012).

Appendix A

Before developing the proof of Theorem 1, Lemma 3 is given.

Lemma 3. *Under Assumption 4, the following holds*:

1. $$V^*(X) \le V_i(X); \tag{A1}$$

2. *for any* $V_{i-1} \in \mathcal{P}$, *satisfying*

$$\nabla V_{i-1}^T\big(F(X)+G(X)u_i\big) + \tilde{r}(X,u_i) - \gamma V_{i-1}(X) \le 0 \tag{A2}$$

it follows that $V_i(X) \le V_{i-1}(X)$;

3. $$\nabla V_i^T\big(F(X)+G(X)u_{i+1}\big) + X^T\tilde{Q}X + B_\vartheta(\rho X) - \gamma V_i(X) + u_{i+1}^T R u_{i+1} \le 0. \tag{A3}$$

Proof.

1. First, we want to prove that $V_i(X) \ge V^*(X)$. Under Assumption 4, we have

$$\nabla V^{*T}(F(X)+G(X)u^*) + \tilde{r}(X,u^*) - \gamma V^*(X) = 0 \tag{A4}$$

It follows that

$$\begin{aligned}
&\big(\nabla V_i - \nabla V^*\big)^T\big(F(X)+G(X)u_i\big) + u_i^T R u_i - u^{*T} R u^* + \\
&\nabla V^{*T}(X)G(x)(u_i-u^*) + \gamma\big(V^*(X) - V_i(X)\big) \\
&\quad = \big(\nabla V_i - \nabla V^*\big)^T\big(F(X)+G(X)u_i\big) - 2u^{*T}R(u_i-u^*) \\
&\qquad + u_i^T R u_i - u^{*T} R u^* + \gamma\big(V^*(X) - V_i(X)\big) \\
&\quad = \big(\nabla V_i - \nabla V^*\big)^T\big(F(X)+G(X)u_i\big) \\
&\qquad + (u^*-u_i)^T R(u^*-u_i) + \gamma\big(V^*(X) - V_i(X)\big) \\
&\quad = 0
\end{aligned} \tag{A5}$$

Multiplying both sides by $e^{-\gamma t}$, for all X_0, along the trajectories of the augmented system (6) with $u = u_i$ and $X(0) = X_0$, the following holds:

$$e^{-\gamma t_0}\big(V_i(X_0) - V^*(X_0)\big) = \int_0^\infty e^{-\gamma t}(u^*-u_i)^T R(u^*-u_i)dt \ge 0 \tag{A6}$$

which implies that $V_i(X) \ge V^*(X)$.

2. We have

$$\nabla V_{i-1}^T(X)\big(F(X)+G(X)u_i\big) + \tilde{r}(X,u_i) - \gamma V_{i-1}(X) \le 0 \tag{A7}$$

Let $m(X) \ge 0$, such that

$$\nabla V_{i-1}^T(X)\big(F(X)+G(X)u_i\big) + \tilde{r}(X,u_i) - \gamma V_{i-1}(X) = -m(X) \tag{A8}$$

Since $\nabla V_i^T(X)\big(F(X)+G(X)u_i\big) + \tilde{r}(X,u_i) - \gamma V_i(X) = 0$, it follows

$$\begin{aligned}
&\big(\nabla V_{i-1}(X) - \nabla V_i(X)\big)^T\big(F(X)+G(X)u_i\big) + \gamma\big(V_i - V_{i-1}\big) \\
&\qquad = -m(X)
\end{aligned} \tag{A9}$$

Multiplying both sides by $e^{-\gamma t}$, for all X_0, along the trajectories of the augmented system (6) with $u = u_i$ and $X(0) = X_0$, the following holds:

$$e^{-\gamma t_0}\big(V_i(X_0) - V_{i-1}(X_0)\big) = -\int_0^\infty e^{-\gamma t} m(X) < 0 \tag{A10}$$

Thus $V_i(X) < V_{i-1}(X)$.

3. The goal is to show that

$$\begin{aligned}
&\nabla V_i^T\big(F(X)+G(X)u_{i+1}\big) + X^T\tilde{Q}X + B_\vartheta(\rho X) \\
&\qquad - \gamma V_i(X) + u_{i+1}^T R u_{i+1} \le 0.
\end{aligned} \tag{A11}$$

By definition

$$\begin{aligned}
&\nabla V_i^T\big(F(X)+G(X)u_{i+1}\big) + X^T\tilde{Q}X + B_\vartheta(\rho X) - \gamma V_i(X) \\
&\quad + u_{i+1}^T R u_{i+1} \\
&= \nabla V_i^T(x)\big(F(X)+G(X)u_{i+1}\big) + X^T\tilde{Q}X + u_{i+1}^T R u_{i+1} \\
&\quad + B_\vartheta(\rho X) - \gamma V_i(X) \\
&\quad + \nabla V_i^T G(X)u_i - \nabla V_i^T G(X)u_i + u_i^T R u_i - u_i^T R u_i \\
&= \nabla V_i^T\big(F(X)+G(X)u_i\big) + X^T\tilde{Q}X + u_i^T R u_i + B_\vartheta(\rho X) \\
&\quad - \gamma V_i(X) + \nabla V_i^T G(X)(u_{i+1}-u_i) + u_{i+1}^T R u_{i+1} - u_i^T R u_i \\
&= \nabla V_i^T G(X)(u_{i+1}-u_i) + u_{i+1}^T R u_{i+1} - u_i^T R u_i \\
&= -2u_{i+1}^T R(u_{i+1}-u_i) + u_{i+1}^T R u_{i+1} - u_i^T R u_i \\
&= -(u_{i+1}-u_i)^T R(u_{i+1}-u_i) \le 0
\end{aligned}$$

The proof of Lemma 3 is complete. □

Now, the proof of Theorem 1 will be developed in the following.

Proof. First, Theorem 1.1 and Theorem 1.2 are shown to be true by induction, and it is proved that $V_i \in \mathcal{P}$, for all $i = 0, 1, \ldots$

a. For $i = 1$, it follows from Assumption 3, Lemma 3.1 and Lemma 3.2 that Theorem 1.1 and Theorem 1.3 are true. Under Assumption 3 and 4, $V^* \in \mathcal{P}$ and $V_0 \in \mathcal{P}$ thus $V_1 \in \mathcal{P}$.

b. Suppose Theorem 1.1 and Theorem 1.3 hold for $i = j > 1$ and $V_j \in \mathcal{P}$, We want to show that Theorem 1.1 and Theorem 1.3 also hold for $i = j + 1$ and $V_{j+1} \in \mathcal{P}$.

Since $V^* \in \mathcal{P}$ and $V_j \in \mathcal{P}$, we deduce that $V_{j+1} \in \mathcal{P}$. By Lemma 3.3, one obtains

$$\begin{aligned} &\nabla V_{j+1}^T(X)\big(F(X) + G(X)u_{j+2}\big) + X^T\tilde{Q}X + B_\vartheta(\rho X) \\ &\quad - \gamma V_{j+1}(X) + u_{j+2}^T R u_{j+2} \leq 0 \end{aligned} \tag{A12}$$

Along the solutions of system (6) for $u = u_{j+2}$, one obtains $\dot{V}_{j+1} \leq 0$. Since $V_{j+1} \in \mathcal{P}$, it is a well-defined Lyapunov function for the augmented system (6) with $u = u_{j+2}$. Therefore, u_{j+2} is a stabilizing policy which implies that Theorem 1.3 holds for $i = j + 1$.

From Lemma 3.2, we have $V_{j+2} \leq V_{j+1}$ and by induction assumption we have $V^*(X) \leq V_{j+1}(X) \leq V_j(X)$, which gives

$$V^*(X) \leq V_{j+2}(X) \leq V_{j+1}(X)$$

Hence, Theorem 1.1 holds for $i = j + 1$.

If such a pair (V, u) exists, we already know that the solution of safe-HJB is unique, and thus we can deduce that $V^* = V$ and $u^* = u$.

Now, it must be shown that at each iteration u_i is safe, and thus we want to show that the states under policy u_i remain in the safe set $\mathscr{C}$. Earlier, we have proved that $V^*(X) \leq V_{i+1}(X) \leq V_i(X) \leq V_0$, which implies that at each iteration V_i is bounded and consequently the reward $\tilde{r}(X, u_i)$ and the barrier function B_ϑ remains bounded after each policy improvement step. Moreover, B_ϑ tends to infinity near the boundary of the safe set, implying that the system states do not cross $\partial\mathscr{C}$. This in turn guarantees safety and proves that $u_i \in \mathscr{U}_s$. □