Proceedings of Machine Learning Research vol XX:1–14, 2023

Provably Safe Model-Free Control with Low Latency

Ritabrata Ray RITABRAR@ANDREW.CMU.EDU
B42, Porter Hall, Carnegie Mellon University
Pittsburgh, PA 15213

Yorie Nakahira YNAKAHIR@ANDREW.CMU.EDU
B20, Porter Hall, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this paper, we propose a low-latency safe control algorithm that guarantees system safety at all times. This algorithm applies to any non-linear control-affine system and is robust against uncertainties in the system dynamics model. It can also be adapted with reinforcement learning algorithms like REINFORCE to guarantee safe exploration at all times for learning in model-free settings. Further, we extend our methods to develop another low-latency algorithm that applies to the same settings and enjoys a provable guarantee for quick recovery from an initial unsafe state to safe behaviour.

Keywords: Safe control, Forward invariance/convergence, Model-free reinforcement learning.

1. Introduction

Safety is of critical importance for many intelligent control systems. Although many safe control algorithms are developed for known systems, it is challenging to ensure safety for systems with unknown dynamics. The system dynamics may have large uncertainty when learning the system model and control actions in real time, or the dynamics could be too complex to accurately model. In this paper we present an algorithm that assumes very little knowledge of the system dynamics and can be merged with existing model-based and model-free reinforcement learning algorithms to provably guarantee safe behavior at all times.

Related Work. Several methods exist for safe control of non-linear systems with unknown dynamics, which can be broadly classified into model-free and model-driven approaches. Model-based techniques are usually preferred when the system dynamics can be modelled properly and the uncertainties stem from unknown parameters in the system dynamics model. Common model-based techniques include safe adaptive control with Lyapunov methods: Taylor and Ames (2020); Lopez et al. (2021); Fan et al. (2020); Choi et al. (2021); Nguyen and Sreenath (2020); Dean et al. (2020); Cosner et al. (2021); system identification for safe control techniques: Ho et al. (2021); Grover et al. (2021); Wigren et al. (2022); Jagtap et al. (2020); Nelles (2001), regret-based online-learning algorithms: Luo et al. (2022); Kakade et al. (2020); Gaussian process methods for model-based safe reinforcement learning: Berkenkamp et al. (2017); Ma et al. (2022); Cowen-Rivers et al. (2020); Fan and Li (2019); Polymenakos et al. (2017); Sui et al. (2015); Turchetta et al. (2016). The model-based techniques usually require restricted classes of functions for system dynamics, and small uncertain-

© 2023 R. Ray & Y. Nakahira.

ties (bounded or Gaussian) in the unknown parameters. Such techniques lead to safe behavior only after the algorithm has learned the unknown parameters to a sufficient degree of accuracy and suffer from high latency issues.

On the other hand, model-free reinforcement learning algorithms are used when the system dynamics cannot be accurately modelled, and the source of uncertainties are unknown. Such techniques directly learn the safe control action from the observed roll-out trajectories and gradually learn a policy which commits lesser mistakes (unsafe behavior) as the training progresses. Such techniques either penalize the reward functions of the MDP: Zhang et al. (2022); Dong et al. (2020), or use trust region methods for constrained reinforcement learning: Achiam et al. (2017); Bharadhwaj et al. (2021); Bisi et al. (2020); Han et al. (2020); Vuong et al. (2019); Yang et al. (2020), or use Lyapunov techniques along with reinforcement learning: Qin et al. (2022); Zhao et al. (2021); Choi et al. (2020); Chow et al. (2018, 2019); Dong et al. (2020); Huh and Yang (2020); Jeddi et al. (2021); Kumar and Sharma (2017); Perkins and Barto (2003). Gu et al. (2022); García et al. (2015); Dawson et al. (2022) survey some of the existing techniques for safe control.

Challenges. The existing methods for safe control have the following limitations:

1. Model-based adaptive control techniques assume a certain structure to the dynamics model and assume bounded uncertainties in the unknown parameters. Such methods can lead to unsafe behavior near the starting time until the algorithm learns the parameters to a sufficient degree of accuracy.
2. Model-based Reinforcement learning algorithms using Gaussian processes also assume Gaussian uncertainties and similar to adaptive control methods, may lead to unsafe behavior during the initial phases of exploration.
3. Model-free reinforcement learning algorithms suffer from large sample complexity and it takes longer for the agent to learn the safe control. Trust region based constrained optimization methods only limit the safety violations up to a constant upper bound and hence cannot avoid safety violations at all times.
4. Lyapunov based techniques (could be combined with Reinforcement Learning algorithms) also require restrictions on the initial state to guarantee forward invariance for safety. These techniques usually incur high latency.

Our contributions. We present a safe control algorithm which provably guarantees safe behavior at all times (zero violations). Our proposed algorithm is designed for systems with dynamics which are affine in the control variable. It only requires the knowledge of the signs (with a non-zero confidence margin) along which the control variables affect the system state dynamics. It has the following merits:

1. The proposed method assumes no knowledge of the system dynamics other than the signs along which the control variables affect the system dynamics. This addresses the challenges: 1,2 of bounded/Gaussian uncertainty as well as restricted system dynamics model. Figures 1, 2 in 7 and theorem 2 in 6 demonstrate this.
2. The proposed method does not require to perform any computationally heavy task to compute the safe control action. This addresses the latency challenge 4 and makes it suitable for real-time applications.
3. The proposed method ensures zero-safety violations at all times, thus addressing challenge 3 associated with standard constrained policy optimization based deep reinforcement learning algorithms. Figures 2, 4 and theorem 2 illustrate this merit.

4. The above merits allow our method to be used as a low-latency tool that can be used on top of model-free reinforcement learning algorithms and allow for provable zero-violation guarantees during the exploration and lead to learning of safe control policies. Figures 3 and 4 in section 7 demonstrate this merit.
5. Similarly, if a nominal controller (satisfying performance objectives) is known for model-based adaptive control/reinforcement learning settings then our method can be used to create a low-latency tool that sits on top of the control algorithm and modifies the control action to guarantee zero-violations when the system state comes close to the unsafe region. Figures 1, 2 and theorem 2 support this merit.
6. Our technique can be easily extended to get a low-latency recovery tool which works in the same settings as above and can quickly recover the system from an initial unsafe state. Figures 5, 6 and theorem 3 show this merit.

Notation. Throughout the paper we adopt the following notations:

1. For any function $f : \mathbb{R}^d \to \mathbb{R}$, ∇f denotes its gradient with respect to x, i.e. $\nabla_x f(x)$. Let $x_t : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ denote any dynamical system as a function of time t, then $\dot{f}$ denotes the time derivative of the function $f(x_t)$, $\dot{f}^+$ denotes its right hand time derivative, and $\dot{f}^-$ denotes its left hand time derivative: i.e.,
$$\dot{f}^+ = \lim_{\delta t \to 0^+} \frac{f(x_{t+\delta t}) - f(x_t)}{\delta t} \tag{1}$$
$$\dot{f}^- = \lim_{\delta t \to 0^+} \frac{f(x_t) - f(x_{t-\delta t})}{\delta t}. \tag{2}$$
2. For any two vectors u, v of the same dimensions, $u \cdot v$ denotes their standard inner product in Euclidean space.

2. Preliminaries and Problem Statement

We consider a continuous time system whose state x_t is continuous in time t and it's dynamics is assumed to take the following form:
$$\dot{x}_t^+ = f(x_t) + g(x_t)u_t, \tag{3}$$
where $f : \mathbb{R}^d \to \mathbb{R}^d$, and $g : \mathbb{R}^d \to \mathbb{R}^{d \times p}$ are continuous functions, $u \in \mathbb{R}^p$ is the control action. We assume that the state history $\{x_\tau : \tau \leq t\}$ is fully observable by the controller at time t. We assume that $g(x_t)$ has full row-rank for any x_t and consequently its right Moore-Penrose pseudo-inverse $g^+(x_t) \in \mathbb{R}^{p \times d}$ exists and satisfies:
$$g(x_t)g^+(x_t) = I, \tag{4}$$
where I is the $d \times d$ identity matrix. A system is considered *safe* when the state x_t lies within a certain region $\mathcal{S} \subset \mathbb{R}^d$ called the safe set at all times t.

Definition 1 (Forward Invariant Set) *A set $\mathcal{S}$ is said to be forward invariant with respect to a dynamical system x_t, if starting at a point $x_0 \in \mathcal{S}$, we have $x_t \in \mathcal{S}$, $\forall t$.*

Therefore, the system is safe if the safe set $\mathcal{S}$ is forward invariant with respect to the closed loop dynamics. We characterize the safe set $\mathcal{S}$ by the level set of a continuously differentiable *barrier function* $\phi : \mathbb{R}^d \to \mathbb{R}$ as follows:
$$\mathcal{S} = \{x : \phi(x) \geq 0,\ x \in \mathbb{R}^d\}. \tag{5}$$

Given a threshold parameter $\theta > 0$, we define the safety subset:

$$\mathcal{S}_\theta = \{x : \phi(x) \geq \theta\} \subset \mathcal{S}, \tag{6}$$

to be the super-level set of the barrier function ϕ, and its boundary to be

$$\partial\mathcal{S}_\theta = \{x : \phi(x) = \theta\}. \tag{7}$$

In certian cases, there may exist a *nominal controller*:

$$u_t = u_{nom}(x_t), \tag{8}$$

where $u_{nom} : \mathbb{R}^d \to \mathbb{R}^p$ is a continuous mapping from the state space to the control action space, which is designed to achieve the operational/performance objectives of the system but it does not necessarily guarantee safety. For a non-linear system with control dynamics given by (3), the objective is to design a controller that guarantees safety at all times with minimum operational interruptions. We assume that the user has no knowledge of the function $f(.)$, and very little knowledge of the function $g(.)$. We develop the algorithm in two stages, first when the user knows the function $g(.)$ (see Appendix of the full paper[1]) and then the general case with only partial structural information about $g(x_t)$ through an Oracle call. See the appendix of the full paper for some justifications behind the Oracle assumptions.

3. Proposed Method

Here, we present our proposed algorithm that can work for systems with dynamics described by (3) with a completely unknown $f(x)$ and the user has access to only partial information about the function $g(x)$. This partial information comes in two forms: existing offline information about the structure of the matrix $g(x_t)$ and online information about some parameters of the matrix $g(x_t)$ which comes from an Oracle call. We assume the following regarding the partial information about $g(x_t)$ that is available to the user at any time instant t:

Assumption I: Offline Structural Assumption.
The rows of the matrix $g(x_t)$ (denoted by $v_{1,t}, v_{2,t}, \ldots, v_{d,t}$) are orthogonal and non-zero vectors, for every x_t. We have:

$$v_{i,t} \cdot v_{j,t} = 0,\ v_{i,t} \neq 0\ \forall i \neq j, i, j \in \{1, \ldots, d\}. \tag{9}$$

Consequences of this assumption: The singular value decomposition of $g(x_t)$ can be written as:

$$g(x_t) = I\Sigma_t V_t^\intercal. \tag{10}$$

Here I is the $d \times d$ identity matrix, $\Sigma \in \mathbb{R}^{d\times p}$ with

$$\Sigma_{t,(i,j)} = \begin{cases} \lambda_{i,t}, \text{ if } i = j, i \in \{1, \ldots, d\} \\ 0, \text{ otherwise} \end{cases} \tag{11}$$

and

$$V_t = \begin{bmatrix} u_1, u_2, \ldots, u_d \end{bmatrix} \tag{12}$$

1. Full-Version

is an orthonormal matrix satisfying:

$$V_t V_t^\intercal = I, \tag{13}$$

where the columns of V_t are the normalized row vectors of $g(x_t)$, i.e.

$$u_i = \frac{1}{\left\|v_{i,t}\right\|} v_{i,t} \;\; \forall i \in \{1, \ldots, d\}.$$

Since the rows of V_t are non-zero, the singular values $\lambda_{i,t} \neq 0, \; \forall t, i \in \{1, \ldots, d\}$.

Assumption II: Online Oracle Assumption.
There exists an oracle $\mathcal{O} : \mathbb{R}^d \to \mathbb{R}^{p \times d} \times \mathbb{R}^d$, which takes the state variable x_t as input and returns the output:

$$\mathcal{O}(x_t) = \left(V_t, (\hat{\lambda}_{1,t}, \ldots, \hat{\lambda}_{d,t})\right). \tag{14}$$

Here, the matrix V_t is the orthonormal matrix corresponding to the rows of $g(x_t)$ given by (12), and the numbers $\hat{\lambda}_{i,t}$ are coarse bounds representing the unknown singular values $\lambda_{i,t}$ with the correct sign, i.e. $\forall t, i \in \{1, \ldots, d\}$, we have:

$$\hat{\lambda}_{i,t} \in \begin{cases} (0, \lambda_{i,t}] \text{ if } \lambda_{i,t} > 0, \\ [\lambda_{i,t}, 0) \text{ if } \lambda i, t < 0. \end{cases} \tag{15}$$

Consequences of this assumption: The Oracle's output given by (14), allows the algorithm to construct the following representative singular value matrix $\hat{\Sigma}_t \in \mathbb{R}^{d \times p}$, given by:

$$\hat{\Sigma}_{t,(i,j)} = \begin{cases} \hat{\lambda}_{i,t}, & \text{if } i = j, i \in \{1, \ldots, d\} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

and computes its right pseudo-inverse as the matrix $\hat{\Sigma}_t^+ \; (\in \mathbb{R}^{p \times d})$ given by:

$$\hat{\Sigma}^+_{t,(i,j)} = \begin{cases} \frac{1}{\hat{\lambda}_{i,t}}, & \text{if } i = j, i \in \{1, \ldots, d\} \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The algorithm uses these to compute the representative matrix $\hat{g}(x_t)$ of the original matrix $g(x_t)$ as:

$$\hat{g}(x_t) = I \hat{\Sigma}_t V_t^\intercal. \tag{18}$$

Algorithm Description.

The proposed algorithm is summarized in algorithm 1. The algorithm calls the oracle with the state variable x_t as input and from its output computes the representative matrix $\hat{g}(x_t)$ of the matrix $g(x_t)$ as described by (18). It then computes the right pseudo-inverse $\hat{g}^+(x_t)$ of this representative matrix as:

$$\hat{g}^+(x_t) = V_t \hat{\Sigma}_t^+ I, \tag{19}$$

where I is the $d \times d$ identity matrix, $\hat{\Sigma}_t^+$ is the $p \times d$ matrix given by (17), and V_t is the $p \times p$ matrix given by (12). The algorithm then computes the correction strength multiplier matrix $\gamma_t(x_t) \in \mathbb{R}^{d \times d}$ as follows:

$$\gamma_{t,(i,j)}(x_t) = \begin{cases} 1, \text{ if } i = j, \text{ and } \nabla\phi(x_t)_i \dot{x}_{t,i}^- < 0, i \in \{1, \dots, d\} \\ 0, \text{ otherwise.} \end{cases} \tag{20}$$

Using the above correction strength multiplier matrix, the algorithm computes its correction control u_{corr} as:

$$u_{corr} = u_{last} - 2\hat{g}^+(x_t)\gamma_t(x_t)\dot{x}_t^-. \tag{21}$$

Algorithm 1 Proposed Algorithm with Oracle

Input : Control Barrier Function: $\phi(.)$, Threshold: $\theta > 0$, Initial value of the state variable: $x_0 \in \mathcal{S}$ such that $\phi(x_0) > \theta$, the Oracle $\mathcal{O}$ as defined in (14), and the nominal controller: $u_{nom}(.)$.

Hyperparameters: Discretization time step: T_s.

Initialize : $x_{t=0} \leftarrow x_0,\ u_{last} \leftarrow u_{nom}(x_0),\ x_{t=0}^- \leftarrow x_0$.

for *every* $t > 0$, **do**
- Receive the current state x_t from observation.
- Compute $\phi(x_t)$.
- Compute the time derivative of state variable: $\dot{x}_t^- \leftarrow \frac{1}{T_s}(x_t - x_t^-)$.
- Compute: $\dot{\phi}^-(x_t) \leftarrow \nabla\phi(x_t) \cdot \dot{x}_t^-$.
- **switch do**
  - **case** $\phi(x_t) \leq \theta$, *and* $\dot{\phi}^-(x_t) \geq 0$ **do**
    - $u \leftarrow u_{last}$.
  - **end**
  - **case** $\phi(x_t) \leq \theta$, *and* $\dot{\phi}^-(x_t) < 0$ **do**
    - Call the Oracle with input x_t, and receive: $\mathcal{O}(x_t) = \left(V_t, (\hat{\lambda}_{1,t}, \dots, \hat{\lambda}_{d,t})\right)$, according to (14).
    - Compute the representative matrix: $\hat{g}(x_t)$ using (18).
    - Compute the right Pseudo-inverse: $\hat{g}^+(x_t)$ of the matrix $\hat{g}(x_t)$ using (19).
    - Compute: u_{corr}, using (21) and assign $u \leftarrow u_{corr}$.
  - **end**
  - **default:** $u \leftarrow u_{nom}(x_t)$.
- **end**
- Play u, assign $u_{last} \leftarrow u$, and store last value of x_t as $x_t^- \leftarrow x_t$.

end

4. Safe exploration for model-free reinforcement learning algorithms

In this section, we illustrate merit 4 by presenting our technique as a low-latency tool which can be merged on top of the model-free reinforcement learning algorithm REINFORCE Williams (1992). We discretize the state and action spaces to finite spaces $\mathcal{S}$ and $\mathcal{A}$ respectively. At each time step, the agent receives a reward $r(s, a)$ for the current state and action pair (s, a) where the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is stationary with respect to time and is designed according to the environment and the desired task. The goal is to maximize the expected discounted aggregate sum of rewards, i.e.,

Algorithm 2 REINFORCE with Safety

Input : Finite state and action spaces $\mathcal{S}$, $\mathcal{A}$ respectively, terminal state $T \in \mathcal{S}$, control barrier function $\phi(.)$, the correction controller $C(x, u, u_{last})$ as defined in (22), Initial value of the state variable: $s_0 \in \mathcal{S}$ such that $\phi(s_0) > 0$, the Oracle $\mathcal{O}$ as defined in (14), a deep neural network with parameters $\theta \in \Theta$.

Hyperparameters: Discounting Factor γ, discretization time step: T_s.

Initialize : $x_{t=0} \leftarrow x_0,\ u_{last} \leftarrow u_{nom}(x_0),\ x^-_{t=0} \leftarrow x_0, \theta \leftarrow \theta_0$.

for *every episode* $\tau = 0, 1, 2, \ldots$ **do**

Roll-out Episode τ.

Set $t \leftarrow 0$, $G_{-1} \leftarrow 0$.

while $x_t \neq T$ **do**

Receive the current state x_t from observation, obtain the corresponding discrete state s_t.

Compute the time derivative of state variable: $\dot{x}^-_t \leftarrow \frac{1}{T_s}(x_t - x^-_t)$.

Sample $a_t \sim \pi_{\theta_\tau}(.|s_t)$.

Overwrite the sampled control action a_t as: $u_t = C(s_t, \dot{x}^-_t, a_t, u_{last})$.

Play u_t and collect reward $r_t \leftarrow r(s_t, u_t)$.

$u_{last} \leftarrow u_t$, $x^-_t \leftarrow x_t$, $t \leftarrow t + 1$.

end

For the above episode τ: Let $(s_0, a_0, u_0, s_1, a_1, u_1, s_2, \ldots)$ be the roll-out trajectory collected.

Compute the variables G_t as: $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$.

Estimate the policy gradient sample as:

$$\nabla J(\theta_\tau) = \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta=\theta_\tau} G_t.$$

Update the policy network parameters as: $\theta_{\tau+1} \leftarrow \theta_\tau + \nabla J(\theta_\tau)$.

end

$J = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, where $\gamma \in [0, 1)$ is the discounting factor. Let $C(x, u, u_{last})$ denote the correction controller given by algorithm 1 where the safety threshold θ is chosen at $\theta = 0$. That is,

$$C(x, \dot{x}^-, u, u_{last}) = \begin{cases} u & \text{if } \phi(x) > 0 \\ u_{last} & \text{if } \phi(x) \leq 0 \text{ and } \dot{\phi}^-(x) \geq 0 \\ u_{corr} \text{ given by (21)} & otherwise \end{cases} \tag{22}$$

We show in the Appendix of the full paper that the policy gradient estimated by the algorithm 2 is indeed an unbiased estimate of the true policy gradient. During the entire process, safety is guaranteed by theorem 2, where the action sampled from the stochastic policy could be seen as the nominal controller action. This allows for safe exploration while learning a safe controller in the model-free setting.

Algorithm 3 Safety and Recovery with Oracle

Input : Control Barrier Function: $\phi(.)$, threshold: $\theta > 0$, initial value of the state variable: x_0, the Oracle $\mathcal{O}$ as defined in (14), and the nominal controller: $u_{nom}(.)$.
Hyperparameters: Discretization time step: T_s, recovery hyperparameter $\eta > 0$.
Initialize : $x_{t=0} \leftarrow x_0,\ u_{last} \leftarrow u_{nom}(x_0),\ x^-_{t=0} \leftarrow x_0$.
for *every* $t > 0$, **do**
  Receive the current state x_t from observation and compute $\phi(x_t)$.
  Compute the time derivative of state variable: $\dot{x}^-_t \leftarrow \frac{1}{T_s}(x_t - x^-_t)$.
  Compute: $\dot{\phi}^-(x_t) \leftarrow \nabla\phi(x_t) \cdot \dot{x}^-_t$.
  switch do
    case 1. $\phi(x_t) \leq \theta$, *and* $\dot{\phi}^-(x_t) \geq 0$ **do**
      Assign $u \leftarrow u_{last}$.
    end
    case 2. $\phi(x_t) \leq \theta$, *and* $\dot{\phi}^-(x_t) < 0$ **do**
      Call the Oracle with input x_t, and receive: $\mathcal{O}(x_t) = \left(V_t, (\hat{\lambda}_{1,t}, \ldots, \hat{\lambda}_{d,t})\right)$, according to (14).
      Compute the representative matrix: $\hat{g}(x_t)$ using (18).
      Compute the right Pseudo-inverse: $\hat{g}^+(x_t)$ of the matrix $\hat{g}(x_t)$ using (19).
      Compute: u_{corr}, using (21) as $u \leftarrow u_{corr}$.
    end
    case 3. $\phi(x_t) < \theta$ **do**
      Call the Oracle with input x_t, and receive: $\mathcal{O}(x_t) = \left(V_t, (\hat{\lambda}_{1,t}, \ldots, \hat{\lambda}_{d,t})\right)$.
      Compute the representative matrix: $\hat{g}(x_t)$ using (18).
      Compute the right Pseudo-inverse: $\hat{g}^+(x_t)$ of the matrix $\hat{g}(x_t)$ using (19).
      Compute the matrix $\gamma_t(x_t)$ using (20).
      Compute: $n_d(x)$ using (24) and u_{rec} using (23), assign $u \leftarrow u_{rec}$.
    end
    default: $u \leftarrow u_{nom}(x_t)$.
  end
  Play u, $u_{last} \leftarrow u$, $x^-_t \leftarrow x_t$.
end

5. Recovery from unsafe initial state

In this section, we illustrate merit 6 of our technique by extending algorithm 1 with added guarantee of recovery to the safe region, when started at an unsafe point $x_0 \notin \mathcal{S}$. Here, we use the recovery control as:

$$u_{rec} = u_{last} - \hat{g}^+(x_t)\left(\gamma_t(x_t)\dot{x}^-_t + n_d(x_t)\nabla\phi(x_t)\right), \tag{23}$$

where $n_d(x)$ is given by

$$n_d(x) = -\frac{\eta}{\left\|\nabla\phi(x)\right\|^2}. \tag{24}$$

6. Performance Guarantees

In this section, we state the main results of this paper: theorems stating the performance guarantees of the algorithms in the above sections. The proofs of the following results can be found in the Appendix of the full version of the paper. The first result guarantees safety at all times for our proposed method 1.

Theorem 2 (Forward Invariance for algorithm 1) *If $x_0 \in \mathcal{S}_\theta$, then the set $\mathcal{S}_\theta$ is forward invariant with respect to the controller given by algorithm 1.*

Next we have the guarantee for quick recovery from an unsafe initial state for algorithm 3.

Theorem 3 (Forward Persistence for algorithm 3) *If $x_0 \notin \mathcal{S}_\theta$, then the set $\mathcal{S}_\theta$ is forward convergent (i.e., $\exists \tau > 0$ such that $x_\tau \in \mathcal{S}_\theta$) when the controller given by algorithm 3 is used and the rate of convergence is lower bounded by the hyperparameter η. Further, once the system state enters $\mathcal{S}_\theta$, the set $\mathcal{S}_\theta$ is forward invariant (i.e., $x_t \in \mathcal{S}_\theta \; \forall t \geq \tau$.) for the system when algorithm 3 is used.*

7. Numerical Study

In this section, we experimentally demonstrate the merits: 1, 3, 4, 6 of our method through numerical simulations [2]. First, we consider an unknown dynamical system with parametric uncertainties and run our algorithm (denoted here as sign flip) with an unsafe nominal controller and show that our algorithm guarantees safety at all times. We compare it against the performance of the following adaptive safe control algorithms on the same system and with the same initialization: aCBF from Taylor and Ames (2020), RaCBF and RaCBF+SMID (here denoted as RaCBFS) from Lopez et al. (2021), $\mathcal{A}_\pi-$SEL algorithm from Ho et al. (2021) (denoted here as cbc), and the BALSA algorithm from Fan et al. (2020) (denoted here as balsa). The implementation details of this simulation could be found in the appendix. We obtain the plots 1, 2, which show that our algorithm achieves safety at all times whereas the other algorithms take some time to learn the unknown parameters and exhibit safe behavior.

Next, we demonstrate algorithm 2 and compare it against some model-free reinforcement learning algorithms like REINFORCE (Williams (1992)), CPO (Achiam et al. (2017)). The following plots: 3, and 4 are obtained which show that our algorithm stays safe at all times compared to occasional safety violations of the other algorithms. Our algorithm also converges to learn a safe route from source to destination (longer) without much loss in the convergence rate.

Lastly, we demonstrate the quick recovery ability (forward convergence and persistence) of algorithm 3 by starting the system at five distinct initial positions, one of which is safe and the other four unsafe. From the plots: 5, and 6, we observe that the system quickly recovers to the safe region (forward convergence; recovery time depends on initial state), and the system remains safe afterwords (forward persistence). The details of the experimental setup can be found in the Appendix of the full version of the paper.

Acknowledgments

We would like to thank Xiyu Deng for her help with the draft writing.

2. Simulation Codes

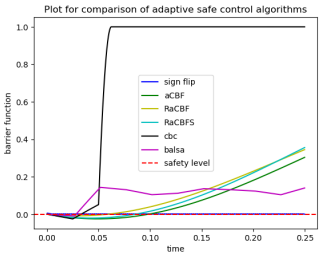

Figure 1: The barrier function $\phi(x)$ with respect to time for several safe adaptive control algorithms vs our algorithm Sign flip

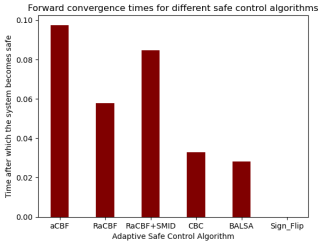

Figure 2: Comparison of Forward Convergence times of several adaptive safe control algorithms vs our algorithm Sign flip.

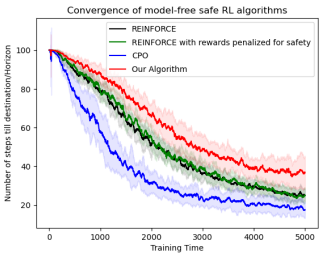

Figure 3: The average number of steps needed to reach the destination with respect to the training epoch.

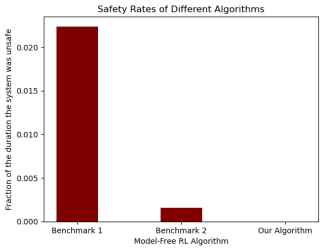

Figure 4: Fraction of time the system was in unsafe states for different algorithms. Benchmark 1 is REINFORCE with rewards adjusted for safety, Benchmark 2 is CPO.

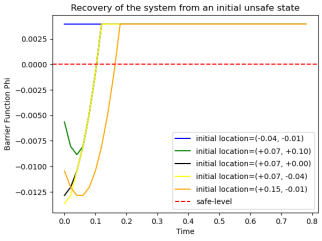

Figure 5: The average number of steps needed to reach the destination with respect to the training epoch.

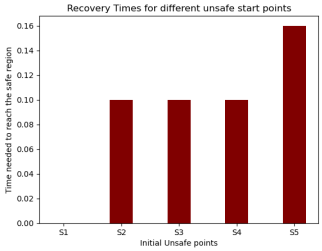

Figure 6: Fraction of time the system was unsafe for different algorithms. Benchmark 1 is REINFORCE with safety-penalized rewards, Benchmark 2 is CPO.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees, 2017. URL https://arxiv.org/abs/1705.08551.

Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=iaO86DUuKi.

Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4583–4589. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/632. URL https://doi.org/10.24963/ijcai.2020/632. Special Track on AI in FinTech.

Jason Choi, Fernando Castañeda, Claire J. Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions, 2020. URL https://arxiv.org/abs/2004.07584.

Jason J. Choi, Donggun Lee, Koushil Sreenath, Claire J. Tomlin, and Sylvia L. Herbert. Robust control barrier–value functions for safety-critical control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6814–6821, 2021. doi: 10.1109/CDC45484.2021.9683085.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/4fe5149039b52765bde64beb9f674940-Paper.pdf.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Mohammad Ghavamzadeh, and Edgar A. Duéñez-Guzmán. Lyapunov-based safe policy optimization for continuous control. *CoRR*, abs/1901.10031, 2019. URL http://arxiv.org/abs/1901.10031.

Ryan K. Cosner, Andrew W. Singletary, Andrew J. Taylor, Tamas G. Molnar, Katherine L. Bouman, and Aaron D. Ames. Measurement-robust control barrier functions: Certainty in safety with uncertainty in state, 2021. URL https://arxiv.org/abs/2104.14030.

Alexander Imani Cowen-Rivers, Daniel Palenicek, Vincent Moens, Mohammed Amin Abdullah, Aivar Sootla, Jun Wang, and Haitham Ammar. SAMBA: safe model-based & active reinforcement learning. *CoRR*, abs/2006.09436, 2020. URL https://arxiv.org/abs/2006.09436.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods. *arXiv preprint arXiv:2202.11762*, 2022.

Sarah Dean, Andrew J. Taylor, Ryan K. Cosner, Benjamin Recht, and Aaron D. Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions, 2020. URL https://arxiv.org/abs/2010.16001.

Yunlong Dong, Xiuchuan Tang, and Ye Yuan. Principled reward shaping for reinforcement learning via lyapunov stability theory. *Neurocomputing*, 393:83–90, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.02.008. URL https://www.sciencedirect.com/science/article/pii/S0925231220301831.

David D. Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali-akbar Agha-mohammadi, and Evangelos A. Theodorou. Bayesian learning-based adaptive control for safety critical systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4093–4099, 2020. doi: 10.1109/ICRA40945.2020.9196709.

Jiameng Fan and Wenchao Li. Safety-guided deep reinforcement learning via online gaussian process estimation. *CoRR*, abs/1903.02526, 2019. URL http://arxiv.org/abs/1903.02526.

Javier García, Fern, and o Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL http://jmlr.org/papers/v16/garcia15a.html.

Jaskaran Grover, Changliu Liu, and Katia Sycara. System identification for safe controllers using inverse optimization. *IFAC-PapersOnLine*, 54(20):346–353, 2021. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2021.11.198. URL https://www.sciencedirect.com/science/article/pii/S2405896321022424. Modeling, Estimation and Control Conference MECC 2021.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications, 2022. URL https://arxiv.org/abs/2205.10330.

Minghao Han, Yuan Tian, Lixian Zhang, Jun Wang, and Wei Pan. Reinforcement learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee, 2020. URL https://arxiv.org/abs/2011.06882.

Dimitar Ho, Hoang M. Le, John C. Doyle, and Yisong Yue. Online robust control of nonlinear systems with large uncertainty. 2021. doi: 10.48550/ARXIV.2103.11055. URL https://arxiv.org/abs/2103.11055.

Subin Huh and Insoon Yang. Safe reinforcement learning for probabilistic reachability and safety specifications: A lyapunov-based approach. *CoRR*, abs/2002.10126, 2020. URL https://arxiv.org/abs/2002.10126.

Pushpak Jagtap, George J. Pappas, and Majid Zamani. Control Barrier Functions for Unknown Nonlinear Systems using Gaussian Processes. 2020. URL http://arxiv.org/abs/2010.05818.

Ashkan B. Jeddi, Nariman L. Dehghani, and Abdollah Shafieezadeh. Lyapunov-based uncertainty-aware safe reinforcement learning. *CoRR*, abs/2107.13944, 2021. URL https://arxiv.org/abs/2107.13944.

S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun. Information theoretic regret bounds for online nonlinear control. In *Advances in Neural Information Processing Systems*, 2020.

Abhishek Kumar and Rajneesh Sharma. Fuzzy lyapunov reinforcement learning for non linear systems. *ISA Transactions*, 67, 01 2017. doi: 10.1016/j.isatra.2017.01.026.

Brett T. Lopez, Jean-Jacques E. Slotine, and Jonathan P. How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3): 1031–1036, 2021. doi: 10.1109/LCSYS.2020.3005923.

Wenhao Luo, Wen Sun, and Ashish Kapoor. Sample-efficient safe learning for online nonlinear control with control barrier functions, 2022. URL https://arxiv.org/abs/2207.14419.

Yecheng Jason Ma, Andrew Shen, Osbert Bastani, and Jayaraman Dinesh. Conservative and adaptive penalty for model-based safe reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5404–5412, Jun. 2022. doi: 10.1609/aaai.v36i5.20478. URL https://ojs.aaai.org/index.php/AAAI/article/view/20478.

Oliver Nelles. *Nonlinear system identification. From classical approaches to neural networks and fuzzy models*. 01 2001. ISBN 978-3-642-08674-8. doi: 10.1007/978-3-662-04323-3.

Quan Nguyen and Koushil Sreenath. Robust safety-critical control for dynamic robotics, 2020. URL https://arxiv.org/abs/2005.07284.

Theodore Perkins and Andrew Barto. Lyapunov design for safe reinforcement learning control. *J. Mach. Learn. Res.*, 3:803–, 05 2003. doi: 10.1162/jmlr.2003.3.4-5.803.

Kyriakos Polymenakos, Alessandro Abate, and Stephen Roberts. Safe policy search with gaussian process models, 2017. URL https://arxiv.org/abs/1712.05556.

Zengyi Qin, Dawei Sun, and Chuchu Fan. Sablas: Learning safe control for black-box dynamical systems, 2022. URL https://arxiv.org/abs/2201.01918.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 997–1005, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sui15.html.

Andrew J. Taylor and Aaron D. Ames. Adaptive safety with control barrier functions. In *2020 American Control Conference (ACC)*, pages 1399–1405, 2020. doi: 10.23919/ACC45564.2020.9147463.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. 2016. doi: 10.48550/ARXIV.1606.04753. URL https://arxiv.org/abs/1606.04753.

Quan Vuong, Yiming Zhang, and Keith W. Ross. SUPERVISED POLICY UPDATE. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJxTroR9F7.

Anna Wigren, Johan Wågberg, Fredrik Lindsten, Adrian G. Wills, and Thomas B. Schön. Nonlinear system identification: Learning while respecting physical models using a sequential monte carlo method. *IEEE Control Systems Magazine*, 42(1):75–102, 2022. doi: 10.1109/MCS.2021.3122269.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based constrained policy optimization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rke3TJrtPS.

Linrui Zhang, Li Shen, Long Yang, Shixiang Chen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Penalized proximal policy optimization for safe reinforcement learning, 2022. URL https://arxiv.org/abs/2205.11814.

Weiye Zhao, Tairan He, and Changliu Liu. Model-free safe control for zero-violation reinforcement learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=UGp6FDaxB0f.