
Safe Reinforcement Learning in Constrained Markov Decision Processes

Akifumi Wachi¹ Yanan Sui²

Abstract

Safe reinforcement learning has been a promising approach for optimizing the policy of an agent that operates in safety-critical applications. In this paper, we propose an algorithm, SNO-MDP, that explores and optimizes Markov decision processes under unknown safety constraints. Specifically, we take a stepwise approach for optimizing safety and cumulative reward. In our method, the agent first learns safety constraints by expanding the safe region, and then optimizes the cumulative reward in the certified safe region. We provide theoretical guarantees on both the satisfaction of the safety constraint and the near-optimality of the cumulative reward under proper regularity assumptions. In our experiments, we demonstrate the effectiveness of SNO-MDP through two experiments: one uses a synthetic data in a new, openly-available environment named GP-SAFETY-GYM, and the other simulates Mars surface exploration by using real observation data.

1. Introduction

In many real applications, environmental hazards are first detected *in situ*. For example, a planetary rover exploring Mars does not obtain high-resolution images at the time of its launch. In usual cases, after landing on Mars, the rover takes close-up images or observes terrain data. Leveraging the acquired data, ground operators identify whether each position is safe. Hence, for fully automated operation, an agent must autonomously *explore* the environment and *guarantee* safety.

In most cases, however, guaranteeing safety (i.e., surviving) is *not* the primary objective. The optimal policy for ensuring safety is often extremely conservative (e.g., stay at the current position). Even though avoiding hazards is an

essential requirement, the primary objective is nonetheless to obtain rewards (e.g., scientific gain).

As a framework to solve this problem, safe reinforcement learning (safe RL, [Garcia & Fernández \(2015\)](#)) has recently been noticed by the research community. The objective of safe RL is to maximize the cumulative reward while guaranteeing or encouraging safety. Especially in problem settings in which the reward and safety functions are *unknown a priori*, however, a great deal of previous work (e.g., [Wachi et al. \(2018\)](#)) theoretically guarantees the satisfaction of the safety constraint, but the acquired policy is not necessarily near-optimal in terms of the cumulative reward. In this paper, we propose a safe RL algorithm that guarantees a near-optimal cumulative reward while guaranteeing the satisfaction of the safety constraint as well.

Related work. Conventional reinforcement learning literature has been agnostic with respect to safety, while pursuing efficiency and optimality of the cumulative reward. Representatives of such work are *probably approximately correct Markov decision process (PAC-MDP)* algorithms ([Brafman & Tennenholtz, 2002](#); [Kearns & Singh, 2002](#); [Strehl et al., 2006](#)). Algorithms with the PAC-MDP property enable an agent to learn a near-optimal behavior with a polynomial number of samples. In addition, [Kolter & Ng \(2009\)](#) and [Araya et al. \(2012\)](#) proposed algorithms to obtain an ϵ -close solution to the Bayesian optimal policy.

As the research community tries to apply RL algorithms to real-world systems, however, safety issues have been highlighted. RL algorithms inherently require an agent to explore unknown state-action pairs, and algorithms that are agnostic with respect to safety may execute unsafe actions without deliberateness. Hence, it is important to develop algorithms that guarantee safety even during training, at least with high probability.

A notable approach is *safe model-based RL* ([Berkenkamp et al., 2017](#); [Fisac et al., 2018](#)). In this domain, safety is associated with a state constraint; thus, the resulting algorithm is well suited for such contexts as a drone learning how to hover. The parameters of a drone are not perfectly known a priori, but we have prior knowledge on what states are unsafe (e.g., a pitch angle of more than 50 degrees is unsafe). In the field of control theory, constrained model predictive

¹IBM Research AI, Tokyo, Japan ²Tsinghua University, Beijing, China. Correspondence to: Akifumi Wachi <akifumi.wachi@ibm.com>, Yanan Sui <ysui@tsinghua.edu.cn>.

control (Mayne et al., 2000) has been popular. For example, Aswani et al. (2013) proposed an algorithm for guaranteeing robust feasibility and constraint satisfaction for a learned model using constrained model predictive control.

On the other hand, safe *model-free* RL has also been successful, especially in continuous control tasks. For example, Achiam et al. (2017) proposed the constrained policy optimization (CPO) algorithm while guaranteeing safety in terms of constraint satisfaction. Moreover, Chow et al. (2019) leveraged Lyapunov functions to learn policies with high expected cumulative reward, while guaranteeing the satisfaction of safety constraints.

Finally, several previous studies have addressed how to explore a safe space in an environment that is unknown a priori (Sui et al., 2015; Turchetta et al., 2016). This type of problem setting is well-suited for cases such as a robot exploring an uncertain environment (e.g., a planetary surface, a disaster site). In particular, under the safety constraint, Sui et al. (2018) proposed a stepwise algorithm for finding the maximum value of the reward function in a state-less setting (i.e., the bandit problem), while Wachi et al. (2018) proposed an algorithm for maximizing the cumulative reward in an MDP setting (i.e., the planning problem).

Our contributions. We propose a safe near-optimal MDP, SNO-MDP algorithm, for achieving a near-optimal cumulative reward while guaranteeing safety in a constrained MDP. This algorithm first explores the safety function and then optimizes the cumulative reward in the certified safe region. We further propose an algorithm called Early Stopping of Exploration of Safety (ES²) to achieve faster convergence while maintaining probabilistic guarantees with respect to both safety and reward. We examine SNO-MDP by applying PAC-MDP analysis and prove that, with high probability, the acquired policy is near-optimal with respect to the cumulative reward while guaranteeing safety. We build an openly-available test-bed called GP-SAFETY-GYM for synthetic experiments.¹ The safety and efficiency of SNO-MDP are then evaluated with two experiments: one in the GP-SAFETY-GYM synthetic environment, and the other using real Mars terrain data.

2. Problem Statement

A safety constrained MDP is defined as a tuple

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, f, r, g, \gamma \rangle,$$

where \mathcal{S} is a finite set of states $\{s\}$, \mathcal{A} is a finite set of actions $\{a\}$, $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is a deterministic state transition function, $r : \mathcal{S} \rightarrow (0, R_{\max}]$ is a bounded reward function,

¹https://github.com/akifumi-wachi-4/safe_near_optimal_mdp

$g : \mathcal{S} \rightarrow \mathbb{R}$ is a safety function, and $\gamma \in \mathbb{R}$ is a discount factor. We assume that both the reward function r and the safety function g are *not known a priori*. At every time step $t \in \mathbb{N}$, the agent must be in a “safe” state. More concretely, for a state s_t , the safety function value $g(s_t)$ must be above a threshold $h \in \mathbb{R}$; that is, the safety constraint is represented as $g(s_t) \geq h$.

A policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ maps a state to an action. The value of a policy is evaluated according to the discounted cumulative reward under the safety constraint. Let $V_{\mathcal{M}}$ denote the value function in the MDP, \mathcal{M} . In summary, we represent our problem as follows:

$$\begin{aligned} \text{maximize: } & V_{\mathcal{M}}^{\pi}(s_t) = \mathbb{E} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r(s_{t+\tau}) \mid s_t \right] \\ \text{subject to: } & g(s_{t+\tau}) \geq h, \quad \forall \tau = [0, \infty]. \end{aligned}$$

Difficulties. In conventional safety-constrained RL algorithms, the safety function is assumed to be known a priori. The key difference lies in the fact that we need to explore a safety function that is unknown a priori while guaranteeing satisfaction of the safety constraint.

However, it is intractable to solve the above problem without further assumptions. First of all, without prior information on the state-and-action pairs known to be safe, an agent cannot take any viable action at the very beginning. Second, if the safety function does not exhibit any regularity, then the agent cannot infer the safety of decisions.

Assumptions. To overcome the difficulties mentioned above, we adopt two assumptions from Sui et al. (2015) and Turchetta et al. (2016). For the first difficulty, we simply assume that the agent starts in an initial set of states, S_0 , that is known a priori to be safe. Second, we assume regularity for the safety function. Formally speaking, we assume that the state space \mathcal{S} is endowed with a positive definite kernel function, k^g , and that the safety function has a bounded norm in the associated reproducing kernel Hilbert space (RKHS, Schölkopf & Smola (2001)). The kernel function, k^g is employed to capture the regularity of the safety function. Finally, we further assume that the safety function g is L -Lipschitz continuous with respect to some distance metric $d(\cdot, \cdot)$ on \mathcal{S} .

As with the safety function, we also assume that the reward function has a bounded norm in the associated RKHS, and that its regularity is captured by another positive definite kernel function, k^r .

The above assumptions allow us to characterize the reward and safety functions by using Gaussian processes (GPs, see Rasmussen (2004)). By using the GP models, the values of r and g at unobserved states are predicted according to

previously observed functions' values. An advantage of leveraging GPs is that we can obtain both optimistic and pessimistic measurements of the two functions by using the inferred means and variances. A GP is specified by its mean, $\mu(s)$, and covariance, $k(s, s')$. The reward and safety functions are thus modeled as

$$\begin{aligned} r(s) &= \mathcal{GP}(\mu^r(s), k^r(s, s')), \\ g(s) &= \mathcal{GP}(\mu^g(s), k^g(s, s')). \end{aligned}$$

Without loss of generality, let $\mu(s) = 0$ for all $s \in \mathcal{S}$. For the reward and safety functions, we respectively model the observation noise as $y^r = r(s) + n^r$ and $y^g = g(s) + n^g$, where $n^r \sim \mathcal{N}(0, \sigma_r^2)$ and $n^g \sim \mathcal{N}(0, \sigma_g^2)$. The posteriors over r and g are computed on the basis of t observations at states $\{s_1, \dots, s_t\}$. Then, for both the reward and safety functions, the posterior mean, variance, and covariance are respectively represented as

$$\begin{aligned} \mu_t(s) &= \mathbf{k}_t^\top(s) (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t, \\ \sigma_t(s) &= \mathbf{k}_t(s, s), \\ k_t(s, s') &= \mathbf{k}_t^\top(s) (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_t(s'), \end{aligned}$$

where $\mathbf{k}_t(s) = [k(s_1, s), \dots, k(s_t, s)]^\top$, and \mathbf{K}_t is the positive definite kernel matrix.

3. Background

We define two kinds of predicted safe spaces inferred by a GP as in [Turchetta et al. \(2019\)](#). First, we consider a *pessimistic* safe space, which contains states identified as safe with a greater probability than a pre-defined confidence level. Second, we derive an *optimistic* safe space that includes all states that may be safe with even a small probability.

Predicted pessimistic safe space. We use the notion of a safe space in [Turchetta et al. \(2016\)](#) as a *predicted pessimistic safe space*. For the probabilistic safety guarantee, two sets are defined. The first set, S_t^- , simply contains the states that satisfy the safety constraint with high probability. The second one, \mathcal{X}_t^- , additionally considers the ability to reach states in S_t^- (i.e., reachability) and the ability to return to the previously identified safe set, \mathcal{X}_{t-1}^- (i.e., returnability). The algorithm probabilistically guarantees safety by allowing the agent to visit only states in \mathcal{X}_t^- .

Safety is evaluated in terms of the confidence interval inferred by the GP, which is represented as

$$Q_t(s) := [\mu_{t-1}^g(s) \pm \beta^{1/2} \sigma_{t-1}^g(s)],$$

where $\beta_t \in \mathbb{R}$ is a scaling factor for the required level of safety. We consider the intersection of Q_t up to iteration t , which is defined as $C_t(s) = Q_t(s) \cap C_{t-1}(s)$, where $C_0(s) = [h, \infty]$ for all $s \in S_0$. The lower and upper

bounds on $C_t(s)$ are denoted by $l_t(s) := \min C_t(s)$ and $u_t(s) := \max C_t(s)$, respectively.

The first set S_t^- contains states such that the safety constraint is satisfied with high probability. It is formulated using the lower bound of the safety function, l and the Lipschitz constant, L , as follows:

$$S_t^- = \{s \in \mathcal{S} \mid \exists s' \in \mathcal{X}_{t-1}^- : l_t(s') - L \cdot d(s, s') \geq h\}.$$

Next, the reachable and returnable sets are considered. Even though a state is in S_t^- , it might be surrounded by unsafe states. Given a set X , the states reachable from X in one step are given by $R_{\text{reach}}(X) = X \cup \{s \in \mathcal{S} \mid \exists s' \in X, a \in \mathcal{A} : s = f(s', a)\}$. Even after arriving at a state with reachability, the agent may *not* be able to move to another state because of a lack of safe actions. Hence, before moving to a state s , we consider whether or not there is at least one viable path from s . The set of states from which the agent can return to a set \bar{X} through another set of states X in one step is given by $R_{\text{ret}}(X, \bar{X}) = \bar{X} \cup \{s \in X \mid \exists a \in \mathcal{A} : f(s, a) \in \bar{X}\}$. Thus, an n -step returnability operator is given by $R_{\text{ret}}^n(X, \bar{X}) = R_{\text{ret}}(X, R_{\text{ret}}^{n-1}(X, \bar{X}))$, with $R_{\text{ret}}^1(X, \bar{X}) = R_{\text{ret}}(X, \bar{X})$. Finally, the set containing all the states that can reach \bar{X} along an arbitrary long path in X is defined as $\bar{R}_{\text{ret}}(X, \bar{X}) = \lim_{n \rightarrow \infty} R_{\text{ret}}^n(X, \bar{X})$.

Finally, the desired *pessimistic* safe space, \mathcal{X}_t^- is a subset of S_t^- and also satisfies the reachability and returnability constraints; that is,

$$\mathcal{X}_t^- = \{s \in S_t^- \mid s \in R_{\text{reach}}(\mathcal{X}_{t-1}^-) \cap \bar{R}_{\text{ret}}(S_t^-, \mathcal{X}_{t-1}^-)\}.$$

Predicted optimistic safe space. As defined in [Wachi et al. \(2018\)](#) and [Turchetta et al. \(2019\)](#), an *optimistic* safe space has rich information for inferring the safety function. Let \mathcal{X}_t^+ denote the predicted optimistic safe space. Similarly to \mathcal{X}_t^- , the *optimistic* safe space, \mathcal{X}_t^+ , is defined as

$$\mathcal{X}_t^+ = \{s \in S_t^+ \mid s \in R_{\text{reach}}(\mathcal{X}_{t-1}^+) \cap \bar{R}_{\text{ret}}(S_t^+, \mathcal{X}_{t-1}^+)\},$$

where S_t^+ is the set of states that may satisfy the safety constraint, which is written as

$$S_t^+ = \{s \in \mathcal{S} \mid \exists s' \in \mathcal{X}_{t-1}^+ : u_t(s') - L \cdot d(s, s') \geq h\}.$$

Intuitively, \mathcal{X}_t^+ contains all states that may turn out to be safe even if the probability is low. In other words, $\mathcal{S} \setminus \mathcal{X}_t^+$ contains states that are unsafe with high probability.

Confidence interval. The correctness of \mathcal{X}_t^+ and \mathcal{X}_t^- depends on the accuracy of the confidence interval inferred by the GP. The *conservativeness* can be tuned by using the parameter β , and the choice of this parameter was well-studied in [Srinivas et al. \(2010\)](#) and [Chowdhury & Gopalan \(2017\)](#).

In the rest of this paper, we set the parameter to

$$\beta_t = B^g + \sigma_g \sqrt{2(\Gamma_{t-1}^g + 1 + \log(1/\Delta^g))},$$

where B^g is a bound on the RKHS norm of g , Δ^g is the allowed failure probability, and the observation noise is σ_g -sub-Gaussian. Also, Γ^g quantifies the effective degrees of freedom associated with the kernel function, which represents the maximal mutual information that can be obtained about the GP prior.

Under the above definitions and assumptions, we have the following lemma regarding the correctness of the confidence intervals.

Lemma 1. Assume that $\|g\|_k^2 \leq B^g$ and $n_t^g \leq \sigma_g$, $\forall t \geq 1$. If $\beta_t = B^g + \sigma_g \sqrt{2(\Gamma_{t-1}^g + 1 + \log(1/\Delta^g))}$, then

$$|g(s) - \mu_{t-1}^g(s)| \leq \beta_t^{1/2} \sigma_{t-1}^g(s)$$

holds for all $t \geq 1$ with probability at least $1 - \Delta^g$.

The above paradigm for the safety function can also be applied to the reward function. Hence, we have a similar lemma for the reward function as well.

Lemma 2. Assume that $\|r\|_k^2 \leq B^r$ and $n_t^r \leq \sigma_r$, $\forall t \geq 1$. If $\alpha_t = B^r + \sigma_r \sqrt{2(\Gamma_{t-1}^r + 1 + \log(1/\Delta^r))}$, then

$$|r(s) - \mu_{t-1}^r(s)| \leq \alpha_t^{1/2} \sigma_{t-1}^r(s)$$

holds for all $t \geq 1$ with probability at least $1 - \Delta^r$.

These lemmas follow from Theorem 2 in Chowdhury & Gopalan (2017).

Optimal solution. Here, we define the optimal policy in our problem setting. Under the optimal policy, π^* , the value function, $V_{\mathcal{M}}$, satisfies the following Bellman equation:

$$V_{\mathcal{M}}^*(s_t) = \max_{s_{t+1} \in \bar{R}_{\epsilon_g}(S_0)} [r(s_{t+1}) + \gamma V_{\mathcal{M}}^*(s_{t+1})],$$

where $\bar{R}_{\epsilon_g}(S_0)$ is the largest set that can be safely learned up to ϵ_g accuracy (for a formal definition, see Appendix A or Turchetta et al. (2016)). In our problem setting, in which the reward and safety functions are unknown a priori, the above Bellman equation cannot be solved directly. Our ultimate objective is to obtain a policy whose value is close to $V_{\mathcal{M}}^*$ while guaranteeing satisfaction of the safety constraint.

4. Algorithm

We now introduce our proposed algorithm, SNO-MDP, for achieving a near-optimal policy with respect to the cumulative reward while guaranteeing safety.

Algorithm 1 SNO-MDP with ES²

Input: states \mathcal{S} , actions \mathcal{A} , transition function f , kernel k^r for reward, kernel k^g for safety, GP prior for reward, GP prior for safety, safety threshold h , discount factor γ , Lipschitz constant L , initial safe space S_0 .

```

1:  $C_0(s) \leftarrow [h, \infty)$  for all  $s \in S_0$ 
2: // Exploration of safety
3: loop
4:    $S_t^- \leftarrow \{s \in \mathcal{S} \mid \exists s' \in \mathcal{X}_{t-1}^- : l_t(s') - L \cdot d(s, s') \geq h\}$ 
5:    $S_t^+ \leftarrow \{s \in \mathcal{S} \mid \exists s' \in \mathcal{X}_{t-1}^+ : u_t(s') - L \cdot d(s, s') \geq h\}$ 
6:    $\mathcal{X}_t^- \leftarrow \{s \in S_t^- \mid s \in R_{\text{reach}}(\mathcal{X}_{t-1}^-) \cap \bar{R}_{\text{ret}}(S_t^-, \mathcal{X}_{t-1}^-)\}$ 
7:    $\mathcal{X}_t^+ \leftarrow \{s \in S_t^+ \mid s \in R_{\text{reach}}(\mathcal{X}_{t-1}^+) \cap \bar{R}_{\text{ret}}(S_t^+, \mathcal{X}_{t-1}^+)\}$ 
8:    $G_t \leftarrow \{s \in \mathcal{X}_t^- \mid e_t(s) > 0\}$ 
9:    $\xi \leftarrow \arg \max_{s \in G_t} w_t(s)$ 
10:  Update GPs for both reward and safety on way to  $\xi$ 
11:   $t \leftarrow t + T_{s_{t-1} \rightarrow \xi}$  and  $s_t \leftarrow \xi$ 
12:  // ES2 algorithm
13:   $\mathcal{Y}_t \leftarrow \{s' \in \mathcal{S}^+ \mid \forall s \in \mathcal{X}_t^- : s' = f(s, \pi_y^*(a \mid s))\}$ 
14:  if  $\mathcal{Y}_t \subseteq \mathcal{X}_t^-$  then break
15:  // Typical termination condition
16:  if  $\max_{s \in G_t} w_t(s) < \epsilon_g$  then break
17: end loop
18: // Exploration and exploitation of reward
19: loop
20:    $U_t \leftarrow \mu_t^r + \alpha_{t+1} \cdot \sigma_t^r$ 
21:    $J_{\mathcal{Y}}^*(s_t) \leftarrow \max_{s_{t+1} \in \mathcal{Y}_t} [U_t(s_{t+1}) + \gamma J_{\mathcal{Y}}^*(s_{t+1})]$ 
22:    $s_{t+1} \leftarrow \arg \max_{s_{t+1} \in \mathcal{Y}_t} J_{\mathcal{Y}}^*(s_t)$ 
23: end loop

```

We first give an overview of SNO-MDP, which is outlined as Algorithm 1. We extend a stepwise approach in Sui et al. (2018) from state-less to stateful settings. Basically, our algorithm consists of two steps. In the first step, the agent expands the pessimistic safe region while guaranteeing safety (lines 2–17). Next, it explores and exploits the reward in the safe region certified in the first step (lines 18–23). The reason for this stepwise approach is that we can neglect uncertainty related to the a priori unknown safety function once the safe region is fixed.

However, a pure stepwise approach does not stop exploring the safe region until the convergence of the GP confidence interval (lines 15–16). This formulation often requires the agent to execute a great number of actions for exploring safety. Hence, to achieve near-optimality while executing a smaller number of actions, we also propose the ES² algorithm.² This algorithm checks whether the current safe region is sufficient for achieving near-optimality (lines 12–14), which maintains the theoretical guarantee with respect to both the satisfaction of the safety constraint and the near-optimality of the cumulative reward. We further propose a practical ES² algorithm, called P-ES², with better empirical performance, although it does not provide a theoretical guarantee in terms of the near-optimality of the cumulative reward.

²Both ES² and P-ES² do not affect the agent's safety.

4.1. Exploration of Safety (Step 1)

First, we consider how to explore the safety function. As a scheme to expand the safe region, we consider “expanders” as in Sui et al. (2015) and Turchetta et al. (2016). Expanders are states that may expand the predicted safe region, which is defined as $G_t = \{s \in \mathcal{X}_t^- \mid e_t(s) > 0\}$, where $e_t(s) = |s' \in \mathcal{S} \setminus \mathcal{S}_t^- \mid u_t(s) - Ld(s, s') \geq h|$.

The *efficiency* of expanding the safe region is measured by the width of the safety function’s confidence interval, defined as

$$w_t(s) = u_t(s) - l_t(s).$$

The agent safely and efficiently expands the safe region by sampling the state with the maximum value of w among the expanders, G_t . Hence, the agent sets the temporal goal according to

$$\xi = \arg \max_{s \in G_t} w_t(s).$$

Then, within the predicted safe space \mathcal{X}_t^- , it chooses a path to get to ξ from the current state s_{t-1} so as to minimize the cost (e.g., the path length). In our experiment, we simply minimized the path length. By defining the cost as related to w (e.g., $1/w$), however, the agent could explore safety more actively on the way to ξ .

The previous work (Sui et al., 2015; Turchetta et al., 2016; Sui et al., 2018) terminated safety exploration when the desired accuracy was achieved for every state in G_t ; that is,

$$\max_{s \in G_t} w_t(s) \leq \epsilon_g. \quad (1)$$

Unfortunately, this termination condition often requires a great number of iterations. For the purpose of maximizing the cumulative reward, it often leads to the loss of reward. Therefore, in Section 4.3, we propose the ES² algorithm to improve this point.

4.2. Exploration and Exploitation of Reward (Step 2)

Once expansion of the safe region is completed, the agent guarantees safety as long as it is in \mathcal{X}^- and does not have to expand the safe region anymore. Hence, all we have to do is optimize the cumulative reward in \mathcal{X}^- . As such, a simple approach is to follow the *optimism in the face of uncertainty* principle as in Strehl & Littman (2008) and Auer & Ortner (2007), then to consider the “exploration bonus” represented by R-MAX (Brafman & Tennenholtz, 2002) and Bayesian Exploration Bonus (BEB, Kolter & Ng (2009)).

Specifically, in accordance with Lemma 2, we optimize the policy by *optimistically* measuring the reward with the (probabilistic) upper confidence bound,

$$U_t(s) := \mu_t^r(s) + \alpha_{t+1}^{1/2} \cdot \sigma_t^r(s).$$

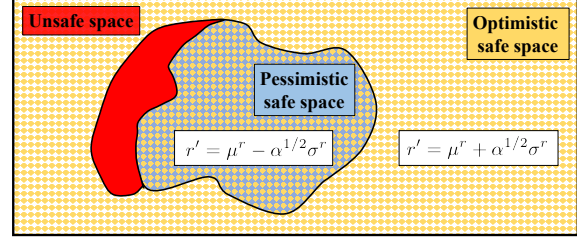


Figure 1. Illustration of \mathcal{M}_y , used in the ES² algorithm. The yellow and blue regions represent \mathcal{X}_t^+ and \mathcal{X}_t^- , respectively. The red region (i.e., $\mathcal{X} \setminus \mathcal{X}_t^+$) is unsafe with high probability.

In this reward setting, the second term on the right-hand side corresponds to the exploration bonus. For balancing the exploration and exploitation in terms of reward, we solve the following Bellman equation:

$$J_{\mathcal{X}}^*(s_t, \mathbf{b}_t^r, \mathbf{b}_t^g) = \max_{s_{t+1} \in \mathcal{X}_{t+1}^-} [U_t(s_{t+1}) + \gamma J_{\mathcal{X}}^*(s_{t+1}, \mathbf{b}_t^r, \mathbf{b}_t^g)],$$

where $\mathbf{b}^r = (\mu^r, \sigma^r)$ and $\mathbf{b}^g = (\mu^g, \sigma^g)$ are the beliefs over reward and safety, respectively. Also, t^* is the time step when the termination condition (1) is satisfied. Note that \mathbf{b}^r and \mathbf{b}^g are not updated; hence, we can solve the above equation with standard algorithms (e.g., value iteration).

4.3. Early Stopping of Exploration of Safety (ES²)

We have proposed a stepwise approach for exploring and optimizing the constrained MDP. In the first step when the safe region is expanded, however, the existing safe exploration algorithms (Sui et al., 2015; Turchetta et al., 2016; Sui et al., 2018) continue exploring the state space until convergence of the confidence interval, w , which generally leads to a large number of iterations. Our primary objective is to maximize the cumulative reward; hence, we should stop exploring safety if further exploration will not lead to maximizing the cumulative reward.

While exploring the safety function, we check whether the step can be migrated. As such, we consider the following additional MDP,

$$\mathcal{M}_y = \langle \mathcal{X}^+, \mathcal{A}, f, r', g, \gamma \rangle.$$

As shown in Figure 1, the differences from the original MDP, \mathcal{M} , lie in the state space and the reward function. The state space of \mathcal{M}_y is defined as the optimistic safe space (i.e., \mathcal{X}^+), while the reward function is defined as follows:

$$r' := \begin{cases} \mu^r + \alpha^{1/2} \sigma^r & \text{if } s \in \mathcal{X}_t^+ \setminus \mathcal{X}_t^-, \\ \mu^r - \alpha^{1/2} \sigma^r & \text{if } s \in \mathcal{X}_t^-. \end{cases} \quad (2)$$

In the pessimistic safe space, the reward is defined as the lower bound; otherwise, it is defined as the upper bound. This definition of the reward function encourages the agent

to explore outside the predicted safe space, \mathcal{X}_t^- . Using the new MDP above, we consider the set of states that the agent will visit at the next time step, defined as

$$\mathcal{Y}_t = \{s' \in \mathcal{S}^+ \mid \forall s \in \mathcal{X}_t^- : s' = f(s, \pi_y^*(a \mid s))\},$$

where π_y^* is the optimal policy for \mathcal{M}_y , obtained by maximizing the following value function:

$$V_{\mathcal{M}_y}(s_t) = \max_{s_{t+1} \in \mathcal{X}_t^+} [r'(s_{t+1}) + \gamma V_{\mathcal{M}_y}(s_{t+1})]. \quad (3)$$

Finally, we stop exploring the safety function if the following equation holds:

$$\mathcal{Y}_t \subseteq \mathcal{X}_t^-. \quad (4)$$

Intuitively, we stop expanding the safe space if the direction of the optimal policy for \mathcal{M}_y heads for the inside of \mathcal{X}_t^- . If the agent tries to stay in \mathcal{X}_t^- even under the condition that the reward is defined as in (2), then we do not have to expand the safe region anymore.

When the ES² algorithm confirms satisfaction of the above condition, we move on to the next step and then optimize the cumulative reward in \mathcal{Y}_t ; that is,

$$J_{\mathcal{Y}}^*(s_t, \mathbf{b}_t^r, \mathbf{b}_t^g) = \max_{s_{t+1} \in \mathcal{Y}_t} [U_t(s_{t+1}) + \gamma J_{\mathcal{Y}}^*(s_{t+1}, \mathbf{b}_t^r, \mathbf{b}_t^g)].$$

4.4. More Practical ES² Algorithm (P-ES²)

As we will prove in Section 5, the ES² algorithm provides us with a theoretical guarantee with respect to the cumulative reward. Unfortunately, this guarantee is achieved at the expense of empirical performance. The issue with the pure ES² algorithm lies in the state constraint in (3); that is, the value function is calculated under the assumption that all the states in the *optimistic* safe space, \mathcal{X}^+ , will be identified as safe. This assumption is necessary for the theoretical guarantee, but, in practice, it would be more reasonable to measure the probability of a state being identified as safe. Because the safety function is inferred as a Gaussian distribution for each state, an example of such a probability is a complementary error function; that is, we define the following probability,

$$p(s, \mathbf{b}^g) = \Pr[g(s) \geq h \mid \mathbf{b}^g] \approx 1 - \frac{1}{2} \operatorname{erfc} \left(\frac{\mu^g(s) - h}{\sqrt{2}\sigma^g(s)} \right).$$

Here, we introduce a new *virtual* state, z . Concretely, for z , the reward and transition probability are defined as $r(z) = 0$ and $P(z \mid z, a, \mathbf{b}^g) = 1$ for all a and \mathbf{b}^g , respectively. Hence, using z , we define a virtual transition probability $P_x^z = \Pr[x \mid s_t, a_t, \mathbf{b}_t^g]$ as follows:

$$P_x^z := \begin{cases} p(s_{t+1}, \mathbf{b}_t^g) & \text{if } x = s_{t+1}, \\ 1 - p(s_{t+1}, \mathbf{b}_t^g) & \text{if } x = z. \end{cases}$$

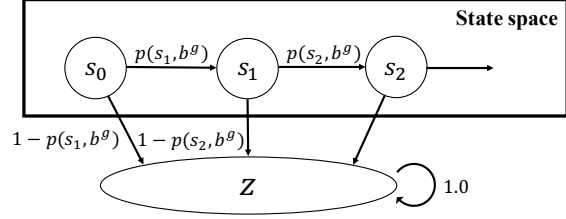


Figure 2. Illustration of \mathcal{M}_z . This MDP is characterized by the virtual state z and the virtual transition probability P^z .

Hence, by introducing z and P^z , we define the following MDP with the smooth, continuous transition probability:

$$\mathcal{M}_z = \langle \mathcal{X}^+ \cup \{z\}, \mathcal{A}, P^z, r', g, \gamma \rangle.$$

Figure 2 shows a conceptual image of this MDP. Intuitively, in \mathcal{M}_z , the agent optimizes the policy under the virtual condition that a state-action pair may lead to the extremely undesirable state, z , with probability $1 - p$. Then, we solve the following equation instead of solving (3):

$$V_{\mathcal{M}_z}(s_t) = \max_{s_{t+1} \in \mathcal{X}_t^+} [P_{s_{t+1}}^z \cdot \{r'(s_{t+1}) + \gamma V_{\mathcal{M}_z}(s_{t+1})\}].$$

For this equation, we used $r(z) = 0$ and $V(z) = 0$. For the optimal policy π_z^* obtained by solving the above equation, we stop exploring the safety function if the following equation holds:

$$\mathcal{Z}_t := \{s' \in \mathcal{S}^+ \mid \forall s \in \mathcal{X}_t^- : s' = f(s, \pi_z^*(a \mid s))\} \subseteq \mathcal{X}_t^-.$$

Then, we optimize the cumulative reward in \mathcal{Z}_t by solving the following equation:

$$J_{\mathcal{Z}}^*(s_t, \mathbf{b}_t^r, \mathbf{b}_t^g) = \max_{s_{t+1} \in \mathcal{Z}_t} [U_t(s_{t+1}) + \gamma J_{\mathcal{Z}}^*(s_{t+1}, \mathbf{b}_t^r, \mathbf{b}_t^g)].$$

5. Theoretical Results

We now provide theoretical guarantees on the safety and near-optimality of our proposed algorithm. Theorem 1 is associated with the safe expansion stage (i.e., step 1), which guarantees safety and convergence to the safe region. Theorem 2 ensures convergence toward the near-optimal cumulative reward. Theorem 3 ensures that SNO-MDP still achieves the near-optimal cumulative reward even when the ES² algorithm is used. In the rest of this paper, let $V_{\max} = R_{\max}/(1-\gamma)$. Also, let $D : \mathcal{M} \rightarrow \mathbb{R}$ be a diameter of an MDP, defined as $D(\mathcal{M}) = \min_{\pi} \max_{s^1 \neq s^2} T_{s^1 \rightarrow s^2}^{\pi}$, where $T_{s^1 \rightarrow s^2}^{\pi}$ is the expected number of time steps that policy π takes to move from s^1 to s^2 .

5.1. Safety Guarantee and Completeness

We first present a theorem related to the safety guarantee and completeness.

Theorem 1. Assume that the safety function g satisfies $\|g\|_k^2 \leq B^g$ and is L -Lipschitz continuous. Also, assume that $S_0 \neq \emptyset$ and $g(s) \geq h$ for all $s \in S_0$. Fix any $\epsilon_g > 0$ and $\Delta^g \in (0, 1)$. Suppose that we conduct the stage of “exploration of safety” with the noise n_t^g being σ_g -sub-Gaussian, and that $\beta_t = B^g + \sigma_g \sqrt{2(\Gamma_{t-1}^g + 1 + \log(1/\Delta^g))}$ until $\max_{s \in G_t} w_t(s) < \epsilon_g$ is achieved. Finally, let t^* be the smallest integer satisfying

$$\frac{t^*}{\beta_{t^*} \Gamma_{t^*}^g} \geq \frac{C_g |\bar{R}_0(S_0)|}{\epsilon_g^2} D(\mathcal{M}),$$

with $C_g = 8/\log(1 + \sigma_g^{-2})$. Then, the following statements jointly hold with probability at least $1 - \Delta^g$:

- $\forall t \geq 1, g(s_t) \geq h$,
- $\exists t_0 \leq t^*, \bar{R}_{\epsilon_g}(S_0) \subseteq \mathcal{X}_{t_0}^- \subseteq \bar{R}_0(S_0)$.

A proof is presented in the supplemental material. Theorem 1 guarantees that SNO-MDP is safe in the stage of exploration of safety (i.e., step 1), as well as in the stage of optimization of reward (i.e., step 2), with high probability. In addition, after a sufficiently large number of time steps, \mathcal{X}^- is guaranteed to be a super-set of $\bar{R}_{\epsilon_g}(S_0)$.

5.2. Near-Optimality

We next present a theorem on the near-optimality with respect to the cumulative reward.

Theorem 2. Assume that the reward function r satisfies $\|r\|_k^2 \leq B^r$, and that the noise is σ_r -sub-Gaussian. Let π_t denote the policy followed by SNO-MDP at time t , and let s_t and $\mathbf{b}_t^r, \mathbf{b}_t^g$ be the corresponding state and beliefs, respectively. Let t^* be the smallest integer satisfying $\frac{t^*}{\beta_{t^*} \Gamma_{t^*}^g} \geq \frac{C_g |\bar{R}_0(S_0)|}{\epsilon_g^2} D(\mathcal{M})$, and fix any $\Delta^r \in (0, 1)$. Finally, set $\alpha_t = B^r + \sigma_r \sqrt{2(\Gamma_{t-1}^r + 1 + \log(1/\Delta^r))}$ and

$$\epsilon_V^* = V_{\max} \cdot (\Delta^g + \Sigma_{t^*}^r / R_{\max}),$$

with $\Sigma_{t^*}^r = \frac{1}{2} \sqrt{\frac{C_r \alpha_{t^*} \Gamma_{t^*}^r}{t^*}}$. Then, with high probability,

$$V^{\pi_t}(s_t, \mathbf{b}_t^r, \mathbf{b}_t^g) \geq V^*(s_t) - \epsilon_V^*$$

— i.e., the algorithm is ϵ_V^* -close to the optimal policy — for all but t^* time steps, while guaranteeing safety with probability at least $1 - \Delta^g$.

A detailed proof of Theorem 2 is presented in the supplemental material. The proof is based on the following idea. After the agent fully explores the safe space, \mathcal{X}^- satisfies $\bar{R}_{\epsilon_g}(S_0) \subseteq \mathcal{X}^- \subseteq \bar{R}_0(S_0)$, and states in \mathcal{X}^- are safe with high probability. Once \mathcal{X}^- converges, the probability of leaving the “known” safe space is small; hence, Theorem 2

follows by adapting standard arguments from previous PAC-MDP results. The key condition that allows us to prove the near-optimality of SNO-MDP is that, at every time step, the agent is optimistic with respect to the reward, and this optimism decays given a sufficient number of samples. By optimizing the cumulative reward in \mathcal{X}^- according to the optimism in the face of uncertainty principle, the acquired policy is ϵ_V^* -close to the optimal policy in the original safety-constrained MDP.

Finally, we present a theoretical result related to the ES² algorithm. Specifically, we prove that ES² maintains the near-optimality of SNO-MDP.

Theorem 3. Assume that the reward function r satisfies $\|r\|_k^2 \leq B^r$, and that the noise is σ_r -sub-Gaussian. Let π_t denote the policy followed by SNO-MDP with the ES² algorithm at time t , and let s_t and $\mathbf{b}_t^r, \mathbf{b}_t^g$ be the corresponding state and beliefs, respectively. Let \tilde{t} be the smallest integer for which (4) holds, and fix any $\Delta^r \in (0, 1)$. Finally, set $\alpha_t = B^r + \sigma_r \sqrt{2(\Gamma_{t-1}^r + 1 + \log(1/\Delta^r))}$ and

$$\tilde{\epsilon}_V = V_{\max} \cdot (\Delta^g + \Sigma_{\tilde{t}}^r / R_{\max}),$$

with $\Sigma_{\tilde{t}}^r = \frac{1}{2} \sqrt{\frac{C_r \alpha_{\tilde{t}} \Gamma_{\tilde{t}}^r}{\tilde{t}}}$. Then, with high probability,

$$V^{\pi_t}(s_t, \mathbf{b}_t^r, \mathbf{b}_t^g) \geq V^*(s_t) - \tilde{\epsilon}_V$$

— i.e., the algorithm is $\tilde{\epsilon}_V$ -close to the optimal policy — for all but \tilde{t} time steps while guaranteeing safety with probability at least $1 - \Delta^g$.

The proof of Theorem 3 is presented in the supplemental material. The proof is based on the following idea. When the condition in (4) is satisfied, the agent will not leave \mathcal{Y} , and a near-optimal policy is obtained by optimizing the cumulative reward only in \mathcal{Y} with the optimistically measured reward. Also, as long as the agent is in $\mathcal{Y} (\subseteq \mathcal{X}^-)$, safety is guaranteed with high probability. The proof is similar to that for Theorem 2.

6. Experiment

In this section, we evaluate the performance of SNO-MDP through two experiments. One used a synthetic environment, while the other simulated Mars surface exploration. We also show the effectiveness of our ES² and P-ES² algorithms.

6.1. Synthetic GP-SAFETY-GYM Environment

Settings. We constructed a new open-source environment for safe RL simulations named GP-SAFETY-GYM. This environment was built based on OpenAI Safety-Gym (Ray et al., 2019). As shown in Figure 3(a), GP-SAFETY-GYM represents the reward by a color (yellow: high; green: medium; blue: low), and the safety by height.

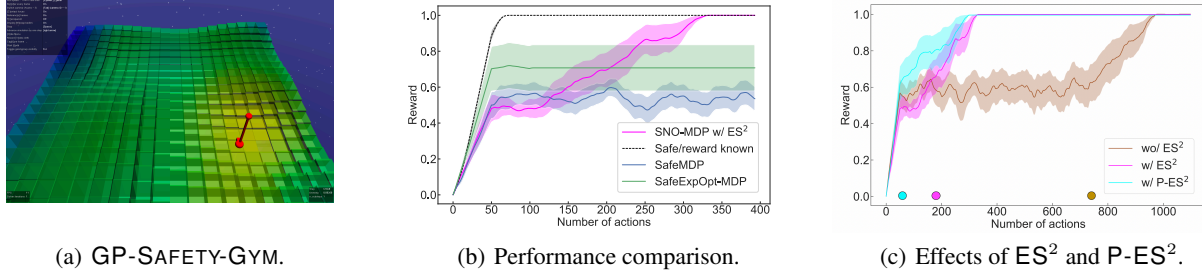


Figure 3. Experiment with synthetic data. (a) Example screen capture from the GP-SAFETY-GYM environment. (b) Average reward over the episodes, comparing the performance of SNO-MDP with ES^2 and the baselines. (c) Average reward over the episodes, showing the effects of ES^2 and P- ES^2 . The colored circles represent when the transition from safe exploration to reward optimization happens for each method. In both (b) and (c), the reward is normalized with respect to the SAFE/REWARD KNOWN case.

We considered a 20×20 square grid in which the reward and safety functions were randomly generated. At every time step, an agent chose an action from *stay*, *up*, *right*, *down*, and *left*. The agent predicted the reward and safety functions by using different kernels on the basis of previous observations. In this simulation, we allowed the agent to observe the reward and safety function values of the current state and neighboring states. The kernel for reward was a radial basis function (RBF) with the length-scales of 2 and prior variance of 1. The kernel for safety was also an RBF with the length-scales of 2 and prior variance of 1. Finally, we set the discount factor to $\gamma = 0.99$, and confidence intervals parameters to $\alpha_t = 3$ and $\beta_t = 2$ for all $t \geq 1$.

Baselines. We empirically compared the performance of our SNO-MDP with SAFEMDP (Turchetta et al., 2016) and SAFEEXPOPT-MDP (Wachi et al., 2018), as well as a case called SAFE/REWARD KNOWN. In SAFEMDP, the agent tries to expand the safe region without considering the reward. In SAFEEXPOPT-MDP, the agent attempts to maximize the cumulative reward while leveraging the difference between the value function in \mathcal{X}_t^+ and that in \mathcal{X}_t^- . Finally, SAFE/REWARD KNOWN is a non-exploratory case in which the safety and reward functions are known a priori.

Metrics. We used the cumulative reward and the number of unsafe actions as comparison metrics.

Results. Figure 3(b) compares the performance of SNO-MDP and the baselines in terms of the reward. For these results, the average reward was measured over the previous 50 time steps. SNO-MDP achieved the optimal reward after shifting to the stage of reward optimization, which outperforms SAFEMDP and SAFEEXPOPT-MDP in terms of the reward after a sufficiently large number of time steps. The SAFEMDP agent did not aim to maximize the cumulative reward, and the SAFEEXPOPT-MDP agent was sometimes

stucked in a local optimum when the expansion of the safe region was insufficient. Figure 3(c) shows the empirical performance of the ES^2 and P- ES^2 algorithms. P- ES^2 achieved faster convergence in terms of the reward than the original ES^2 did. Also, all methods, including the baselines, did *not* take any unsafe actions.

6.2. Simulated Mars Surface Exploration

Settings. We then conducted an experiment based on a Mars surface exploration scenario, as in Turchetta et al. (2016) and Wachi et al. (2018). In this simulation, we used a publicly available Mars digital elevation model (DEM) that was created from observation data captured by the high-resolution imaging science experiment (HiRISE) camera (McEwen et al., 2007).

We created a 40×30 rectangular grid-world by clipping a region around latitude $30^\circ 6'$ south and longitude $202^\circ 2'$ east, as shown in Figure 4. At every time step, the rover took one of five actions: *stay*, *up*, *down*, *left*, and *right*. We assumed that any state in which the slope angle was greater than 25° were unsafe. The safety function g was defined as the slope angle calculated from the DEM, and the safety threshold was $h = -\tan(25^\circ)$.

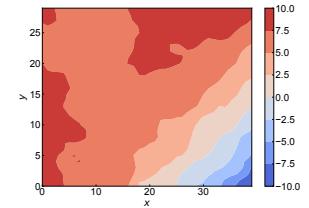


Figure 4. Mars terrain data.

The rover predicted the elevation by using a GP with a Matérn kernel with $\nu = 5/2$. The length-scales were 15 m, and the prior variance over elevation was 100 m^2 . We assumed a noise standard deviation of 0.075 m. For the reward, we randomly defined a smooth, continuous reward. To predict the reward function, the rover used a GP with RBF kernel having length-scales of 2 and a prior variance

Table 1. Experimental results with real Mars data.

| | REWARD | UNSAFE ACTIONS |
|------------------------------|-------------|----------------|
| SNO-MDP w/ P-ES ² | 0.81 | 0 |
| SNO-MDP w/ ES ² | 0.78 | 0 |
| SNO-MDP | 0.49 | 0 |
| SAFEMDP | 0.34 | 0 |
| SAFEEXPOPT-MDP | 0.59 | 0 |
| SAFE/REWARD KNOWN | 1.00 | 0 |

over the reward of 2. We set the confidence levels as $\alpha_t = 3$ and $\beta_t = 2, \forall t \geq 0$, and the discount factor as $\gamma = 0.9$.

Baselines and metrics. We used the same baselines and metrics as in the previous synthetic experiment.

Results. Table 1 summarizes the results. The reward was accumulated over the episode, which was normalized with respect to the SAFE/REWARD KNOWN case. Our SNO-MDP with either P-ES² or ES² outperformed SAFEMDP and SAFEEXPOPT-MDP in terms of the reward. This was expected, because SafeMDP does not aim to maximize the cumulative reward, and SAFEEXPOPT-MDP does not guarantee the near-optimality of the cumulative reward. Also, *no* unsafe action was executed by any of the tested algorithms.

7. Conclusion

We have proposed SNO-MDP, a stepwise approach for exploring and optimizing a safety-constrained MDP. Theoretically, we proved a bound of the sample complexity to achieve ϵ_V -closeness to the optimal policy while guaranteeing safety, with high probability. We also proposed the ES² and P-ES² algorithms for improving the efficiency in obtaining rewards. We developed an open-source environment, GP-SAFETY-GYM, to test the effectiveness of SNO-MDP. We also demonstrated the advantages of SNO-MDP using the real Mars terrain data.

Acknowledgement

This work is sponsored by IBM Research AI and Toyota-Tsinghua joint project on health challenges for future city and smart health system design (20203910025).

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*, 2017.

Araya, M., Buffet, O., and Thomas, V. Near-optimal BRL using optimistic local transitions. In *International Conference on Machine Learning (ICML)*, 2012.

Aswani, A., Gonzalez, H., Sastry, S. S., and Tomlin, C. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

Auer, P. and Ortner, R. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2007.

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. In *Neural Information Processing Systems (NeurIPS)*, 2017.

Brafman, R. I. and Tennenholtz, M. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research (JMLR)*, pp. 213–231, 2002.

Chow, Y., Nachum, O., Faust, A., Duenez-Guzman, E., and Ghavamzadeh, M. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2017.

Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 2018.

Garcia, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 16(1):1437–1480, 2015.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Kolter, J. Z. and Ng, A. Y. Near-Bayesian exploration in polynomial time. In *International Conference on Machine Learning (ICML)*, 2009.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Sokaert, P. O. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

McEwen, A. S., Eliason, E. M., Bergstrom, J. W., et al. Mars reconnaissance orbiter’s high resolution imaging science experiment (HiRISE). *Journal of Geophysical Research: Planets*, 112 (E5), 2007.

Rasmussen, C. E. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pp. 63–71. Springer, 2004.

Ray, A., Achiam, J., and Amodei, D. Benchmarking safe exploration in deep reinforcement learning. 2019.

Schölkopf, B. and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

- Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. PAC model-free reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2006.
- Sui, Y., Gotovos, A., Burdick, J. W., and Krause, A. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning (ICML)*, 2015.
- Sui, Y., Zhuang, V., Burdick, J. W., and Yue, Y. Stagewise safe Bayesian optimization with Gaussian processes. In *International Conference on Machine Learning (ICML)*, 2018.
- Turchetta, M., Berkenkamp, F., and Krause, A. Safe exploration in finite Markov decision processes with Gaussian processes. In *Neural Information Processing Systems (NeurIPS)*, 2016.
- Turchetta, M., Berkenkamp, F., and Krause, A. Safe exploration for interactive machine learning. In *Neural Information Processing Systems (NeurIPS)*, 2019.
- Wachi, A., Sui, Y., Yue, Y., and Ono, M. Safe exploration and optimization of constrained MDPs using Gaussian processes. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.