Off-policy model-based end-to-end safe reinforcement learning

Soha Kanso, Mayank Shekhar Jha, Didier Theilliol

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Off-Policy Model-Based End-to-End Safe Reinforcement Learning

Soha KANSO | **Mayank Shekhar JHA** | **Didier THEILLIOL**

[1]CRAN, UMR 7039, CNRS, Université de Lorraine, 54506 Vandoeuvre-lès-Nancy Cedex, France.

Correspondence
Mayank Shekhar JHA, CRAN, UMR 7039, CNRS, Université de Lorraine.
Email: mayank-shekhar.jha@univ-lorraine.fr

Abstract

Safety and stability considerations play a crucial role in the development of learning based strategies for control design of systems that require high levels of safety. Safe Reinforcement learning (RL) based approaches traditionally seek learning of the control laws that are optimal with respect to system performance whilst ensuring system stability and safety. In this paper, an off-policy safe RL based approach is proposed for nonlinear systems affine in control in continuous time. In this novel work, safety and stability are guaranteed during initialization and exploration phases by adjusting the control input with the solution of a quadratic programming (QP) problem combining both Input to State Stable - Control Lyapunov Function (ISS-CLF) and Robust control barrier function (R-CBF) conditions. Moreover, the safety of the learned policy is assured by augmenting the cost function with a CBF to maintain safety and optimize performance simultaneously. Novel mathematically rigorous proofs are provided to establish the stability and safety guarantees, offering a sound theoretical foundation for the approach.
To demonstrate the effectiveness of the algorithm, two examples are presented: engine surge and stall dynamics, and an unstable nonlinear system

KEYWORDS
Control barrier function, control Lyapunov function, model-based control, reinforcement learning, safety.

1 | INTRODUCTION

In recent years, safety-critical systems[1] have become increasingly prevalent in various domains, such as transportation, air-traffic control, nuclear plants and automated industrial processes to name a few. Consequently, it has become extremely essential to ensure safety in the design of control systems to mitigate potential risks[2].
However, ensuring safety is not the only requirement for control system design. One of the crucial aspects is the consideration of input and state constraints, which restrict the behavior of the system to safe and acceptable bounds, preventing it from exceeding its limits and causing damage[2]. There has been a growing interest in the development of optimal controllers that achieve predefined performance while satisfying safety constraints. This task poses significant challenges, particularly since system dynamics exhibit uncertainty and are subject to changes over time[3,4]. Therefore, a recent focus has been on developing learning-based controllers that can balance the trade-offs between safety and performance whilst considering uncertainties of system dynamics[5].

Reinforcement learning (RL) has emerged as a powerful machine learning tool for designing optimal controllers for uncertain systems by iteratively interacting with the environment[6]. Its ability to operate in real-time and adapt to dynamic system changes makes it a promising approach for controlling uncertain systems and for learning robust and optimal control policies[7,8]. RL algorithms typically operate in two phases: exploration and exploitation[9]. During the exploration phase, random noisy inputs are applied to the system to collect rich data. This data is then used in the exploitation phase to optimize the control policy. However, under uncertainty, absence of complete dynamics knowledge can put the RL agent at risk of stability or safety violation, including input and state constraints, which can further complicate the learning of safe and optimal control policies. Moreover, the use of exploration noise can lead to visiting unsafe regions which can lead to catastrophic outcomes, such as

Abbreviations: RL, reinforcement learning; R-CBF, robust control barrier function; ISS-CLF, input to state stable - control Lyapunov function; MPC, model predictive control; QP, quadratic programming; PI, policy iteration; VI, value iteration; HJI, Hamilton–Jacobi–Isaacs; HJB, Hamilton–Jacobi–Bellman.

damage to the agent or failure to accomplish the task. Balancing the trade-offs between exploration and exploitation while ensuring safety and stability is a challenging problem that must be carefully addressed, especially in practical applications.
Different approaches have been proposed in the literature to solve the safe RL problem [10 11]. One family of strategies is based on modifying the optimality criterion such as worst case criterion [12], risk-sensitive criterion [13] and constrained criterion [14]. Another approach involves modifying the exploration process itself, which can be done by incorporating external knowledge into the exploration process, such as using teacher advice [15], expert demonstrations [16] or prior knowledge about the environment [17]. Additionally, incorporating a risk metric to guide the exploration process [18] can also be an effective approach to ensuring safety. Moreover, reachability analysis methods have been employed to address safety in the exploration process. [19] proposes a method for computing the backward reachable set starting from the obstacles, where a state is safe for all actions when outside of the backward reachable set. This later is obtained by finding the solution of a time-dependent HJI (Hamilton–Jacobi–Isaacs) equation. This method has the disadvantage that a different backward reachable set has to be computed for each obstacle which is computationally expensive.

Control barrier functions (CBF) based approaches are another prominent method of action projection techniques that have been widely used for safety verification and control synthesis of nonlinear systems [20]. CBFs provide a framework for designing control policies that guarantee forward invariance of the safe set such that the system remains in a safe set and never crosses the boundary. CBFs have been incorporated within the Model Predictive Control (MPC) framework for handling state and input constraints. In [21 22], the problem is formulated as an unconstrained optimization problem where a re-centred barrier function is directly added to the MPC cost function with the origin within the safe set. [23] proposes a CBF candidate based approach to deal with scenarios where safety and optimality may remain in conflict, and the safe set does not include the origin. Furthermore, recent works have shown that control barrier functions can be used in RL frameworks for safe and efficient exploration of complex environments. By incorporating CBF into the reward function [24], RL agents can learn policies that guarantee safety while achieving high performance on the given task.

On the other hand, Quadratic Programming (QP) based approaches can be employed to ensure the safety during the exploration phase by making minimal adjustments to the unsafe policy while satisfying CBF conditions [25]. However, herein, the availability of the system knowledge is essential making it challenging to integrate with RL frameworks that do not rely on explicit system models.
For nonlinear systems, [26] develops a safe exploration scheme for jointly learning the dynamics of an uncertain control system and the optimal value function/policy. The proposed approach uses Lyapunov-like barrier functions [27] to build robust safeguarding controller that can guarantee safety when combined with an arbitrary learning-based control policy. The safeguarding controller is leveraged to develop a safe exploration scheme in which the value function is learned online via simulation of experience, addressing the trade-off between exploration and safety. In [24], off-policy RL algorithm is employed to learn an optimal safe policy that minimizes a cost augmented by a CBF, while a safe and potentially conservative policy is applied for data collection during the learning process.
However, these aforementioned approaches require data collection both from the safe region as well as the vicinity of the safety boundary where the CBF is active. This however, renders the system unsafe during initialization as well as exploration phase. Additionally, the system should remain input-to-state stable (ISS) even in the presence of noise during the exploration, which can be difficult to achieve in practice. As such, the existing approaches fail to propose safe and admissible initialization and ensure safety during exploration and exploitation in a uniform manner. Notably, some methods such as Value Iteration (VI) and λ-Policy Iteration (λ-PI) [28] do not necessitate the initialization with an admissible policy. However, in this work, the specific focus is on the Policy Iteration (PI) algorithm where the initialisation is a crucial step.

To address the existing scientific gap, this article proposes a novel safe off-policy approach for nonlinear systems that guarantees safety at the three levels: initialization, exploration and exploitation. To that end, the proposed algorithm learns a control law that combines an ISS-control Lyapunov function (ISS-CLF) and a robust control barrier function (R-CBF) to ensure the safety and stability of the system during the initialization and exploration phases. The ISS-CLF is used to ensure the admissibility of the policy and the system's stability, while the R-CBF is used to ensure that the system remains safe during exploration, even in the presence of probing noise. To further guarantee the learned policy's safety, the reward function is augmented with a barrier function that penalizes the policy for taking actions that violate safety constraints.
The main contributions of this article lies on:

- *Safe and admissible initial policy:* In reinforcement learning, a critical condition for PI algorithms is the initialization with an admissible policy. In this approach, policies are initially generated randomly and subsequently subjected to modifications through a safety and stability filter.
- *Safe exploration and exploitation*: This approach ensures the collection of rich data from both the safe region and the vicinity of the safety boundary during the exploration phase. It's achieved by adjusting the unsafe policy with the solution of a QP problem. Notably, the system's stability remains guaranteed throughout the exploration phase, even after the addition of probing noise. During exploitation, the reward function is augmented by adding the barrier function, assuring the safety of the learned policy.
- Providing novel rigorous mathematical proofs to establish stability and safety guarantees of the developed algorithm.

The effectiveness of the proposed scheme is demonstrated through simulation on nonlinear systems.

The paper is organized as follows. Problem statement, background information and preliminaries are given in Section 2. Section 3 presents the proposed approach with safety and stability proofs. Section 4 examines the feasibility of the proposed approach using two examples. 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. C^1 denotes the set of continuously differentiable functions. For a differentiable function $V(x)$ and a vector $f(x)$, the notation $L_fV(x)$ corresponds to $\frac{\partial V}{\partial x}f(x)$. The symbol $\otimes$ denotes the Kronecker product. The Table 1 of notation contains all the other variables and their respective definitions.

2 | PROBLEM STATEMENT

This paper focuses on the analysis and control of nonlinear systems described by the following differential equation, which is affine in control input:

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

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

$$\mathscr{C} = \{x| \quad h(x) \geq 0\}, \tag{2}$$

for a smooth function h: $\mathbb{R}^n \to \mathbb{R}$. The objective is control law design $u(t)$ that optimizes performance function whilst assuring safety specifications. The safety objective is to guarantee that system states never leave the safe set $\mathscr{C}$ as the system's states evolve according to (1).

Before starting with the solution formulation, a brief overview of some concepts in RL, CLF, and CBF is provided.

2.1 | Reinforcement Learning

The objective of RL algorithms is to find a control policy $u(t)$ that minimizes a cumulative reward function over an infinite time horizon such as:

$$V(x_0) = \int_0^\infty r(x,u)dt, \quad x(0) = x_0 \tag{3}$$

where $r(x,u) = q(x) + u^T Ru$, $q(x)$ a positive definite function for all $x \in \mathbb{R}^n$, and R is symmetric and positive definite. Assume that the system (1) is stabilizable on some set $\mathscr{C}$, implying that there exists a control policy $u(t)$ such that the closed-loop system is asymptotically stable on $\mathscr{C}$. A control policy $u(t)$ is considered admissible if it stabilizes the system and leads to a finite cost V. The value function (3) can be expanded as follows:

$$V(x_0) = \int_0^T r(x,u)dt + \int_T^\infty r(x,u)dt \tag{4}$$

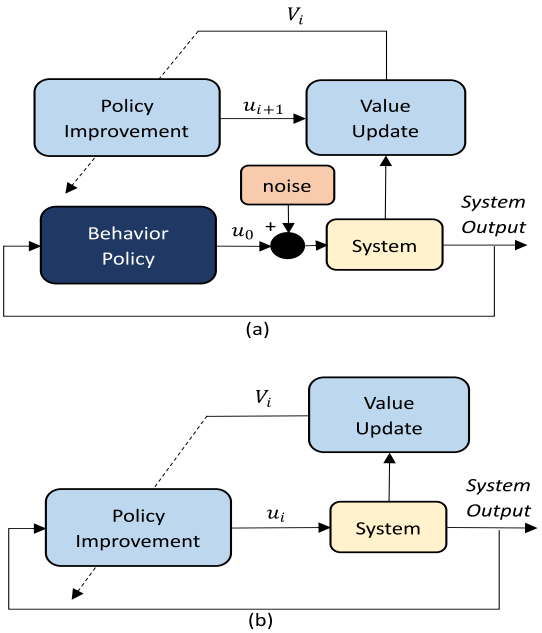

FIGURE 1 Two different categories of PI. In on-policy, the policy applied to the system (behavior policy) to generate data is the same policy being learned (learned policy). On the other hand, in off-policy RL, these two policies are separated and can be different. (a) Off-policy PI block diagram. (b) On-policy PI block diagram.

If $V \in C^1$, then (4) becomes

$$\lim_{T\to 0} \frac{V(x_0) - V(x(T))}{T} = \lim_{T\to 0} \frac{1}{T} \int_0^T r(x,u)dt \tag{5}$$

which gives

$$\dot{V} = \nabla V^T [f(x) + g(x)u] = -q(x) - u^T R u \tag{6}$$

(6) is the infinitesimal form of (3), it is in fact the Lyapunov equation (LE) for nonlinear systems.

$$LE(V,u) \triangleq \nabla V^T [f(x) + g(x)u] + r(x,u) = 0, \quad V(0) = 0. \tag{7}$$

The optimal control policy is obtained by

$$u^* = \underset{u}{argmin}\ (\nabla V^{*T}[f(x) + g(x)u] + r(x,u)) = -\frac{1}{2} R^{-1} g^T(x) \nabla V^*(x) \tag{8}$$

where $V^*(x_0)$ is optimal cost with respect to the initial condition $x(0) = x_0$, given by

$$V^*(x_0) = \min_u \int_0^\infty r(x,u)dt \quad \forall x \in \mathscr{X} \tag{9}$$

By substituting the optimal control (8) in (7), the LE becomes the well established Hamilton–Jacobi–Bellman (HJB) equation:

$$H(V^*)(x) \triangleq \nabla V^{*T}(x) f(x) + q(x) - \frac{1}{4} \nabla V^{*T}(x) g(x) R^{-1} g^T(x) \nabla V^*(x) = 0 \tag{10}$$

The optimal control problem is solved by finding the solution of HJB equation (10) with respect to the value function V^*. Then, by substituting the solution in (8) the optimal control is obtained. PI Algorithm 1 [6] is a widely used RL method that consists of two stages: policy evaluation (value update) and policy improvement.

PI algorithm can be applied using either on-policy or off-policy methods [29] (Fig. 1). On-policy based PI algorithms improve the same policy used to make decisions. On the other hand, in off-policy based PI algorithms, a behavior policy is used to generate data, and it may be unrelated to the evaluated and improved policy known by the target policy. Off-policy methods are

Algorithm 1 Policy Iteration Algorithm

Initialization. Initialize u_0 with an admissible policy.

Policy Evaluation. Update the value using:

$$\nabla V_i^T(x)[f(x) + g(x)u_i] + r(x,u_i) = 0 \tag{11}$$

Policy Improvement. The control policy is improved by:

$$u_{i+1}(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla V_i(x) \tag{12}$$

more efficient as they re-utilize the stream of experiences obtained by executing a given behavior policy to update several value functions associated with different learning policies[30]. Additionally, PI algorithms consider the effect of probing noise required for exploration. Adding such noise during exploration can be risky as it can lead to

- unstable behavior;
- exploration actions that may lead to undesirable or unsafe states.

One approach to ensuring safety during exploration is through the use of CBFs.

2.2 | Combining Control Barrier and Control Lyapunov Functions

A Barrier function (BF) is a function that is positive within the safe set and increases to infinity as it approaches the boundary of the set. It has a negative derivative in the vicinity of the boundary, which prevents it from reaching infinity. Existence of a BF within a given set implies the forward invariance of the set under the system's dynamics [39]. If the initial state is in the given set, the state remains in the set as time evolves. The presented formulation of BFs allows for a straightforward extension of these concepts to control systems by introducing the notion of CBFs defined in[31] as follows.

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

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))} \tag{13}$$

$$\inf_{u\in\mathscr{U}} [L_f B(x) + L_g B(x)u - \alpha_B(h(x))] \leq 0 \qquad \forall x \in Int\mathscr{C} \tag{14}$$

Condition (13) implies that the CBF behaves like the function $\frac{1}{\bar{\alpha}(h(x))}$, where $\bar{\alpha}$ is a class κ function. This means that $B(x)$ tends to infinity as $h(x)$ approaches zero (the states approach the boundaries), and B equals zero as $h(x)$ becomes very large (far from the boundaries). Additionally, the condition (14) allows the CBF to increase rapidly when solutions are far from $\partial\mathscr{C}$, and this growth gradually decreases as solutions approach $\partial\mathscr{C}$. The CBF serves as a repulsive force that prevents the trajectory from crossing the boundary of the safe set. Therefore, its properties are essential in establishing the forward invariance of the set $\mathscr{C}$ under the system's dynamics. Although any CBF function that satisfies Definition 1 can be employed, the following candidate is adopted in this article[24 23]:

$$B_\gamma(x) = -log(\frac{\gamma h(x)}{\gamma h(x)+1}) \tag{15}$$

where $\gamma > 0$ plays a crucial role in determining the behavior of the CBF. It controls the rate of decay of $B_\gamma(x)$ as the system moves away from the safety boundary. Specifically, larger values of γ result in faster attenuation. By a proper selection of γ, $B_\gamma(x)$ can reach very close to zero as shown in Fig. 2.

The CBF concept is closely related to CLF. While CBFs are used to enforce constraints on the system's behavior, CLFs are employed to guarantee the closed-loop system dynamics stability. The standard definition[32] of a CLF for system (1) is given as follows.

Definition 2. A positive definite, radially unbounded, differentiable function $V(x)$ is a CLF if there exists a class κ function $\alpha(x)$ such that for all $x \in \mathscr{X}$, $x \neq 0$,

$$\inf_{u \in \mathscr{U}} \left[L_f V(x) + L_g V(x) u + \alpha(\|x\|) \right] \leq 0 \tag{16}$$

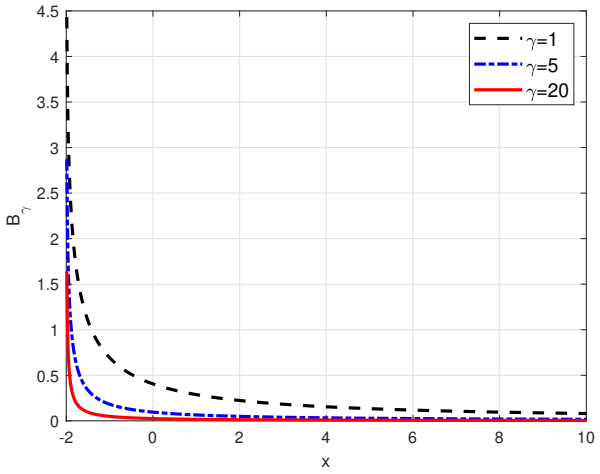

FIGURE 2 Effect of γ on B_γ for $h(x) = x + 1.5$.

The combination of CBFs and CLFs provides a powerful framework for designing control laws that not only stabilize the system but also ensure that it operates safely and within operational constraints. One effective method is based on QP where in control input based quadratic function is optimised subjected to CBF and CLF constraints that are affine in control[31 33]. The problem is formulated as:

QP Problem : Find the control input u and the relaxation variable δ that satisfy

$$\begin{aligned} \min_{u,\delta} \quad & \frac{1}{2}(u^T u + \ell \delta^T \delta) \\ s.t. \quad & L_f V(x) + \alpha(\|x\|) + L_g V(x) u + \delta \leq 0 \\ & L_f B_\gamma(x) - \alpha_B(h(x)) + L_g B_\gamma(x) u \leq 0 \end{aligned} \tag{17}$$

where $\ell \geq 1$ is a large constant intended to render δ in the solution as small as possible. In[34], it was demonstrated that the obtained controller is Lipschitz continuous for $x \in int\mathscr{C}$ with $L_g B(x) \neq 0$.

The following sections provide the novel propositions and proofs of this paper. The objective of the proposed methodology is to ensure safety throughout both the exploration and exploitation phases by incorporating CBFs.

3 | SAFE REINFORCEMENT LEARNING

To ensure safety of the learned policy, the reward function is augmented with a CBF function $B_\gamma(x)$ and the cost function defined in (3) is modified to

$$W(x_0) = \int_0^\infty r_{safe}(x,u)dt, \quad x(0) = x_0 \tag{18}$$

with

$$r_{safe}(x,u) = q(x) + u^T R u + B_\gamma(x) \tag{19}$$

The proposed cost function imposes significant penalties for points that approach a constraint boundary, thereby changing the controller dynamics in those regions. As a result, a more cautious control action is enforced when operating near the safety

boundary.
This formulation allows to determine a control policy that guarantees the satisfaction of all inequality constraints while ensuring a smooth transition between interior and boundary of the safe set. Before solving the optimal control problem, Definition 3 is introduced and it is assumed that there exist at least one safe admissible policy, as stated in Assumption 1.

Definition 3. The set of safe inputs $\mathscr{U}_c$ for the current state x is defined as

$$\mathscr{U}_c = \{u \in \mathbb{R}^m | x_u \in int\mathscr{C}\} \tag{20}$$

x_u is the state of the system evolved by the input u.
The set of safe and admissible inputs is then defined by

$$\mathscr{U}_a = \mathscr{U} \cap \mathscr{U}_c \tag{21}$$

where $\mathscr{U}$ is the admissible control policy for the optimal control problem (3).

Assumption 1. *There exists a safe feedback control policy $u_0 : \mathscr{C} \to \mathscr{U}_a$ that asymptotically stabilizes the system (1) at the origin and the associated cost defined in (18) is finite.*

The safe Lyapunov equation (Safe-LE) for the nonlinear system is given by

$$\nabla W^T[f(x) + g(x)u] + r_{safe}(x,u) = 0, \quad W(0) = 0. \tag{22}$$

The safe Hamiltonian function H_{safe} is defined by

$$H_{safe}(W)(x) \triangleq \nabla W^T(x)f(x) + q(x) + B_\gamma(x) - \frac{1}{4}\nabla W^T(x)g(x)R^{-1}g^T(x)\nabla W(x) \tag{23}$$

It is assumed that there exists a safe optimal control policy implying the existence of an optimal value function that satisfies the safe-HJB equation defined by

$$H_{safe}(W^*(x)) = 0 \tag{24}$$

where $W^*(x)$ is a well-defined Lyapunov function for the closed-loop system (1) defined as

$$W^* = \min_u \int_t^\infty [q(x) + u^T R u + B_\gamma(x)]d\tau \tag{25}$$

and u^* is the safe optimal control policy

$$u^*(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla W^*(x) \tag{26}$$

that asymptotically stabilizes the system at $x = 0$.

Assumption 2. *There exists $W^* \in \mathcal{P}$, where $\mathcal{P}$ is set of all functions in C^1 that are positive definite and radially bounded, such that the safe-HJB equation (24) holds.*

Lemma 1 establishes the uniqueness of solution to the Safe-HJB equation (24).

Lemma 1. *W^* is the unique solution to the Safe-HJB equation (24), given that $W^* \in \mathcal{P}$.*

Proof. Let $\tilde{W} \in \mathcal{P}$ be another solution to (24). Then, along the solutions of the closed-loop system composed of (1) and the control policy

$$\tilde{u}(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla\tilde{W}(x) \tag{27}$$

We have

$$\nabla W^{*T}[f(x) + g(x)u^*] + r_{safe}(x,u^*) = 0 \tag{28}$$

and

$$\nabla\tilde{W}^T[f(x) + g(x)\tilde{u}] + r_{safe}(x,\tilde{u}) = 0 \tag{29}$$

By subtracting (28) from (29), it gives:

$$
\begin{aligned}
&[\nabla\tilde{W}(x)-\nabla W^*(x)]^T f(x)+\nabla\tilde{W}^T(x)g(x)\tilde{u}-\nabla W^{*T}(x)g(x)u^*+\nabla\tilde{W}^T(x)g(x)u^*-\nabla\tilde{W}^T(x)g(x)u^*+\tilde{u}^TR\tilde{u}-u^{*T}Ru^*\\
&\quad=[\nabla\tilde{W}(x)-\nabla W^*(x)]^T[f(x)-g(x)u^*]+\nabla\tilde{W}^T(x)g(x)(\tilde{u}-u^*)+\tilde{u}^TR\tilde{u}-u^{*T}Ru^*\\
&\quad=[\nabla\tilde{W}(x)-\nabla W^*(x)]^T[f(x)-g(x)u^*]-2\tilde{u}^TR[\tilde{u}-u^*]+\tilde{u}^TR\tilde{u}-u^{*T}Ru^*\\
&\quad=[\nabla\tilde{W}(x)-\nabla W^*(x)]^T[f(x)-g(x)u^*]-(\tilde{u}-u^*)^TR(\tilde{u}-u^*)\\
&\quad=0
\end{aligned}
\tag{30}
$$

Therefore, for any $x_0\in\mathscr{C}$, along the trajectories of system (1) with $u=u^*$, it gives

$$
[\tilde{W}(x_T)-W^*(x_T)]-[\tilde{W}(x_0)-W^*(x_0)]=\int_0^T(\tilde{u}-u^*)^TR(\tilde{u}-u^*)dt \tag{31}
$$

Since u^* is stabilizing, $\lim_{T\to+\infty} W^*(x(T))=0$ and $\lim_{T\to+\infty}\tilde{W}^*(x(T))=0$, it follows that

$$
-[\tilde{W}(x_0)-W^*(x_0)]=\int_0^T(\tilde{u}-u^*)^TR(\tilde{u}-u^*)dt\geq 0 \tag{32}
$$

Thus $\tilde{W}(x)\leq W^*(x),\forall x\in\mathscr{C}$. Moreover, by subtracting (29) from (28), it gives

$$
[\nabla W^*(x)-\nabla\tilde{W}(x)]^T[f(x)-g(x)\tilde{u}]-(\tilde{u}-u^*)^TR(\tilde{u}-u^*)=0 \tag{33}
$$

By following the same steps, we obtain $\tilde{W}(x)\geq W^*(x)$.
Thus, it can be concluded that $\tilde{W}(x)=W^*(x),\forall x\in\mathscr{C}$. □

3.1 | Safe Policy Iteration

As the nonlinear safe-HJB equation (24) is difficult to be solved analytically, a modified version of PI Algorithm 2 can be employed to approximate the solution by iteratively updating the value function W (34) and improving the policy (35).

Algorithm 2 Safe Policy Iteration Algorithm

Initialization. Initialize u_0 with a safe and admissible policy such as $u_0\in\mathscr{U}_a$.

Policy Evaluation. Update the value using:

$$
\nabla W_i^T(x)[f(x)+g(x)u_i]+r_{safe}(x,u_i)=0 \tag{34}
$$

Policy Improvement. The control policy is improved by:

$$
u_{i+1}(x)=-\frac{1}{2}R^{-1}g^T\nabla W_i(x) \tag{35}
$$

The convergence property of the proposed safe-PI algorithm is given in Theorem 1 [35]. It demonstrates that, at each iteration, the algorithm maintains a safe policy, ensuring the invariance of the safe set.

Theorem 1. *Suppose Assumptions 1 and 2 hold, and the solution $W_i(x)\in C^1$ satisfying (34) exists for $i=0,1,\ldots$. Then, the following properties hold $\forall i=0,1,\ldots$.*

1. *$W^*(x)\leq W_{i+1}(x)\leq W_i(x)\ \forall x\in\mathscr{C}$.*
2. *u_i is stabilizing.*
3. *Let $\lim_{i\to\infty} W_i(x_0)=W(x_0)$ and $\lim_{i\to\infty} u_i(x_0)=u(x_0)$, $\forall x_0\in\mathscr{C}$. Then $W^*=W$ and $u^*=u$, if $W\in C^1$.*
4. *For $u=u_i$, $x_u\in int\mathscr{C}$, thus $u_i\in\mathscr{U}_c$.*

The proof of Theorem 1 is given in the appendix section 6.1.

Safe-PI algorithm has been proven to be effective in ensuring safety of the learned policy[24]. However, it comes with few challenges that need to be addressed. One of the main challenges is that the initial policy must be safe and admissible, which can be difficult to ensure in complex systems. Another challenge arises during the data collection stage, where probing noise is often added to the system to explore and improve learning. While exploration noise can provide valuable information about the system, it can also violate safety constraints and potentially destabilize the system. To ensure the stability of the closed-loop system, it is important to satisfy the input-to-state stability condition when the probing noise is considered as an input. To address these challenges, a new method that combines Safe-PI algorithm with a novel exploration strategy is developed in the following section. It aims to improve the exploration efficiency while maintaining the safety and the input-to-state stability of the closed-loop system, even in the presence of the probing noise.

3.2 | Safe Exploration

The definition of ISS-CLF and R-CBF are provided first with respect to nonlinear system (36) with an external disturbance w. These concepts are used to further address the case where probing noise is added.

$$\dot{x} = f(x) + g(x)u + p(x)w \tag{36}$$

where $w \in \mathbb{R}^q$, $p : \mathbb{R}^n \to \mathbb{R}^{n\times q}$. The system (36) is input-to-state stabilizable if and only if there exists an ISS-CLF. The definition of ISS-CLF was introduced in[36].

Definition 4. A positive definite, radially unbounded function V is an ISS-CLF if there exists class κ_∞ functions α, η such that, for $\|x\| \geq \eta(\|w\|)$,

$$\inf_{u\in\mathscr{U}} \left[L_f V(x) + L_g V(x) + \left\|L_p V(x)\right\| \eta^{-1}(\|x\|) + \alpha(\|x\|)\right] \leq 0 \tag{37}$$

A function $B_\gamma(x)$ is an R-CBF[37] with respect to $\mathscr{C}$ if it satisfies the conditions described in (38).

Definition 5. A function $B_\gamma(x)$ is an R-CBF with respect to the set $\mathscr{C}$ if $B_\gamma(x)$ is positive for $x \in int\mathscr{C}$, $B_\gamma(x) \to \infty$ as $x \to \partial\mathscr{C}$, and there exists a class κ function α_B such that

$$\inf_{u\in\mathscr{U}} \left[L_f B_\gamma(x) + L_g B_\gamma(x) + \left\|L_p B_\gamma(x)\right\| w - \alpha_B(h(x))\right] \leq 0 \tag{38}$$

By combining the concepts of ISS-CLF and R-CBF, a control policy can be designed to maintain stability and guarantee system safety. This approach allows for careful and marginal adjustments to the policy while ensuring that the system operates within predefined safe boundaries, thereby minimizing potential risks. Furthermore, for systems affected by probing noise e:

$$\dot{x} = f(x) + g(x)(u + e) \tag{39}$$

The definitions 4 and 5 can be adapted by replacing w with e and $p(x)$ with $g(x)$. In this context, e is added to the feedback input during the learning process to encourage exploration of the state space and prevent the agent from getting stuck in a suboptimal solution. With this setup, consider the following constrained stabilization problem for the system (39) impacted by input noise.
Robust-QP Problem : Find the control u_{safe} and the relaxation variable δ that satisfy

$$\begin{aligned} \min_{u_{safe},\delta} \quad & \frac{1}{2}(u_{safe}^T u_{safe} + \ell\delta^T\delta) \\ s.t. \quad & F_1 = a_1 + b_1(u + u_{safe}) + \delta \leq 0 \\ & F_2 = a_2 + b_2(u + u_{safe}) \leq 0 \end{aligned} \tag{40}$$

with

$$\begin{aligned} a_1 &= L_f V(x) + \left\|L_g V(x)\right\| \eta^{-1}(\|x\|) + \alpha(\|x\|) \\ a_2 &= L_f B_\gamma(x) + \left\|L_g B_\gamma(x)\right\| e(t) - \alpha_B(h(x)) \end{aligned}$$

$$b_1 = L_g V(x)$$
$$b_2 = L_g B_\gamma(x)$$

Assumption 3. The gradients of the R-CBF B_γ and ISS-CLF V, α, α_B, η^{-1} are assumed to be Lipschitz continuous.

The solution of the Robust-QP problem (see appendix 6.2) described above plays a crucial role in ensuring the safety and admissibility of the policy within the context of the off-policy algorithm, as discussed in the following section.

3.3 | Safe Off-Policy Algorithm

In the safe off-policy algorithm, during the exploration phase, data collection is performed using an initial control policy that must satisfy two critical properties: safety and admissibility. These properties ensure that the policy remains safe and valid even in the presence of probing noise. By making minimal adjustments to the policy using the solution derived from the Robust-QP formulation, the resulting policy can effectively maintain its safety and admissibility. This approach enables successful exploration while upholding the necessary safety constraints throughout the learning process. Now, consider the system

$$\dot{x} = f(x) + g(x)[u_0 + e + u_{safe}] \tag{41}$$

Then, (41) can be rewritten as

$$\dot{x} = f(x) + g(x)u_i + g(x)\nu_i \tag{42}$$

where $\nu_i = u_0 + e + u_{safe} - u_i = u_s - u_i$ and $u_{noisy} = u_0 + e$.

Remark 1. u_{noisy} refers to the policy after adding the probing noise e. The purpose of adding e is indeed to excite the system states and collect rich data during the exploration phase. Examples of such noise are random noises[38], sinusoidal signals[39], and decayed signals[40]. However, it's important to note that adding such noise can introduce a risk of violating safety boundaries and potentially destabilizing the system. To address this problem, the solution of the Robust-QP problem u_{safe} is added to u_{noise} giving $u_s = u_{noisy} + u_{safe}$.

For all $i \geq 0$, the time derivative of $W_i(x)$ along the solutions of (42) is given by

$$\dot{W}_i = \nabla W_i^T(x)[f(x) + g(x)u_i + g(x)\nu_i] = -q(x) - u_i^T R u_i - B_\gamma(x) - 2u_{i+1}^T R\nu_i \tag{43}$$

By integrating both sides of (43) over any time interval $[t, t+T]$, it gives

$$W_i(x(t+T)) - W_i(x(t)) = -\int_t^{t+T} [q(x) + u_i^T R u_i + B_\gamma(x) + 2u_{i+1}^T R\nu_i]dt \tag{44}$$

Let Ω be a compact containing the origin as an interior point, the value function W_i (referred to as the critic) and the control policy u_{i+1} (referred to as the actor) can be approximated by using the basis function representation:

$$\hat{W}_i(x) = \hat{C}_i \Phi(x) \tag{45}$$

$$\hat{u}_{i+1}(x) = \hat{U}_i \Psi(x) \tag{46}$$

with $\Phi = [\phi_1, \phi_2 \dots \phi_{N_1}]^T$ and $\Psi = [\psi_1, \psi_2 \dots \psi_{N_2}]^T$, are two finite vectors of linearly independent smooth basis functions on Ω. $\hat{C}_i \in \mathbb{R}^{1\times N_1}$ and $\hat{U}_i \in \mathbb{R}^{m\times N_2}$ are the weights matrices to be determined.

Lemma 2. *The weights $\hat{C}_i$ and $\hat{U}_i$ can be obtained by solving the following least-squares (LS) equation:*

$$\tilde{\Theta}_i^N \begin{bmatrix} vec(\hat{C}_i) \\ vec(\hat{U}_i^T) \end{bmatrix} = \tilde{E}_i^N \tag{47}$$

for $N > N_1 + mN_2$ and

$$\begin{aligned} \tilde{\Theta}_i^N &= [\tilde{\Theta}_i(t_1), \dots, \tilde{\Theta}_i(t_N)]^T \\ \tilde{E}_i^N &= [\tilde{E}_i(t_1), \dots, \tilde{E}_i(t_N)]^T \end{aligned} \tag{48}$$

where

$$\tilde{\Theta}_i(t) = \begin{bmatrix} [\Phi(x(t+T)) - \Phi(x(t))]^T \\ 2[I_{u\Psi}(R \otimes I_{N_2}) - I_{\Psi\Psi}(\hat{U}_{i-1}^T R \otimes I_{N_2})] \end{bmatrix}^T \tag{49}$$

$$\tilde{E}_i(t) = -I_{\Psi\Psi}[\hat{U}_{i-1}^T \otimes \hat{U}_{i-1}^T]vec(R) - \int_t^{t+T} [q(x) + B_\gamma(x)]dt \tag{50}$$

Proof. Replacing W_i, u_i, and u_{i+1} in (44) with their approximations (45) and (46) gives

$$\hat{C}_i[\Phi(x(t)) - \Phi(x(t+T))] = -\int_t^{t+T} 2\Psi^T(x)\hat{U}_i^T R\hat{\nu}_i dt - \int_t^{t+T} [q(x) + \hat{u}_i^T R\hat{u}_i + B_\gamma(x)]dt \tag{51}$$

where $\hat{u}_0 = u_0$, $\hat{\nu}_i = u - \hat{u}_i$. Thus,

$$\int_t^{t+T} \Psi^T(x)\hat{U}_i^T R\hat{\nu}_i dt = \int_t^{t+T} \Psi^T(x)\hat{U}_i^T R[u_s - \hat{u}_i]dt \tag{52}$$

The following equations can be derived:

$$\begin{aligned} \int_t^{t+T} \Psi^T(x)\hat{U}_i^T R u_s dt &= \int_t^{t+T} [u_s^T R \otimes \Psi^T(x)]vec(\hat{U}_i^T)dt \\ &= \int_t^{t+T} [u_s^T \otimes \Psi^T(x)][R^T \otimes I_{N_2}]vec(\hat{U}_i^T)dt \\ &= I_{u\Psi}[R \otimes I_{N_2}]vec(\hat{U}_i^T) \end{aligned} \tag{53}$$

$$\begin{aligned} \int_t^{t+T} \Psi^T(x)\hat{U}_i^T R\hat{u}_i dt &= \int_t^{t+T} \Psi^T(x)\hat{U}_i^T R\hat{U}_{i-1}\Psi(x)dt \\ &= \int_t^{t+T} [\Psi^T(x)\hat{U}_{i-1}^T R \otimes \Psi^T(x)]vec(\hat{U}_i^T)dt \\ &= \int_t^{t+T} [\Psi^T(x) \otimes \Psi^T(x)]dt[\hat{U}_{i-1}^T R \otimes I_{N_2}]vec(\hat{U}_i^T) \\ &= I_{\Psi\Psi}[\hat{U}_{i-1}^T R \otimes I_{N_2}]vec(\hat{U}_i^T) \end{aligned} \tag{54}$$

By substituting (53) and (54) in (52), we obtain

$$\int_{t_k}^{t_{k+1}} \Psi^T(x)\hat{U}_i^T R\hat{\nu}_i dt = I_{u\Psi}[R \otimes I_{N_2}]vec(\hat{U}_i^T) - I_{\Psi\Psi}[\hat{U}_{i-1}^T R \otimes I_{N_2}]vec(\hat{U}_i^T) \tag{55}$$

Moreover, we have

$$\int_t^{t+T} \hat{u}_i^T R\hat{u}_i dt = \int_t^{t+T} \Psi^T(x)\hat{U}_{i-1}^T R\hat{U}_{i-1}\Psi(x)dt = I_{\Psi\Psi}[\hat{U}_{i-1}^T \otimes \hat{U}_{i-1}^T]vec(R) \tag{56}$$

Finally, by replacing (55) and (56) in (51), we get

$$\begin{aligned} &[\Phi(x(t+T)) - \Phi(x(t))]^T \hat{C}_i^T + 2[I_{u\Psi}(R \otimes I_{N_2}) - I_{\Psi\Psi}(\hat{U}_{i-1}^T R \otimes I_{N_2})]vec(\hat{U}_i^T) \\ &\qquad = -I_{\Psi\Psi}[\hat{U}_{i-1}^T \otimes \hat{U}_{i-1}^T]vec(R) - \int_t^{t+T} [q(x) + B_\gamma(x)]dt \end{aligned} \tag{57}$$

(57) is rewritten in regression form as

$$\tilde{\Theta}_i(t) \begin{bmatrix} vec(\hat{C}_i) \\ vec(\hat{U}_i^T) \end{bmatrix} = \tilde{E}_i(t) \tag{58}$$

with

$$\tilde{E}_i(t) = -I_{\Psi\Psi}[\hat{U}_{i-1}^T \otimes \hat{U}_{i-1}^T]vec(R) - \int_t^{t+T} [q(x) + B_\gamma(x)]dt \tag{59}$$

$$\tilde{\Theta}_i(t) = \begin{bmatrix} [\Phi(x(t+T)) - \Phi(x(t))]^T \\ 2[I_{u\Psi}(R \otimes I_{N_2}) - I_{\Psi\Psi}(\hat{U}_{i-1}^T R \otimes I_{N_2})] \end{bmatrix}^T \tag{60}$$

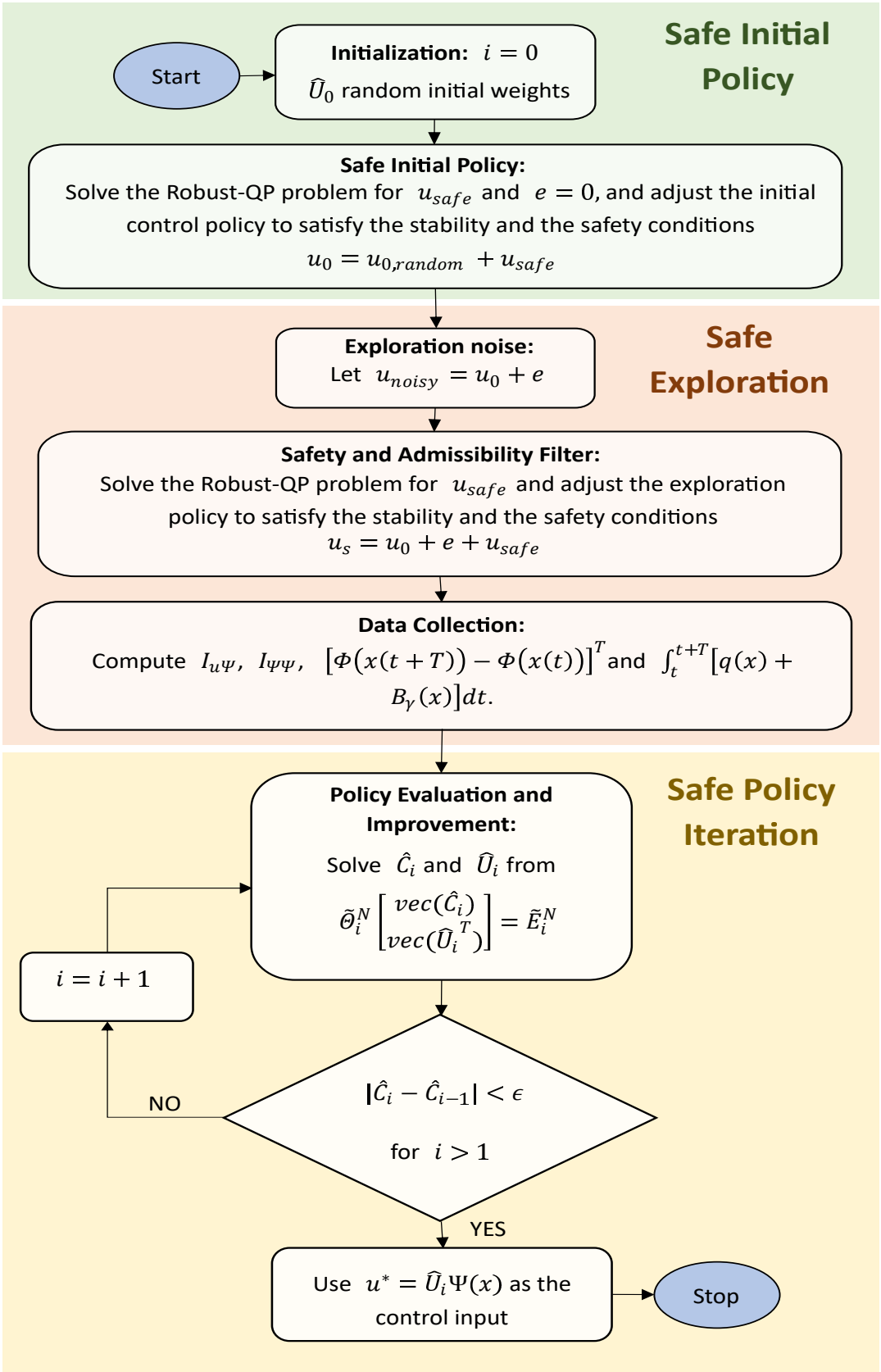

FIGURE 3 Flowchart of Safe Off-Policy algorithm.

(58) includes $N_1 + mN_2$ unknown parameters that can be estimated using Least-Square (LS) method. But first it is important to collect enough state and input data to ensure that there is a sufficient number of equations to solve for these unknown parameters. Let the collected information be saved in matrices $\tilde{\Theta}_i^N$ and $\tilde{E}_i^N$ as

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

Hence, the LS equation becomes

$$\tilde{\Theta}_i^N \begin{bmatrix} vec(\hat{C}_i) \\ vec(\hat{U}_i^T) \end{bmatrix} = \tilde{E}_i^N \tag{61}$$

with $N > N_1 + mN_2$. □

The flowchart, presented in Fig. 3, illustrates the structure of the developed algorithm and highlights its three distinct phases. In the first phase, referred to as policy initialization, a random initial policy $u_{0,random}$ is generated and then marginally modified using the output of the Robust-QP problem u_{safe} for a null noise. This adjustment ensures that the initial policy u_0 satisfies the conditions of safety and admissibility, where $u_0 = u_{0,random} + u_{safe}$. In the second phase, known as the exploration phase, the initial policy is enriched by adding probing noise, allowing to excite the system and to collect diverse and valuable data. Additionally, the Robust-QP problem is employed in this phase to ensure that the safety constraints are satisfied and to guarantee that the system is ISS when e, the exploration noise, is considered as an input. Once the data collection is completed, the algorithm proceeds to the third phase, where the safe-PI algorithm is iteratively computed. This iterative process continues until the weights of the value function converge, leading to the determination of an optimal and safe policy.

To evaluate the effectiveness of the developed algorithm, it was applied to two examples. The first example involved a jet engine surge and stall dynamics, while the second example focuses on controlling a MIMO unstable nonlinear system.

4 | SIMULATIONS AND RESULTS

4.1 | Jet engine dynamics

Consider the following jet engine surge and stall dynamics [30]

$$\begin{aligned} \dot{x}_1 &= -0.35x_1^2 - 0.35x_1(2x_2 + x_2^2) \\ \dot{x}_2 &= -1.4x_2^2 - 0.5x_2^3 - (u + 3x_1x_2 + 3x_1) \end{aligned} \tag{62}$$

where x_1 is rotating stall amplitude which is normalised, x_2 is the deviation of annulus-averaged flow, u is the deviation of the plenum pressure rise and is considered as the control input. In this example, the QP problem will consider only the R-CBF condition. This is due to the fact that the system under consideration is already stable, allowing us to focus exclusively on addressing the state constraints.
The safe set in which the states of the system should belong to is defined by $\mathscr{C} = \{x | \quad -1.1 < x_2 < 0.45\}$.
To guarantee the forward invariance of this set and to ensure the safety of the system, the following CBFs are defined.

$$B_{1,\gamma}(x) = -\frac{\gamma_1 h_1(x)}{\gamma_1 h_1(x) + 1}, \quad B_{2,\gamma}(x) = -\frac{\gamma_2 h_2(x)}{\gamma_2 h_2(x) + 1} \tag{63}$$

with $h_1(x) = -x_2^{min} + x_2$ and $h_2(x) = x_2^{max} - x_2$. The modified formulation (19) is used with the following reward function

$$r_{safe}(x, u) = x^T Q x + u^T R u + B_{1,\gamma}(x) + B_{2,\gamma}(x) \tag{64}$$

where Q, R, γ_1, γ_2 are design parameters, with the following values assigned: $Q = diag(20, 10)$, $R = 0.3 \times I_{2\times 2}$, $\gamma_1 = 0.7$ and $\gamma_2 = 0.2$.
From t = 0s to t = 6.35s, an exploration noise $e(t)$ is injected into the initial policy, with $e(t)$ being specifically set to

$$e(t) = \sum 2 \times \omega \times sin(\vartheta_1 t) \tag{65}$$

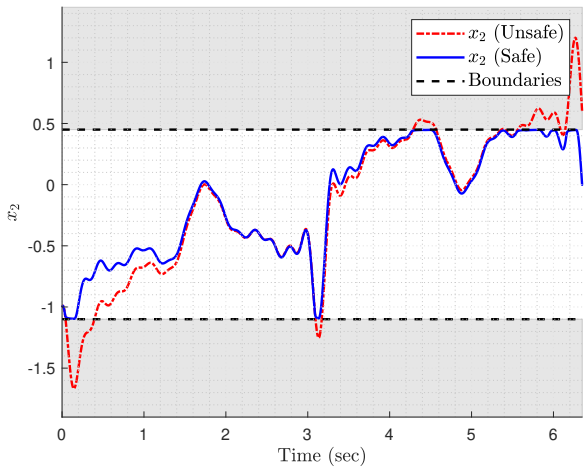

FIGURE 4 Trajectory of x_2 during exploration.

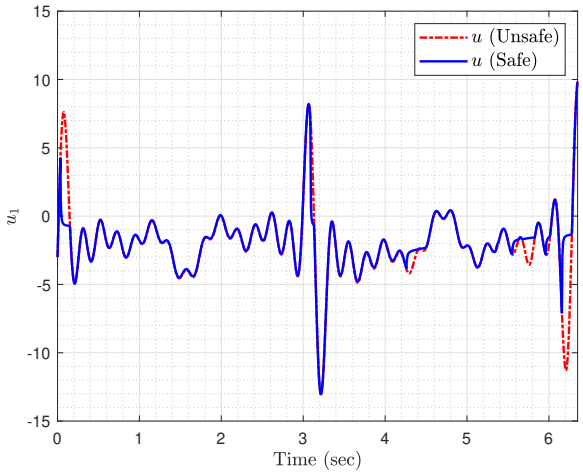

FIGURE 5 Exploration policy under probing noise.

with $\vartheta_1 = [1, 3, 7, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]$ and ω a random variable with a Gaussian distribution. The activation functions are considered, respectively, as

$\Phi(x) = [x_1^2, x_2^2, x_1x_2, x_1^4, x_2^4, x_1^3x2, x_1^2x_2^2, x_1x_2^3, x_1^6, x_2^6, x_1^5x_2, x_1^4x_2^2, x_1^3x_2^3, x_1^2x_2^4, x_1x_2^5, x_1^8, x_2^8, x_1^7x_2, x_1^6x_2^2, x_1^5x_2^3, x_1^4x_2^4, x_1^3x_2^5, x_1^2x_2^6, x_1x_2^7]^T$

$\Psi(x) = \left[x_1, x_2, x_1x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1^2x_2^3, x_2^2x_1^3, x_1^4, x_2^4\right]^T$.

These weights of these networks are trained by finding the solution of (61) for $N = 635$. The state and input data are collected over each interval of $T = 0.01s$. In this example, the initial policy was not randomly set. Instead, a specific value was chosen for the policy weights, which is $\hat{U}_0 = [-3 \quad zeros(1, 10)]$.

R-CBF criteria is formed accordingly based on (40) as

$$\begin{aligned} L_fB_{1,\gamma}(x) + \left\|L_gB_{1,\gamma}(x)\right\| e(t) - \alpha_{1,B}(h_1(x)) + L_gB_{1,\gamma}(x)(u + u_{safe}) \leq 0 \\ L_fB_{2,\gamma}(x) + \left\|L_gB_{2,\gamma}(x)\right\| e(t) - \alpha_{2,B}(h_2(x)) + L_gB_{2,\gamma}(x)(u + u_{safe}) \leq 0 \end{aligned} \tag{66}$$

with $\alpha_{1,B} = 2000 \times h_1(x)$, $\alpha_{2,B} = 2000 \times h_2(x)$. These values are chosen high to assure the collection of data not only from the safe region but also from the vicinity of the safety boundary. Moreover, in the Robust-QP problem, different values are assigned to γ_1 and γ_2 such as $\gamma_1 = 15$ and $\gamma_2 = 20$. Initially, these values were chosen smaller in order to enhance the penalty imposed by the barrier function in the cost function.

The trajectory of the state x_2, during the exploration phase, is shown in Fig.4, where the safety boundaries are plotted with dashed black lines. In order to ensure a safe performance, it is necessary for the system trajectory to remain within these two lines. The blue curve represents the evolution of x_2 when the Robust-QP problem is activated to guarantee safety. In this case, it is evident that the safety and stability of the system are maintained during the exploration where noisy input (Fig. 5) is introduced to the system. This finding confirms that it is unnecessary to include the ISS-CLF condition in this example. However, when the Robust-QP problem is deactivated, it can be observed from the red curve that the state x_2 violates the safety boundaries. Fig. 5 displays the noisy input used for exploration purposes. It can be seen that, in order to ensure safety, the unsafe policy (red curve) is slightly adjusted, leading to a safe policy (blue curve).

After data collection, the safe-PI algorithm is iteratively computed until the convergence of the critic weights is reached. Fig. 6 shows that after 17 iterations, the algorithm has converged. Moreover, for the initial state $x_0 = [1, -1]^T$, the convergence process of the value function is displayed in Fig. 7. This latter confirms the first point of Theorem 1, which states that the value function exhibits a monotonic decreasing behavior.

Once the safe-PI algorithm has reached convergence, the trained actor is used to control the system. Fig. 8 shows the trajectory of the state variable x_2, starting from the initial state and not the state where exploration was interrupted in order to demonstrate the safety guarantees provided by the learned policy. The figure display two curves:

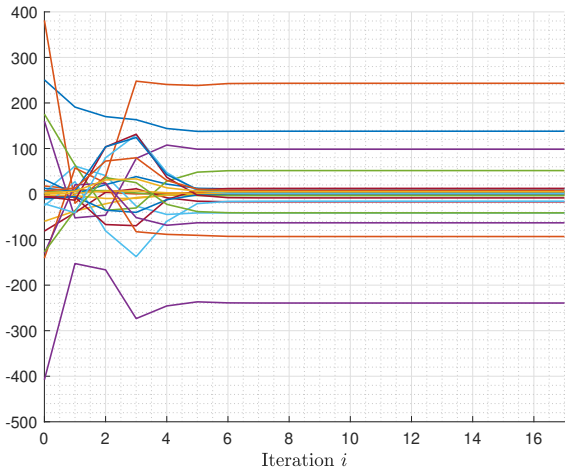

FIGURE 6 Convergence of the Critic weights.

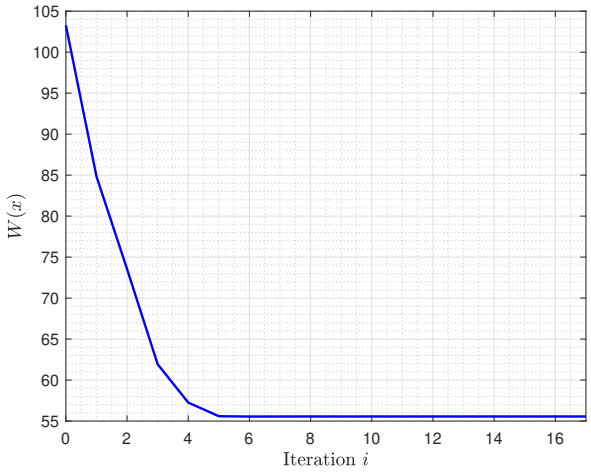

FIGURE 7 Convergence process of the value function at $x = [1, -1]^T$.

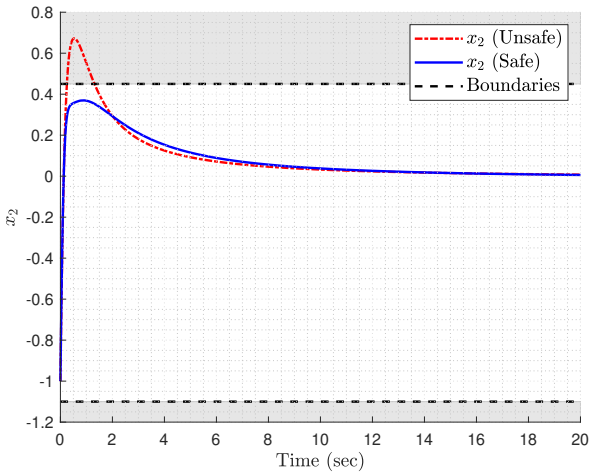

FIGURE 8 Trajectory of x_2 for safe and unsafe learned policy.

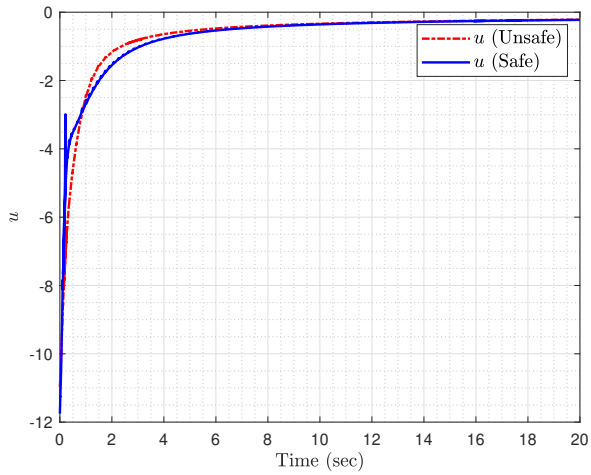

FIGURE 9 Safe and unsafe learned policy.

- the red curve represents the trajectory of x_2 for classical off-policy algorithm where no safety guarantees are considered during the exploration and the exploitation;
- the blue curve represents the trajectory of x_2 under the learned policy where the exploration was done in a safe manner and the reward function was augmented with CBFs.

It can be seen that, under the proposed algorithm, x_2 remains within the safe region. In contrast, when using a classical off-policy algorithm, x_2 would violate the safety boundaries. Moreover, the control input behaviour is shown in Fig. 9 for the two cases. These results were obtained for the first example, demonstrating the effectiveness of the developed algorithm. In the following, the results for a nonlinear unstable MIMO system will be presented. In this next example, the initial policy will be generated randomly and the ISS-CLF will be integrated into the Robust-QP problem.

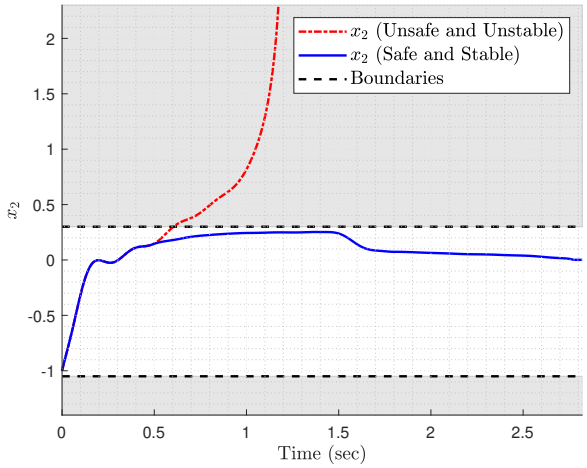

FIGURE 10 Trajectory of x_2 during exploration.

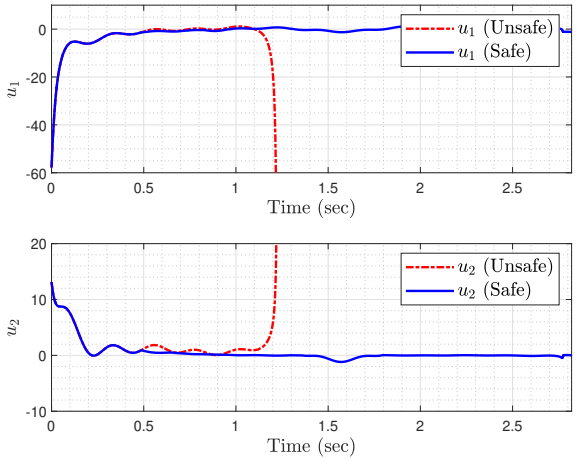

FIGURE 11 Exploration policy under probing noise.

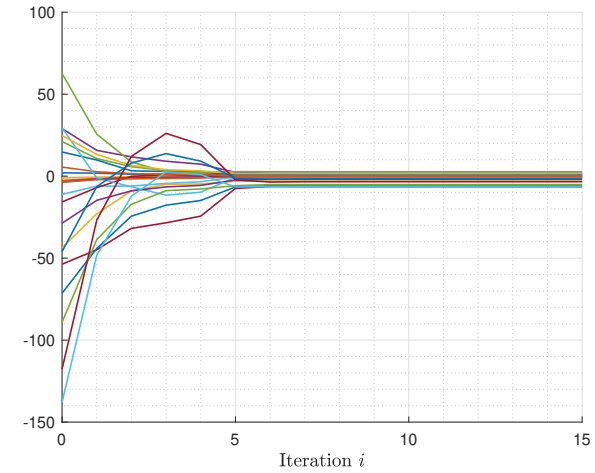

FIGURE 12 Convergence of the Critic weights.

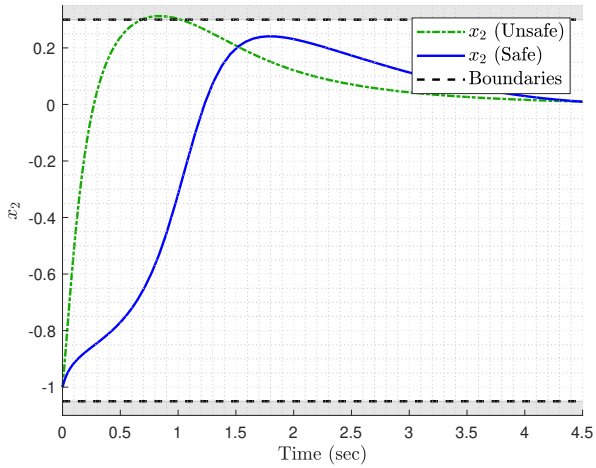

FIGURE 13 Trajectory of x_2 for the safe and unsafe learned policy with safe exploration.

4.2 | Nonlinear MIMO unstable system

Consider a MIMO nonlinear system described by the following differential equations:

$$\begin{aligned} \dot{x}_1 &= x_1^3 - x_2 + 2x_1^2 + u_1 sin(x_1) \\ \dot{x}_2 &= -x_1 + x_2^3 + 3x_2^2 + u_2 cos(x_2) \end{aligned} \tag{67}$$

The safe set is defined by $\mathscr{C} = \{x| \quad -1.05 < x_2 < 0.3\}$. The reward function (64) is used with $Q = 0.35 \times I_{2\times 2}$, $R = 0.1 \times I_{2\times 2}$, $\gamma_1 = 3500$ and $\gamma_2 = 3000$. It's important to note that by assigning a small value to Q, the cost associated with the system's behavior is reduced. However, if the value of γ is also small, the BF term will have a greater influence on the overall cost function. To maintain a balance between the barrier function and the overall cost, higher values for γ_1 and γ_2 are chosen. Moreover, since the upper bound is located close to the system's equilibrium, it is crucial to impose a high penalty only when the system is in close proximity to this bound. By selecting higher values for γ_1 and γ_2, the penalty associated with the BF is increased only when the system approaches the upper bound, thus ensuring strict adherence to safety constraints.

From t = 0s to t = 2.85s, the exploration noise $e(t)$ is injected into the initial policy, with $e(t)$ being set to

$$e(t) = \sum 0.6 \times sin([1, 3, 7, 11, 13, 15, 17, 19, 21, 23, 25, 27]t) \tag{68}$$

The activation functions are considered, respectively, as
$\Phi(x)= [x_1^2, x_2^2, x_1x_2, x_1^4, x_2^4, x_1^3x2, x_1^2x_2^2, x_1x_2^3, x_1^6, x_2^6, x_1^5x_2, x_1^4x_2^2, x_1^3x_2^3, x_1^2x_2^4, x_1x_2^5, x_1^8, x_2^8, x_1^7x_2, x_1^6x_2^2, x_1^5x_2^3, x_1^4x_2^4, x_1^3x_2^5, x_1^2x_2^6, x_1x_2^7]^T$
$\Psi(x) = \left[x_1, x_2, x_1x_2, x_1^2, x_2^2\right]^T$
R-CBF criteria is given in (66) and ISS-CLF condition is given by

$$L_fV(x) + \left\|L_gV(x)\right\| \mathrm{I}e(t)\mathrm{I} + \alpha(\|x\|) + L_gV(x)(u + u_{safe}) + \delta \leq 0 \tag{69}$$

In this example, the initial policy is generated randomly, then it is modified by solving the Robust-QP problem for $e(t) = 0$ in order to satisfy the stability and the safety conditions. During the initialization phase, the hyperparameters of the Robust-QP problem are set to $\gamma_1 = \gamma_2 = 10$, $\alpha_{1,B} = 0.4 \times h_1(x)$, $\alpha_{2,B} = 0.3 \times h_2(x)$, $\alpha = 70 \times \|x\|$, and the Lyapunov function $V(x)$ is chosen as $V(x) = x_1^2 + x_2^2$.
During the exploration phase, the hyperparameters are modified to $\gamma_1 = \gamma_2 = 20$, $\alpha_{1,B} = 10 \times h_1(x)$, $\alpha_{2,B} = 10 \times h_2(x)$, and $\alpha = 0.1 \times \|x\|$. By adjusting these hyperparameters, the algorithm becomes less conservative and allows the collection of more informative and rich data.

The trajectory of the state x_2, during the exploration phase, is shown in Fig.10. The blue curve represents the evolution of x_2 when the Robust-QP problem is activated to guarantee safety and admissibility of the policy. In this case, it is evident that the safety and stability of the system are maintained during the exploration where noisy input is applied to the system. However, when the Robust-QP problem is deactivated, it can be observed from the red curve that the state x_2 violates the safety boundary and the system is destabilized. Fig. 11 displays the evolution of the input for the two cases. It can be seen from the red curve that the policy becomes unsafe and inadmissible once the excitation noise is added. However, when the Robust-QP problem is activated, the opposite behavior is observed, ensuring the safety and admissibility of the policy.
After collecting the data, the safe-PI algorithm is iteratively computed until the convergence of the critic weights is reached. Fig. 12 shows that, after 15 iterations, the algorithm has converged.
Once the safe-PI algorithm has reached convergence, the trained actor is used to control the system. Fig. 13 shows the trajectory of the state variable x_2, starting from the initial state and not the state where exploration was interrupted in order to demonstrate the safety guarantees provided by the learned policy. The figure display two curves:

- the green curve represents the trajectory of x_2 for off-policy algorithm where safety guarantees are considered during the exploration but CBFs are not considered in the reward function;
- the blue curve represents the trajectory of x_2 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 even if the exploration is achieved in a safe manner this does not lead to a safe learned policy. Hence, to address this problem, CBFs were added to the reward function in order to ensure the convergence toward a safe policy.

5 | CONCLUSIONS

Overall, the proposed approach presents a novel solution to the problem of safe control learning in off-policy based approaches for nonlinear systems by introducing guarantees of safety and stability throughout both the exploration and exploitation phases. The approach comprises three main phases: Safe policy initialization, Safe exploration and Safe-Policy Iteration (Safe-PI) computation. During the policy initialization phase, an initial policy is randomly generated and subsequently adjusted to meet safety and admissibility requirements. The exploration phase introduces probing noise, allowing the collection of diverse and informative data. Simultaneously, the algorithm uses the Robust-QP problem to enforce safety constraints and maintain system stability throughout the exploration process. Once the data collection is complete, safe-PI is iteratively computed until convergence to a policy that balances safety, stability and optimality. Simulation results demonstrate the algorithm's effectiveness in generating safe and stable policies, even in the presence of probing noise. Moreover, rigorous mathematical proofs are

Variables	Definitions
$x \in \mathbb{R}^n$	System states
x_u	State of the system evolved by the input u
$u \in \mathbb{R}^m$	Control input
$u_{0,random}$	Random initial policy
u_0	Safe and admissible initial policy
u_{noisy}	Exploration policy
u_s	Safe Exploration policy
$\mathscr{C}$	Safe set of system states
$\mathscr{U}_c$	Set of safe inputs
$\mathscr{U}$	Set of admissible inputs
$\mathscr{U}_a$	Set of admissible and safe inputs
H_{safe}	Safe Hamiltonian function
r_{safe}	Safe reward function
$W(x)$	Safe value function
e	Probing noise
r_{safe}	Safe reward function
$\Phi(x)$	Critic basis function
$\Psi(x)$	Actor basis function
$\hat{C}$	Critic weights
$\hat{U}$	Actor weights
B_γ	Control barrier function
α_B	Class κ function associated to the control barrier function condition
α	Class κ function associated to the ISS-control Lyapunov function condition

TABLE 1 Table of notations

provided to establish the stability and safety guarantees of the algorithm.
While the proposed algorithm has demonstrated promising results in ensuring safety and stability, one aspect to consider is the reliance on the system model to solve the Robust-QP problem. To address this limitation, future work will focus on assuming only a nominal model is available, and the uncertainty of the model will be approximated using machine learning techniques like neural networks or Gaussian processes.

6 | APPENDIX

6.1 | Proof of Theorem 1

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

Lemma 3. *Under Assumption 2, the following holds:*

1. $$W^*(x) \leq W_i(x); \tag{70}$$
2. *for any $W_{i-1} \in \mathcal{P}$, satisfying*
$$\nabla W_{i-1}^T[f(x) + g(x)u_i] + q(x) + u_i^T R u_i + B_\gamma(x) \leq 0 \tag{71}$$
it follows that $W_i \leq W_{i-1}$;
3. $$\nabla W_i^T[f(x) + g(x)u_{i+1}] + q(x) + u_{i+1}^T R u_{i+1} + B_\gamma(x) \leq 0. \tag{72}$$

Proof. 1. First, we want to prove that $W_i(x) \geq W^*(x)$. Under Assumption 2, we have

$$\nabla W^{*T}(x)[f(x) + g(x)u^*] + r(x, u^*) = 0 \tag{73}$$

It follows that

$$\begin{aligned}&[\nabla W_i(x)-\nabla W^*(x)]^T[f(x)+g(x)u_i]+\nabla W^{*T}(x)g(x)[u_i-u^*]+u_i^TRu_i-u^{*T}Ru^*\\&=[\nabla W_i(x)-\nabla W^*(x)]^T[f(x)+g(x)u_i]-2u^{*T}R[u_i-u^*]+u_i^TRu_i-u^{*T}Ru^*\\&=[\nabla W_i(x)-\nabla W^*(x)]^T[f(x)+g(x)u_i]+(u^*-u_i)^TR(u^*-u_i)\\&=0\end{aligned}\tag{74}$$

Hence, $\forall\, x_0\in\mathscr{C}$, along the trajectories of system (1) with $u=u_i$ and $x(0)=x_0$, the following holds:

$$W_i(x_0)-W^*(x_0)=\int_0^\infty (u^*-u_i)^TR(u^*-u_i)dt\geq 0\tag{75}$$

which implies that $W_i(x)\geq W^*(x)$, $\forall x\in\mathscr{C}$.

2. We have

$$\nabla W_{i-1}^T(x)[f(x)+g(x)u_i]+q(x)+u_i^TRu_i+B_\gamma(x)\leq 0\tag{76}$$

Let $m(x)\geq 0$, such that

$$\nabla W_{i-1}^T(x)[f(x)+g(x)u_i]+q(x)+u_i^TRu_i+B_\gamma(x)=-m(x)\tag{77}$$

Since $\nabla W_i^T(x)[f(x)+g(x)u_i]+r_{safe}(x,u_i)=0$, it follows

$$[\nabla W_{i-1}(x)-\nabla W_i(x)]^T[f(x)-g(x)u_i]=-m(x)\tag{78}$$

Hence, $\forall\, x_0\in\mathscr{C}$, along the trajectories of system (1) with $u=u_i$ and $x(0)=x_0$, the following holds:

$$W_i(x_0)-W_{i-1}(x_0)=-\int_0^\infty m(x)<0\tag{79}$$

Thus $W_i(x)<W_{i-1}(x)$, $\forall x\in\mathscr{C}$.

3. The goal is to show that

$$\nabla W_i^T(x)[f(x)+g(x)u_{i+1}]+q(x)+u_{i+1}^TRu_{i+1}+B_\gamma(x)\leq 0\tag{80}$$

By definition

$$\begin{aligned}&\nabla W_i^T[f(x)+g(x)u_{i+1}]+q(x)+u_{i+1}^TRu_{i+1}+B_\gamma(x)\\&=\nabla W_i^T(x)[f(x)+g(x)u_{i+1}]+q(x)+u_{i+1}^TRu_{i+1}+B_\gamma(x)+\nabla W_i^Tg(x)u_i-\nabla W_i^Tg(x)u_i+u_i^TRu_i-u_i^TRu_i\\&=\nabla W_i^T[f(x)+g(x)u_i]+q(x)+u_i^TRu_i+B_\gamma(x)+\nabla W_i^Tg(x)[u_{i+1}-u_i]+u_{i+1}^TRu_{i+1}-u_i^TRu_i\\&=\nabla W_i^Tg(x)[u_{i+1}-u_i]+u_{i+1}^TRu_{i+1}-u_i^TRu_i\\&=-2u_{i+1}^TR[u_{i+1}-u_i]+u_{i+1}^TRu_{i+1}-u_i^TRu_i\\&=-[u_{i+1}-u_i]^TR[u_{i+1}-u_i]\leq 0\end{aligned}\tag{81}$$

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, moreover it is proved that $W_i\in\mathcal{P}$, for all $i=0,1,\ldots$

(a) For $i=1$, it follows from Assumption 1, Lemma 3.1 and Lemma 3.2 that Theorem 1.1 and Theorem 1.2 are true. Under Assumption 1 and 2, $W^*\in\mathcal{P}$ and $W_0\in\mathcal{P}$ thus $W_1\in\mathcal{P}$.

(b) Suppose Theorem 1.1 and Theorem 1.2 hold for $i=j>1$ and $W_j\in\mathcal{P}$, We want to show that Theorem 1.1 and Theorem 1.2 also hold for $i=j+1$ and $W_{j+1}\in\mathcal{P}$.

Since $W^*\in\mathcal{P}$ and $W_j\in\mathcal{P}$, we deduce that $W_{j+1}\in\mathcal{P}$.

By Lemma 3.3, one obtains

$$\nabla W_{j+1}^T(x)[f(x)+g(x)u_{i+2}]+q(x)+u_{i+2}^TRu_{i+2}+B_\gamma(x)\leq 0\tag{82}$$

Along the solutions of system (1) for $u = u_{j+2}$, one obtains $\dot{W}_{j+1} \leq 0$. Since $W_{j+1} \in \mathcal{P}$, it is a well-defined Lyapunov function for the closed-loop system (1) with $u = u_{j+2}$. Therefore, u_{j+2} is a stabilizing policy which implies that Theorem 1.2 holds for $i = j + 1$.

From Lemma 3.2, we have $W_{j+2} \leq W_{j+1}$ and by induction assumption we have $W^*(x) \leq W_{j+1}(x) \leq W_j(x)$, which gives

$$W^*(x) \leq W_{j+2}(x) \leq W_{j+1}(x)$$

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

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

Now, it must be shown that at each iteration u_i is safe, thus we want to show that the states under policy u_i remain in the safe set $\mathscr{C}$. Earlier, we have proved that $W^*(x) \leq W_{i+1}(x) \leq W_i(x) \leq W_0$, which implies that at each iteration W_i is bounded and consequently the reward $r_{safe}(x, u_i)$ and the barrier function B_γ remains bounded after each policy improvement step. Moreover, B_γ tends to infinity near the boundary of the safe set, implying that the system states doesn't cross $\partial\mathscr{C}$. This in turn guarantees safety and prove that $u_i \in \mathscr{U}_c$. □

6.2 | Solution of Robust-QP problem

The Lagrangian $\mathscr{L}(x, u, \delta, \lambda_1, \lambda_2)$ for the Robust-QP problem is given by

$$\mathscr{L} = \frac{1}{2}(\left\|u_{safe}\right\|^2 + \ell\|\delta\|^2) + \lambda_1(a_1 + b_1(u + u_{safe}) + \delta) + \lambda_2(a_2 + b_2(u + u_{safe}))$$

with λ_1 and λ_2 are scalar Lagrange multipliers. By applying the Karush-Kuhn-Tucker (KKT) conditions, the optimality of the solution can be determined.The solution is optimal if and only if $\frac{\partial\mathscr{L}}{\partial u_{safe}}$ and $\frac{\partial\mathscr{L}}{\partial\delta}$ are equal to 0, $\lambda_i \geq 0$, $F_i \leq 0$, and $\lambda_i F_i = 0$ for $i = 1, 2$:

$$\begin{aligned} \frac{\partial\mathscr{L}}{\partial u_{safe}} &= u_{safe}^T + \lambda_1 b_1 + \lambda_2 b_2 = 0 \\ \frac{\partial\mathscr{L}}{\partial\delta} &= \ell\delta^T + \lambda_1 = 0 \\ \lambda_1 F_1 &= \lambda_1[a_1 + b_1(u + u_{safe}) + \delta] = 0 \\ \lambda_2 F_2 &= \lambda_2[a_2 + b_2(u + u_{safe}) + \delta] = 0 \end{aligned} \tag{84}$$

The four cases will be examined and studied based on the active constraints.

Case 1 ($F_1 < 0$ or $x = 0$, $F_2 < 0$, $\lambda_1 = 0$, $\lambda_2 = 0$): In this case, both constraints are inactive, The solutions to the first two equations in (84) yield to:

$$\begin{aligned} u_{safe} &= 0 \\ \delta &= 0 \end{aligned} \tag{85}$$

It is reasonable and consistent that in this particular scenario, where both conditions are already satisfied, there is no necessity to adjust the input.

Case 2 ($F_1 = 0$, $F_2 < 0$, $\lambda_1 \geq 0$, $\lambda_2 = 0$): In this case, the barrier constraint is inactive and the solution to (84) is given by:

$$\begin{aligned} u_{safe} &= \frac{-\ell b_1^T(a_1 + b_1 u)}{\ell\|b_1\|^2 + 1} \\ \delta &= \frac{-a_1 - b_1 u}{\ell\|b_1\|^2 + 1} \end{aligned} \tag{86}$$

However, it is important to emphasize that even when the barrier constraint is not active, the control law does not provide a guarantee that $\dot{V}(x) < 0$, where $\dot{V}$ is given by:

$$\dot{V} = L_f V(x) + L_g V(x)(u + e + u_{safe}). \tag{87}$$

This is because the slack variable δ, which helps satisfy the constraints, is a fictitious quantity. In this case, if m is very small, the expression for $\dot{V}$ reduces to $\dot{V} = L_f V(x) + L_g V(x)(u + e)$, indicating that the controller in (86) may not be able to stabilize the

system. On the other hand, for m very large, $\dot{V}$ takes the form:

$$\dot{V} = L_g V(x)e - \left\| L_g V(x) \right\| \eta^{-1}(\|x\|) - \alpha(\|x\|)$$

It can be observed that the closed-loop system exhibits input-to-state stability with respect to the noise e when $\|x\| \geq \eta(\|e\|)$. Hence, it is important to note that the selection of an appropriate value for the parameter m is crucial. The choice of m can significantly impact the stability of the closed-loop system.

Case 3 ($F_1 < 0$, $F_2 = 0$, $\lambda_1 = 0$, $\lambda_2 \geq 0$):In this case, the ISS constraint is inactive, and the solution to (84) is as follows:

$$\begin{aligned} u_{safe} &= \frac{-b_2^T(a_2 + b_2 u)}{\|b_2\|^2} \\ \delta &= 0 \end{aligned} \tag{88}$$

It is observed that the slack variable δ is null. This means that there is no need for an additional term to satisfy the ISS constraint since the system already satisfies it without any modifications. However, it is important to note that the existence of this solution depends on the condition that $L_g B(x) \neq 0$. This condition ensures that the control input can be properly determined based on the system dynamics and constraints, leading to a well-defined solution.

Case 4 ($F_1 = 0$, $F_2 = 0$, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$): Both constraints are activated and the solution takes the form:

$$\begin{aligned} u_{safe} &= \frac{-b_2^T(a_2 - b_2 u)}{\|b_2\|^2} \\ \delta &= \frac{-a_1\|b_1\|^2 - b_1 u\|b_1\|^2 - b_2^T(a_2 - b_2 u)}{\|b_1\|^2} \end{aligned} \tag{89}$$

Here, b_1 and b_2 must be different from 0 so that the solution can be well-defined.

REFERENCES

1. Knight JC. Safety critical systems: challenges and directions. In: *Proceedings of the 24th international conference on software engineering.* 2002:547–550.
2. Alleyne A, Allgöwer F, Ames A, et al. Control for Societal-scale Challenges: Road Map 2030. In: *2022 IEEE CSS Workshop on Control for Societal-Scale Challenges.*IEEE Control Systems Society. 2023.
3. Zhou K, Doyle JC. *Essentials of robust control*. 104. Prentice hall Upper Saddle River, NJ, 1998.
4. Zhang S, Zhai DH, Xiong Y, Lin J, Xia Y. Safety-critical control for robotic systems with uncertain model via control barrier function. *International Journal of Robust and Nonlinear Control.* 2023;33(6):3661–3676.
5. Brunke L, Greeff M, Hall AW, et al. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems.* 2022;5:411–444.
6. Lewis FL, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine.* 2009;9(3):32-50. doi: 10.1109/MCAS.2009.933854
7. Jha MS, Theilliol D, Weber P. Model-free optimal tracking over finite horizon using adaptive dynamic programming. *Optimal Control Applications and Methods.*
8. Yang Y, Modares H, Vamvoudakis KG, Lewis FL. Cooperative Finitely Excited Learning for Dynamical Games. *IEEE Transactions on Cybernetics.* 2023.
9. Buşoniu L, Bruin dT, Tolić D, Kober J, Palunko I. Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control.* 2018;46:8–28.
10. Garcıa J, Fernández F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research.* 2015;16(1):1437–1480.
11. Yang Y, Vamvoudakis KG, Modares H. Safe reinforcement learning for dynamical games. *International Journal of Robust and Nonlinear Control.* 2020;30(9):3706–3726.
12. Tamar A, Xu H, Mannor S. Scaling up robust MDPs by reinforcement learning. *arXiv preprint arXiv:1306.6189.* 2013.
13. Basu A, Bhattacharyya T, Borkar VS. A learning algorithm for risk-sensitive cost. *Mathematics of operations research.* 2008;33(4):880–898.
14. Moldovan TM, Abbeel P. Safe exploration in markov decision processes. *arXiv preprint arXiv:1205.4810.* 2012.
15. Quintía Vidal P, Iglesias Rodríguez R, Rodríguez González MÁ, Vázquez Regueiro C. Learning on real robots from experience and simple user feedback. 2013.
16. Abbeel P, Coates A, Ng AY. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research.* 2010;29(13):1608–1639.
17. Song Y, Li Yb, Li Ch, Zhang Gf. An efficient initialization approach of Q-learning for mobile robots. *International Journal of Control, Automation and Systems.* 2012;10(1):166–172.
18. Gehring C, Precup D. Smart exploration in reinforcement learning using absolute temporal difference errors. In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems.* 2013:1037–1044.
19. Mitchell IM, Bayen AM, Tomlin CJ. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control.* 2005;50(7):947–957.

20. Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P. Control barrier functions: Theory and applications. In: *2019 18th European control conference (ECC)*.IEEE. 2019:3420–3431.
21. Wills AG, Heath WP. Barrier function based model predictive control. *Automatica.* 2004;40(8):1415–1422.
22. Feller C, Ebenbauer C. Continuous-time linear MPC algorithms based on relaxed logarithmic barrier functions. *IFAC Proceedings Volumes.* 2014;47(3):2481–2488.
23. Marvi Z, Kiumarsi B. Safety planning using control barrier function: A model predictive control scheme. In: *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*.IEEE. 2019:1–5.
24. Marvi Z, Kiumarsi B. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control.* 2021;31(6):1923–1940.
25. Wang L, Theodorou EA, Egerstedt M. Safe learning of quadrotor dynamics using barrier certificates. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*.IEEE. 2018:2460–2465.
26. Cohen MH, Belta C. Safe exploration in model-based reinforcement learning using control barrier functions. *Automatica.* 2023;147:110684.
27. Panagou D, Stipanović DM, Voulgaris PG. Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions. *IEEE Transactions on Automatic Control.* 2015;61(3):617–632.
28. Yang Y, Kiumarsi B, Modares H, Xu C. Model-free λ-policy iteration for discrete-time linear quadratic regulation. *IEEE Transactions on Neural Networks and Learning Systems.* 2021.
29. Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL. Optimal and autonomous control using reinforcement learning: A survey. *IEEE transactions on neural networks and learning systems.* 2017;29(6):2042–2062.
30. Jiang Y, Jiang ZP. *Robust adaptive dynamic programming*. John Wiley & Sons, 2017.
31. Ames AD, Grizzle JW, Tabuada P. Control barrier function based quadratic programs with application to adaptive cruise control. In: *53rd IEEE Conference on Decision and Control*.IEEE. 2014:6271–6278.
32. Haddad WM, Chellaboina V. *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton university press, 2008.
33. Jankovic M. Combining control Lyapunov and barrier functions for constrained stabilization of nonlinear systems. In: *2017 American control conference (ACC)*.IEEE. 2017:1916–1922.
34. Ames AD, Xu X, Grizzle JW, Tabuada P. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control.* 2016;62(8):3861–3876.
35. Saridis GN, Lee CSG. An approximation theory of optimal control for trainable manipulators. *IEEE Transactions on systems, Man, and Cybernetics.* 1979;9(3):152–159.
36. Krstic M, Li ZH. Inverse optimal design of input-to-state stabilizing nonlinear controllers. *IEEE Transactions on Automatic Control.* 1998;43(3):336–350.
37. Jankovic M. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica.* 2018;96:359–367.
38. Al-Tamimi A, Lewis FL, Abu-Khalaf M. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica.* 2007;43(3):473–481.
39. Jiang Y, Jiang ZP. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica.* 2012;48(10):2699–2704.
40. Lewis FL, Vrabie D, Vamvoudakis KG. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine.* 2012;32(6):76–105.

AUTHOR BIOGRAPHY

Author Name. Please check with the journal's author guidelines whether author biographies are required. They are usually only included for review-type articles, and typically require photos and brief biographies for each author.