Anytime Safe Reinforcement Learning

Pol Mestres[1] POMESTRE@UCSD.EDU
Arnau Marzabal[1,2] AMARZABAL@UCSD.EDU
Jorge Cortés[1] CORTES@UCSD.EDU

[1]Department of Mechanical and Aerospace Engineering, University of California San Diego
[2]Centre de Formació Interdisciplinària Superior, Universitat Politècnica de Catalunya

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

This paper considers the problem of solving constrained reinforcement learning problems with anytime guarantees, meaning that the algorithmic solution returns a safe policy regardless of when it is terminated. Drawing inspiration from anytime constrained optimization, we introduce Reinforcement Learning-based Safe Gradient Flow (RL-SGF), an on-policy algorithm which employs estimates of the value functions and their respective gradients associated with the objective and safety constraints for the current policy, and updates the policy parameters by solving a convex quadratically constrained quadratic program. We show that if the estimates are computed with a sufficiently large number of episodes (for which we provide an explicit bound), safe policies are updated to safe policies with a probability higher than a prescribed tolerance. We also show that iterates asymptotically converge to a neighborhood of a KKT point, whose size can be arbitrarily reduced by refining the estimates of the value function and their gradients. We illustrate the performance of RL-SGF in a navigation example.

Keywords: Safe reinforcement learning, constrained optimization, policy gradient methods.

1. Introduction

Reinforcement Learning (RL) seeks to find an optimal policy for an agent that interacts in a given environment through a process of trial and error. At every state, the agent chooses an action, after which it randomly transitions to a new state and obtains the corresponding reward. The optimal policy is that which maximizes a predefined long-horizon reward. Formally, this process is modeled as a Markov Decision Process (MDP) and has exhibited a lot of empirical success in a variety of application domains. However, in many safety-critical applications, such as autonomous driving, robotic manipulation, or frequency control of power systems, the process of trial and error executed by most RL algorithms can lead the agent towards unsafe regions, with potentially catastrophic consequences. This observation has spurred a lot of interest in the development of safe reinforcement learning algorithms, which seek to find the optimal policy and respect safety constraints.

LITERATURE REVIEW: The safe RL literature is vast and, in what follows, we focus on works which are most aligned with the present manuscript. For more exhaustive surveys on safe RL, we refer the reader to (Brunkel et al., 2022; Liu et al., 2021; Garcia and Fernandez, 2015; Gu et al., 2022). Safety constraints in RL are often expressed in the form of *cumulative constraints*, i.e., expected cumulative rewards that need to be kept below a certain threshold over certain time horizon (Altman, 1999; Achiam et al., 2017; Chow et al., 2017; Paternain et al., 2019). MDPs with such type of constraints are referred to as Constrained Markov Decision Processes (CMDPs). A popular approach to solve CMDPs are primal-dual methods (Paternain et al., 2023; Ding et al., 2020), which simultaneously perform a maximization step in the primal variable and a minimization step in the

dual variable and can be shown to converge to the optimal policy for finite state and action CMDPs for a special class of probability transition functions (Ding et al., 2021). In continuous state-action space, (Paternain et al., 2019) also provides a primal-dual scheme, and shows that if a non-convex unconstrained RL problem is solved at every iteration, the algorithm provably converges to the optimal policy. However, solving such unconstrained RL problem at every iteration is computationally intractable. Primal-dual algorithms do not guarantee the satisfaction of the safety constraints during the training process. Other works have employed primal-dual methods to guarantee safety during training, but are either limited to particular policy parameterizations (Zeng et al., 2022) or solve a relaxed version of the problem and hence introduce an optimality gap (Bai et al., 2022, 2023). On the other hand, (Achiam et al., 2017) proposes CPO, an algorithm purely based on primal variable updates that enjoys safety guarantees at every iteration. However, since the exact policy update law is computationally intensive, the work provides a practical algorithm based on a first-order approximations of the objective and constraints that might not satisfy the safety constraints during training. Alternatively, (Liu et al., 2020) presents IPO, another primal method that adds the constraints as penalty terms in the objective function, and also guarantees the satisfaction of safety constraints during training. However, it requires a feasible initial policy and does not possess formal convergence guarantees. Other primal methods such as (Chow et al., 2018) leverage Lyapunov functions to guarantee the satisfaction of constraints during training. However, the method proposed to search for such Lyapunov functions might be computationally intensive, and convergence guarantees are only given for a limited class of problems. Finally, (Suttle et al., 2024) optimizes over a class of truncated policies so that unsafe actions have probability zero, but the restriction to such class of policies also introduces an optimality gap, which is not formally quantified.

STATEMENT OF CONTRIBUTIONS: We consider the problem of designing an algorithm that finds the optimal policy of a constrained RL problem within a parametric family while satisfying the safety constraints at every iterate. Inspired by recent advances in anytime constrained optimization, we introduce Reinforcement Learning-based Safe Gradient Flow (RL-SGF). At every iteration, RL-SGF obtains estimates of the value functions and their respective gradients associated with the objective function and safety constraint of the constrained RL problem. Next, RL-SGF updates the policy parameters by solving a convex quadratically constrained quadratic program that uses such estimates, and has a closed-form solution. We formally show that RL-SGF meets the desired specifications. First, we establish two results of independent interest that quantify the statistical properties of our proposed estimates, including a bound on their variance and the probability that the distance between the estimates and the true values is within a tolerance. Next, we show that for any prescribed confidence, if the estimates of the value functions and their gradients are generated with a sufficiently large number of episodes (with an explicitly computable lower bound), the next iterate of a safe policy under RL-SGF is a safe policy with a probability higher than the prescribed confidence. Moreover, we show that RL-SGF asymptotically converges to a neighborhood of a KKT point and that the size of this neighborhood can be made arbitrarily small by using a sufficiently large number of episodes. Finally, we illustrate the performance of RL-SGF in a navigation example. All proofs are included in the extended version (Mestres et al., 2024)[1].

1. Throughout the paper, we use the following notation. We denote by $\mathbb{Z}_{>0}$, $\mathbb{R}$, and $\mathbb{R}_{\geq 0}$ the set of positive integers, real, and nonnegative real numbers, respectively. Given $N \in \mathbb{Z}_{>0}$, we let $[N] = \{1, 2, \ldots, N\}$. For $x \in \mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. We let $\mathbf{I}_n$ be the n-dimensional identity matrix and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. For a set B and $x \in \mathbb{R}^n$, $\text{dist}(x, B) = \inf_{y \in B} \|x - y\|$. Given a random variables X taking scalar values, $\mathbb{E}[X]$ denotes the expectation of X, $\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$ denotes its variance. Let $\mathcal{S}$ be a set of states, $\mathcal{A}$ a set of actions, and $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ a probability transition function, where $P(s, a, s')$ represents the probability that the agent transitions to state $s' \in \mathcal{S}$ given that it is at state $s \in \mathcal{S}$ and takes action $a \in \mathcal{A}$. We further let $R_0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$

2. Problem Statement

Given a Constrained Markov Decision Process (CMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R_0, R_1)$, the goal of the agent is to maximize the cumulative rewards while keeping the cumulative costs below a certain threshold. We consider a parametric class of policies indexed by a vector $\theta \in \mathbb{R}^d$. We denote the policy associated with θ as π_θ. Given a distribution η of initial states, a discount factor $\gamma \in (0, 1)$, and a time horizon $T \in \mathbb{Z}_{>0}$, we consider the following problem:

$$\min_{\theta \in \mathbb{R}^d} \quad V_0(\theta) = \mathbb{E}_{a \sim \pi_\theta(\cdot|s), s_0 \sim \eta} \left[\sum_{k=0}^{T} -\gamma^k R_0(s_k, a_k, s_{k+1}) \right] \tag{1a}$$

$$\text{s.t.} \quad V_1(\theta) = \mathbb{E}_{a \sim \pi_\theta(\cdot|s), s_0 \sim \eta} \left[\sum_{k=0}^{T} \gamma^k R_1(s_k, a_k, s_{k+1}) \right] \leq 0. \tag{1b}$$

Problem (1) seeks to find the policy π_θ that maximizes the expected cumulative reward given by R_0 (note that, for convenience, we have changed the sign of R_0 to turn (1) into a minimization problem) over T time steps and keeps the expected cumulative reward given by R_1 over T time steps below zero. The discount factor γ determines how much future rewards are valued compared to immediate rewards. In general, both V_0 and V_1 are non-convex, which makes solving (1) NP-hard. Our goal is to develop an algorithm that converges to a KKT point of (1) and is anytime, meaning that at every iteration, the constraint (1b) is satisfied with a prescribed confidence.

3. RL-SGF: Safe RL via Anytime Constrained Optimization

Here we present the algorithm to solve (1) in an anytime fashion. Given $\alpha, h > 0$, at each iteration $i \in \mathbb{Z}_{>0}$, we consider the update law for parameter θ_i given by $\theta_{i+1} = p(\theta_i)$, where $p : \mathbb{R}^d \to \mathbb{R}^d$ is defined as:

$$p(\theta) = \arg\min_{y \in \mathbb{R}^d} \nabla V_0(\theta)^\top (y - \theta) + \frac{1}{2h} \|y - \theta\|^2 \tag{2a}$$

$$\text{s.t. } \alpha h V_1(\theta) + \nabla V_1(\theta)^\top (y - \theta) + \frac{1}{2h} \|y - \theta\|^2 \leq 0. \tag{2b}$$

Note that p is well defined over $\mathcal{F} = \{\theta \in \mathbb{R}^d \ : \ \exists y \in \mathbb{R}^d \text{ s.t. } \alpha h V_1(\theta) + \nabla V_1(\theta)^\top (y - \theta) + \frac{1}{2h} \|y - \theta\|^2 \leq 0\}$. The iterates defined through (2) are inspired by considering discretizations of the Safe Gradient Flow (SGF) (Allibhoy and Cortés, 2024), a continuous-time flow for constrained optimization whose trajectories remain feasible at all times if initialized in the feasible set. We refer the interested reader to (Allibhoy and Cortés, 2024) for intuitive expositions on the synthesis of this flow both from a control-theoretic perspective and from a projected dynamical systems perspective. We note that (2) is also a special case of the Moving Balls Algorithm (MBA), introduced in (Auslender et al., 2010). Both SGF and MBA are anytime algorithms, and (Allibhoy and Cortés, 2024; Auslender et al., 2010) provide conditions under which they converge to KKT points of the corresponding optimization problem. We summarize relevant properties of the update law p next:

and $R_1 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. For every $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $s' \in \mathcal{S}$, $R_0(s, a, s')$ is the reward associated with completing a task when an agent is at state s, takes action a, and transitions to state s'. Instead, $R_1(s, a, s')$ is the cost associated with a safety constraint when an agent is at state s, takes action a, and transitions to state s'. We refer to the tuple $(\mathcal{S}, \mathcal{A}, P, R_0, R_1)$ as a Constrained Markov Decision Process (CMDP). A policy π for the CMDP is a function that maps every state $s \in \mathcal{S}$ to a distribution over the set of actions $\mathcal{A}$. Such distribution is denoted as $\pi(\cdot|s)$, and $\pi(a|s)$ is the probability of taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$.

Lemma 1 (Constraint satisfaction, KKT points, and convergence): *Let V_0 and V_1 be Lipschitz on their domain of definition with Lipschitz constants L_0 and L_1 respectively, and assume V_0 is lower bounded. Let $\alpha > 0$ and $h \in (0, \min\{\frac{1}{\alpha}, \frac{1}{L_0}, \frac{1}{L_1}\})$. For $\theta \in \mathbb{R}^d$ with $V_1(\theta) \leq 0$, we have*

(i) $V_1(p(\theta)) \leq 0$,

(ii) *θ is a KKT point of (1) if and only if $p(\theta) = \theta$,*

(iii) *any limit point of the sequence $\{\theta_k\}_{k\in\mathbb{Z}_{>0}}$ defined as $\theta_0 = \theta$ and $\theta_k = p(\theta_{k-1})$ for $k \geq 1$ is a KKT point of (1).*

We note that Lemma 1 (i) ensures that the update rule $\theta_{i+1} = p(\theta_i)$ is anytime. The proof of Lemma 1 follows from (Auslender et al., 2010, Proposition 3.1), and we include it in Mestres et al. (2024) for completeness. Note also that $\{\theta \in \mathbb{R}^d \ : \ V_1(\theta) \leq 0\} \subset \mathcal{F}$, which ensures that p is well-defined for all safe policies. Although Lemma 1 ensures that the update rule $\theta_{i+1} = p(\theta_i)$ is anytime and leads to convergence to KKT points of (1), the main difficulty in computing $p(\theta)$ in (2) is that $V_1(\theta)$, $\nabla V_0(\theta)$, and $\nabla V_1(\theta)$ are unknown, and need to be estimated. Before introducing such estimates, we make the following assumptions.

Assumption 1 (Boundedness of reward functions): *There exist $B_0, B_1 > 0$ such that $|R_0(s, a, s')| < B_0$ and $|R_1(s, a, s')| < B_1$, for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $s' \in \mathcal{S}$.*

Assumption 2 (Differentiability and Lipschitzness of policy): *For any $a \in \mathcal{A}$, $s \in \mathcal{S}$, and $\theta \in \mathbb{R}^d$, $\pi_\theta(a|s) > 0$. The function $\Phi_{a,s} : \mathbb{R}^d \to \mathbb{R}$ defined as $\Phi_{a,s}(\theta) = \log \pi_\theta(a|s)$ is continuously differentiable and there exist positive constants L and $\tilde{B}$ such that*

$$\|\nabla\Phi_{a,s}(\theta_1) - \nabla\Phi_{a,s}(\theta_2)\| \leq L \|\theta_1 - \theta_2\|, \ \forall\, \theta_1, \theta_2 \in \mathbb{R}^d, a \in \mathcal{A}, s \in \mathcal{S},$$
$$\left\| \frac{\partial \Phi_{a,s}(\theta)}{\partial \theta_i} \right\| \leq \tilde{B}, \ \forall \theta \in \mathbb{R}^d, i \in [d], a \in \mathcal{A}, s \in \mathcal{S}.$$

Assumptions 1 and 2 are standard in the literature (cf. (Zhang et al., 2020; Bai et al., 2024)). By the Policy Gradient Theorem (Sutton and Barto, 2018, Section 13.2), under Assumption 2, the functions V_0 and V_1 in (1) are differentiable. Moreover, Lemma 7 in the extended version Mestres et al. (2024) ensures that V_0 and V_1 are globally Lipschitz, with Lipschitz constants that can be computed in terms of L, B_0, B_1, $\tilde{B}$, γ, and T. In the sequel, we let L_0 and L_1 be the Lipschitz constants of V_0 and V_1, respectively.

We are now ready to introduce unbiased estimates of $V_1(\theta)$, $\nabla V_0(\theta)$, and $\nabla V_1(\theta)$, based on the Policy Gradient Theorem (Sutton and Barto, 2018, Section 13.2). We also provide upper bounds on the variances of such estimates, and probabilistic upper bounds on the distance to their respective means. Throughout the paper, all probabilities, expectations and variances are taken with respect to $a \sim \pi_\theta(\cdot|s)$ and $s_0 \sim \eta$, and from now on we do not denote this explicitly for the sake of simplicity.

Lemma 2 (Statistical properties of estimates of the value functions and their gradients): *Suppose that Assumptions 1 and 2 hold. Let $\theta \in \mathbb{R}^d$, $N \in \mathbb{Z}_{>0}$, and consider N independent episodes of the form $[s_0^n, a_0^n, s_1^n, a_1^n, \ldots, s_T^n, a_T^n, s_{T+1}^n]$ generated with policy π_θ, where $s_0^n \sim \eta$ and $a_i^n \sim \pi_\theta(\cdot|s_i^n)$ for all $i \in [T]$ and $n \in [N]$. Let $q \in \{0, 1\}$ and define the value function estimator $\widehat{V_q}(\theta) = \frac{(-1)^{q+1}}{N} \sum_{n=1}^N \sum_{t=0}^T \gamma^t R_q(s_t^n, a_t^n, s_{t+1}^n)$. Given a function $b : \mathcal{S} \to \mathbb{R}$ (referred to as baseline) such that $|b(s)| \leq \hat{B}$ for all $s \in \mathcal{S}$, consider the value function gradient estimator $\widehat{\nabla V_q}(\theta) = \frac{(-1)^{q+1}}{N} \sum_{n=1}^N \sum_{t=0}^T \gamma^t \nabla\Phi_{a_t^n, s_t^n}(\theta) \sum_{t'=t}^T \left(\gamma^{t'-t} R_q(s_{t'}^n, a_{t'}^n, s_{t'+1}^n) - b(s_t^n)\right)$. Further define $\tilde{\sigma}_q = B_q \frac{1-\gamma^{T+1}}{1-\gamma}$ and $\bar{\sigma}_q = \tilde{B} \sum_{t=0}^T \gamma^t \sum_{t'=t}^T (B_q \gamma^{t'-t} + \hat{B})$ for $q \in \{0, 1\}$. Then,*

(i) The estimator $\widehat{V_q}(\theta)$ *is unbiased, i.e.,* $\mathbb{E}[\widehat{V_q}(\theta)] = V_q(\theta)$. *Moreover,* $\mathrm{Var}(\widehat{V_q}(\theta)) \leq \frac{\tilde{\sigma}_q^2}{N}$ *and for all* $\epsilon > 0$, $\mathbb{P}(|\widehat{V_q}(\theta) - V_q(\theta)| \leq \epsilon) \geq 1 - \exp\Big\{-\frac{N\epsilon^2}{2\tilde{\sigma}_q^2}\Big\}$.

(ii) The estimator $\widehat{\nabla V_q}(\theta)$ *is unbiased, i.e.,* $\mathbb{E}[\widehat{\nabla V_q}(\theta)] = \nabla V_q(\theta)$. *Moreover,* $\mathrm{Var}(\widehat{\nabla V_q}(\theta)_i) \leq \frac{\tilde{\sigma}_q^2}{N}$ *for all* $i \in [d]$, *and for all* $\epsilon > 0$, $\mathbb{P}\Big(\Big\|\nabla V_q(\theta) - \widehat{\nabla V_q}(\theta)\Big\| \leq \epsilon\Big) \geq 1 - d\exp\Big\{-\frac{N\epsilon^2}{2d\tilde{\sigma}_q^2}\Big\}$.

A bound on the variance similar to that of Lemma 2 (ii) appears in (Zhao et al., 2012) for Gaussian policies. Given the estimators introduced in Lemma 2, we propose the update law $\theta_{i+1} = \hat{p}(\theta_i)$, where $\hat{p} : \mathbb{R}^d \to \mathbb{R}^d$ is obtained by replacing $V_1, \nabla V_1$, and ∇V_0 in (2) by their estimates, i.e.,

$$\hat{p}(\theta) = \arg\min_{y \in \mathbb{R}^d} \widehat{\nabla V_0}(\theta)^\top (y - \theta) + \frac{1}{2h}\|y - \theta\|^2 \tag{3a}$$

$$\text{s.t. } \alpha h \widehat{V_1}(\theta) + \widehat{\nabla V_1}(\theta)^\top (y - \theta) + \frac{1}{2h}\|y - \theta\|^2 \leq 0. \tag{3b}$$

Note that $\hat{p}$ is well defined on $\hat{\mathcal{F}} := \{\theta \in \mathbb{R}^d : \exists y \in \mathbb{R}^d \text{ s.t. } \alpha h \widehat{V_1}(\theta) + \widehat{\nabla V_1}(\theta)^\top (y - \theta) + \frac{1}{2h}\|y - \theta\|^2 \leq 0\}$. Furthermore, $\{\theta \in \mathbb{R}^d : \widehat{V_1}(\theta) \leq 0\} \subset \hat{\mathcal{F}}$, which ensures that $\hat{p}$ is defined for all policies that are estimated to satisfy (1b). Our approach is summarized in Algorithm 1, and since it takes inspiration from considering discretizations of the Safe Gradient Flow (SGF) (Allibhoy and Cortés, 2024), we refer to it as Reinforcement Learning-based Safe Gradient Flow (RL-SGF).

Algorithm 1 RL-SGF

Parameters: α, h, T, k, and a sequence $\{N_i\}_{i=1}^k$, **Initial Policy Parameter**: θ_1;
for $i \in [1, \ldots, k]$ **do**
  Generate N_i episodes of length $T + 1$ using policy π_{θ_i};
  Compute $\widehat{V_1}(\theta_i)$ as defined Lemma 2 (i) and $\widehat{\nabla V_1}(\theta_i)$, $\widehat{\nabla V_0}(\theta_i)$ as defined in Lemma 2 (ii);
  $\theta_{i+1} \leftarrow \hat{p}(\theta_i)$ as defined in (3);
end for

4. Anytime Safety and Convergence Properties of RL-SGF

In this section we study the anytime and convergence properties of Algorithm 1. We fix $\alpha > 0$, $h \in (0, \min\{\frac{1}{L_0}, \frac{1}{L_1}, \frac{1}{\alpha}\})$, and assume that for any $\theta \in \mathbb{R}^d$, the estimates $\widehat{V_1}(\theta)$, $\widehat{\nabla V_0}(\theta)$, and $\widehat{\nabla V_1}(\theta)$ are computed according to Lemma 2 with N different episodes of length $T + 1$ of the policy π_θ. First we provide a closed-form expression for the update maps $\hat{p}$ and p.

Lemma 3 (Closed-form of update maps): *Let* $\theta \in \mathbb{R}^d$ *and suppose that Slater's condition holds for* (3) [2]. *Define* $\hat{A}_\theta = \|\widehat{\nabla V_1}(\theta)\|^2 - 2\alpha\widehat{V_1}(\theta)$, $\hat{B}_\theta = 2\hat{A}_\theta$, $\hat{C}_\theta = 2\widehat{\nabla V_1}(\theta)^\top \widehat{\nabla V_0}(\theta) - \|\widehat{\nabla V_0}(\theta)\|^2 - 2\alpha\widehat{V_1}(\theta)$, $\hat{\Delta}_\theta = 4\|\widehat{\nabla V_1}(\theta) - \widehat{\nabla V_0}(\theta)\|^2 \hat{A}_\theta$, *and* $\hat{u}_\theta = \frac{-\hat{B}_\theta + \sqrt{\hat{\Delta}_\theta}}{2\hat{A}_\theta}$. *Then,*

$$\hat{p}(\theta) = \begin{cases} \theta - h\widehat{\nabla V_0}(\theta) & \text{if } \hat{A}_\theta > 0, \hat{C}_\theta \geq 0 \\ \theta - \frac{h}{1+\hat{u}_\theta}(\widehat{\nabla V_0}(\theta) + \hat{u}_\theta \widehat{\nabla V_1}(\theta)) & \text{if } \hat{A}_\theta > 0, \hat{C}_\theta < 0, \\ \theta - h\widehat{\nabla V_1}(\theta) & \text{if } \hat{A}_\theta = 0 \end{cases}$$

2. I.e., $\exists\, y \in \mathbb{R}^d$ with $\alpha h \widehat{V_1}(\theta) + \widehat{\nabla V_1}(\theta)^\top (y - \theta) + \frac{1}{2h}\|y - \theta\|^2 < 0$ (cf. (Boyd and Vandenberghe, 2009, Section 5.2.3))

If Slater's condition holds for (2), an analogous expression holds for p, with the estimates replaced by their true values.

The proof of Lemma 3 follows an argument analogous to that of (Auslender et al., 2010, Proposition 3.1) and is included in the extended version Mestres et al. (2024). Note that if $\widehat{V_1}(\theta) \leq -\eta$ (for some $\eta \geq 0$), then $\hat{A}_\theta \geq 2\alpha\eta \geq 0$. In the sequel, we denote by A_θ, B_θ, C_θ, Δ_θ, and u_θ the quantities analogous to $\hat{A}_\theta$, $\hat{B}_\theta$, $\hat{C}_\theta$, $\hat{\Delta}_\theta$, and $\hat{u}_\theta$, as defined in Lemma 3 but substituting the estimates by their true values.

Remark 4 (Connection with primal-dual methods): Primal-dual methods, see e.g., Paternain et al. (2019), employ the Lagrangian function associated to the constrained optimization problem (1), updating the primal variable θ via gradient descent and the dual variable λ through gradient ascent. If the value of the dual variable at iteration $i \in \mathbb{Z}_{>0}$ is λ_i, the primal update is $\theta_{i+1} = \theta_i - \eta(\widehat{\nabla V_0}(\theta) + \lambda_i \widehat{\nabla V_1}(\theta))$ (here η is a predefined stepsize). One can see the parallelism, cf. Lemma 3, with the policy update performed by RL-SGF, $\hat{p}(\theta) = \theta - \frac{h}{1+\hat{u}_\theta}(\widehat{\nabla V_0}(\theta) + \hat{u}_\theta \widehat{\nabla V_1}(\theta))$ (note how the other two expressions are recovered from this one by setting $\hat{u}_\theta = 0$ and $\hat{u}_\theta = \infty$, respectively). When comparing both algorithms, we can interpret the RL-SGF update rule as a modification of the primal step of the primal-dual algorithm, where the stepsize η is state-dependent and takes the value $\eta = \frac{h}{1+\hat{u}_\theta}$, and $\lambda_i = \hat{u}_\theta$. This is consistent with the fact that the Safe Gradient Flow (Allibhoy and Cortés, 2024, Section IV.A) admits a primal-dual interpretation, where control inputs play the role of approximators of the dual variables. The key difference is that, while primal-dual methods use a fixed stepsize, RL-SGF dynamically adjusts η_θ and λ_i at each iteration in a state-dependent fashion, in order to ensure that the next update remains feasible. This eliminates the need for fine-tuning hyperparameters. Our experiments suggest that this adaptive approach improves algorithm performance, see Figure 2 below. Additionally, the initialization of the dual variable is critical for the overall safety performance of the algorithm. For example, if the dual variable is initialized below the optimal value, the primal-dual algorithm would inevitably violate the safety constraint during the training process, whereas RL-SGF does not require initializing any dual variable and is guaranteed to be safe as long as the initial policy is safe. •

The following result utilizes Lemma 2 to provide safety guarantees for Algorithm 1.

Proposition 5 (Safety guarantees of RL-SGF): *Let $k \in \mathbb{Z}_{>0}$, $i \in [k]$, $\delta \in (0,1)$, and $\bar{\sigma}_1$, $\tilde{\sigma}_1$ be as in Lemma 2. Suppose that Assumptions 1 and 2 hold. For the ith iterate $\theta_i \in \mathbb{R}^d$ of Algorithm 1,*

(i) If $\widehat{V_1}(\theta_i) \leq 0$, let $N_i > \max\left\{\frac{-2\tilde{\sigma}_1^2 \log(\delta)}{\hat{M}_i^2}, \frac{-2d\bar{\sigma}_1^2 \log(\frac{\delta}{d})}{\hat{M}_i^2}\right\}$, with $\hat{M}_i = \frac{(1-\alpha h)|\widehat{V_1}(\theta_i)| + \frac{1}{2}(\frac{1}{h} - L_1)\|\hat{p}(\theta_i) - \theta_i\|^2}{1 + \|\hat{p}(\theta_i) - \theta_i\|}$.

(ii) If $\widehat{V_1}(\theta_i) \geq 0$, let $N_i > \max\left\{\frac{-2\tilde{\sigma}_1^2 \log(\delta)}{\nu^2}, \frac{-2d\bar{\sigma}_1^2 \log(\frac{\delta}{d})}{\nu^2}\right\}$, with $\nu > 0$ such that

$$\nu(1 + \|\hat{p}(\theta_i) - \theta_i\|) + (1 - \alpha h)\widehat{V_1}(\theta_i) < \frac{(\frac{1}{h} - L_1)\|\hat{p}(\theta_i) - \theta_i\|^2}{2}. \tag{4}$$

In either case, we have $\mathbb{P}(V_1(\hat{p}(\theta_i)) \leq 0)) \geq 1 - 2\delta$.

Proposition 5 (i) shows that if $\widehat{V_1}(\theta_i) \leq 0$ (i.e., we estimate that the ith iterate is safe), then by running a sufficiently large number of episodes the next iterate of Algorithm 1 is safe with an arbitrarily high probability. On the other hand, Proposition 5 (ii) provides similar guarantees

when we estimate that the policy is unsafe (i.e., $\widehat{V_1}(\theta_i) \geq 0$). We note that if $\widehat{V_1}(\theta_i) \leq 0$, (4) is satisfied by taking $\nu = 0$. By continuity, this suggests that (4) is also satisfied in a neighborhood of $\{\theta \in \mathbb{R}^d \ : \ \widehat{V_1}(\theta) \leq 0\}$, and it becomes increasingly difficult to satisfy as $\widehat{V_1}(\theta)$ grows, which indicates that safety can be ensured in the next iterate as long as the safety violation is not too large.

Note that the lower bounds on N_i in Proposition 5 depend on $\hat{V}_1(\theta_i)$, $\widehat{\nabla V_1}(\theta_i)$, and $\widehat{\nabla V_0}(\theta_i)$, which in turn also depend on N_i. Therefore, we use the result in Proposition 5 is as follows: given a number of episodes N_i, we construct the estimates $\hat{V}_1(\theta_i)$, $\widehat{\nabla V_1}(\theta_i)$, and $\widehat{\nabla V_0}(\theta_i)$. If the lower bounds in N_i in Proposition 5 hold, the guarantees provided therein apply. If they do not, one should increase N_i. This process is guarantees to terminate, because the estimates $\widehat{V_1}(\theta_i)$, $\widehat{\nabla V_1}(\theta_i)$, $\widehat{\nabla V_0}(\theta_i)$ converge to the true values $V_1(\theta_i)$, $\nabla V_1(\theta_i)$, $\nabla V_0(\theta_i)$ respectively as $N_i \to \infty$, so the lower bounds on N_i in Proposition 5 are finite as $N_i \to \infty$. Therefore, there exists a sufficiently large N_i for which the conditions on N_i in Proposition 5 hold.

From Proposition 5, one can also obtain safety guarantees over a finite time horizon with a prescribed confidence level (cf. (Mestres et al., 2024, Corollary 5)). This ensures that with a choice of $\{N_i\}_{i=1}^k$ as described in Proposition 5, Algorithm 1 is anytime.

Next we show the convergence of Algorithm 1 to a neighborhood of a KKT point of (1). Based on Lemma 1 (ii), given a tolerance $\epsilon^* > 0$, we seek to find $\theta \in \mathbb{R}^d$ such that $V_1(\theta) \leq 0$ and $\|p(\theta) - \theta\| \leq \epsilon^*$. As shown in (Andreani et al., 2010, Theorem 4.1), by sufficiently decreasing ϵ^*, a vector $\theta \in \mathbb{R}^d$ satisfying $\|p(\theta) - \theta\| \leq \epsilon^*$ can be made arbitrarily close to a KKT point.

Proposition 6 (Convergence guarantees of RL-SFG): *Suppose that Assumptions 1 and 2 hold. Suppose there exist positive constants $\widehat{\eta_A}$, η_A such that $\hat{A}_{\theta_i} \geq \widehat{\eta_A}$ and $A_{\theta_i} \geq \eta_A$. Given $\epsilon^* > 0$, there exist constants*[3] *$M_p, \bar{M}_p$ and K_p such that if*

(i) $\epsilon > 0$ *is such that* $\sqrt{\bar{M}_p \epsilon} + h K_p \epsilon \leq \epsilon^*$;

(ii) $k \in \mathbb{Z}_{>0}$ *is such that* $k \geq \frac{2\tilde{\sigma}_0}{M_p \epsilon}$;

(iii) for each $i \in [k]$, *Slater's condition holds for* (2) *and* (3) *with* $\theta = \theta_i$;

(iv) $\widehat{V_1}(\theta_i) \leq 0$;

(v) $N_i > \max\{\frac{d\bar{\sigma}_1^2}{\epsilon}, \frac{d\bar{\sigma}_0^2}{\epsilon}, \frac{\tilde{\sigma}_1^2}{\epsilon}\}$ *for all* $i \in [k]$ *(with* $\bar{\sigma}_0$, $\bar{\sigma}_1$, *and* $\tilde{\sigma}_1$ *as defined in Lemma 2);*

then, there exists $j \in [k]$ *such that* $\mathbb{E}[\|\hat{p}(\theta_j) - \theta_j\|] \leq \sqrt{\bar{M}_p \epsilon}$. *Moreover,* $\|p(\theta_j) - \theta_j\| \leq \epsilon^*$.

Proposition 6 ensures that if both k and each N_i, $i \in [k]$, are sufficiently large, RL-SGF returns a point arbitrarily close to a KKT point of (1). In order to detect the index $j \in [k]$ for which $\mathbb{E}[\|\hat{p}(\theta_j) - \theta_j\|] \leq \sqrt{\bar{M}_p \epsilon}$, we can compute different realizations of $\hat{p}(\theta_i)$ for every $i \in [k]$, compute an empirical estimate of $\mathbb{E}[\|\hat{p}(\theta_i) - \theta_i\|]$, and check whether $\mathbb{E}[\|\hat{p}(\theta_i) - \theta_i\|] \leq \sqrt{\bar{M}_p \epsilon}$. We note that the issue of identifying the index for which the convergence criterion is satisfied also appears in the convergence results of policy gradient (cf. (Zhang et al., 2020, Theorem 4.3)).

3. The constants M_p, $\bar{M}_p$, and K_p are defined as follows. First, let $M_q = \sqrt{d}\bar{\sigma}_q$, $q \in \{0, 1\}$, $M_A = (M_1)^2 + 2\alpha\tilde{\sigma}_1$, $M_B = 2M_A$, $M_C = 2M_1 M_0 + 2\alpha M_1$, $M_\Delta = 4(M_1 + M_0)^2 M_A$. Further define $\widehat{\eta_B} = 2\widehat{\eta_A}$, $M_u = \frac{M_B + \sqrt{M_\Delta}}{2\eta_A}$, $K_A = 2M_1$, $K_B = 4M_1$, $K_C = 2M_1 + 4M_0$, $K_\Delta = \frac{K_B + 2M_B K_B + 4K_A M_C + 4M_A K_C}{\widehat{\eta_\Delta}}$, $K_u = \max\left\{\frac{K_A(M_B + \sqrt{M_\Delta})}{2\eta_A \widehat{\eta_A}} + \frac{K_\Delta + K_B}{2\eta_A}, \frac{2K_C}{\eta_B}, \frac{2K_C}{\widehat{\eta_B}}\right\}$. Then, we define $M_p = h(M_0 + M_u M_1)$, $\bar{M}_p = \frac{2M_p}{\frac{1}{\hbar} - \frac{L_0}{2}}$, and $K_p = 1 + K_u\bar{\sigma}_1 + M_u + K_u$.

5. Simulations

In this section we test RL-SGF in a simple navigation 2D example.

Environment: We consider a continuous state and action space environment. We work with two different dynamics:

- *Single integrator*: $s_{t+1} = s_t + 0.1a_t$ with $s_t \in \mathcal{S} = \mathbb{R}^2$ and $a_t \in \mathcal{A} = [-5, 5]^2$ for $t \in \mathbb{Z}_{>0}$;
- *Differential-drive robot*: $s_{t+1} = s_t + 0.2\, v_t(\cos\theta_t, \sin\theta_t)$ and $\theta_{t+1} = \theta_t + 0.2\,\omega_t$ with $s_t = (x_t, \theta_t) \in \mathcal{S} = \mathbb{R}^2 \times [-\pi, \pi]$, $a_t = (v_t, \omega_t) \in \mathcal{A} = [0, 5] \times [-20\pi/180, 20\pi/180]$ for $t \in \mathbb{Z}_{>0}$.

For the single integrator, s_t is the position of the agent at time t, and for the differential-drive robot, x_t denotes the agent's position at time t and θ_t represents its orientation at time t. For both dynamics, we refer to the position component of the state as $s_x \in \mathbb{R}^2$. Given a target state $x^* = (8, 8)$, we define $R_0(s, a, s') = \max\{-\|s_x - x^*\|, -10\}$. We consider a set of five circular and rectangular obstacles denoted by $\{\mathcal{O}_j\}_{j=1}^5$ as depicted in red in Figure 1, and we let the safe set $\mathcal{C}$ be the subset of $[0, 10] \times [0, 10] \subset \mathbb{R}^2$ not covered by the obstacles. We take $T = 50$ as the time horizon and $\gamma = 0.98$. To define $R_1(s, a, s')$, we let $d_{\min}(s)$ be the minimum distance between s_x and any obstacle and set $R_1(s, a, s') = \beta\left(e^{-d_{\min}(s)} - 1\right)$ if $s_x \in \mathcal{C}$ and $R_1(s, a, s') = 1 - \beta$ if $s_x \notin \mathcal{C}$, with $\beta > 0$ being a design parameter. This choice of R_1 is inspired by Paternain et al. (2023), and in the limit $\gamma \to 1$ guarantees that trajectories generated by policies satisfying the associated safety constraint (1b) do not collide with obstacles with probability greater than $1 - \beta T$. Since R_0 and R_1 are bounded, Assumption 1 holds.

Policy: We consider a truncated Gaussian policy class$\pi_\theta(a|s) = \bar{C} e^{-\frac{1}{2}(a-\mu_\theta(s))^\top \Sigma^{-1}(a-\mu_\theta(s))}$, if $a \in \mathcal{A}$ and $\pi_\theta(a|s) = 0$ if $a \notin \mathcal{A}$, where $\Sigma = 0.5\mathbf{I}_2$, and $\bar{C}$ is a normalization constant. The mean function is defined as $\mu_\theta(s) = \sum_{i=0}^{N_c} \tanh(\theta_i) \exp\left(-\frac{\|s_x - c_i\|^2}{2\sigma^2}\right)$, where $\tanh$ is applied componentwise, $\theta_i \in \mathbb{R}^2$ are the trainable parameters, and $\{c_i\}_{i=1}^{N_c}$ is a set of points in $\mathcal{A}$. For the single-integrator dynamics, the set $\{c_i\}$ is constructed by dividing the safe set $\mathcal{C}$ into a grid with 20 divisions per dimension, resulting in $N_c = 400$ points. For the differential-drive dynamics, an additional dimension with 10 divisions over the range $[-\pi, \pi]$ is added, leading to $N_c = 4000$ points. This policy parameterization satisfies the required properties, i.e., π_θ satisfies Assumption 2. For the single integrator, the initial policy parameter θ_1 is constructed as follows. Let $\rho, f_{\max} \in \mathbb{R}_{>0}$, q_j be the center of obstacle $\mathcal{O}_j$ and $v_j^i = (c_i - q_j)/\|c_i - q_j\|$. Now, for $i \in [400]$ and $j \in [5]$, we let $Q_j^i = f_{\max}(1 - d(c_i, \mathcal{O}_j)/\rho)v_j^i$ if $d(c_i, \mathcal{O}_j) < \rho$ and $Q_j^i = 0$ otherwise and set $(\theta_1)_i = \sum_{j=1}^5 Q_j^i$. As shown in Figure 1 (left), the resulting mean μ_{θ_1} of π_{θ_1} promotes actions that point away from the obstacles, yielding a safe initial policy (it can be checked that for a sufficiently large N_1, $\widehat{V_1}(\theta_1) < 0$). For the differential-drive dynamics and due the difficulty of obtaining an initial safe policy (other than the trivial one where zero inputs are taken at any state), we initizalize parameters θ_1 randomly. This typically results in an initial policy which is unsafe, and we use this to illustrate the ability of RL-SGF to recover.

Training: For the single-integrator case, we take $N = 100$, $\alpha = 1$, $\beta = 0.01$, $h = 0.5$ and $K = 1500$. We compare its performance with the primal-dual method in (Paternain et al., 2023, Algorithm 2) with $\eta_\theta = \eta_\lambda = 0.001$ as the primal and dual step sizes and Constrained Policy Optimization (CPO) Achiam et al. (2017), where we take $\delta = 0.15$ and $H = \mathbf{I}_d$. To make a fair comparison, we use the same estimates introduced in Lemma 2 for all algorithms, setting $b(s) = 0$. For the differential-drive robot case, we use $N = 200$, $\alpha = 9$, $\beta = 0.05$, $h = 0.1$ and $K = 4000$, keeping all other parameters unchanged.

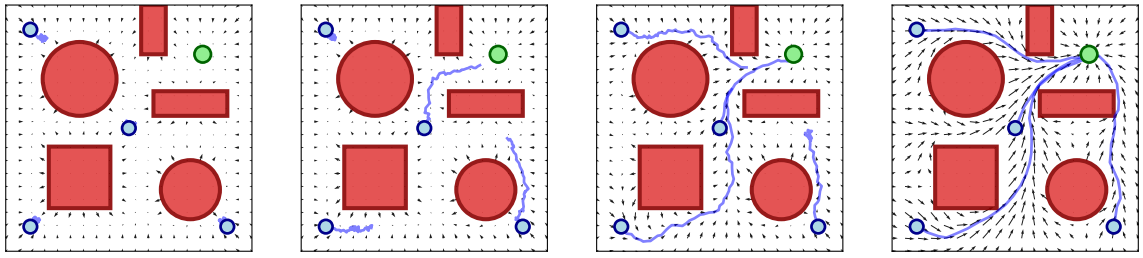

Figure 1: Evolution of the policy π_θ generated by RL-SGF at diferent stages of the learning process for the single-integrator dynamics. The arrows indicate the mean μ_θ, the target state $x^* = (8, 8)$ is marked in green and four trajectories starting at $(1, 1), (5, 5), (9, 1)$ and $(1, 9)$ are plotted in blue. Obstacles are depicted in red. From left to right: initial policy and after 200, 500, and 1500 iterations.

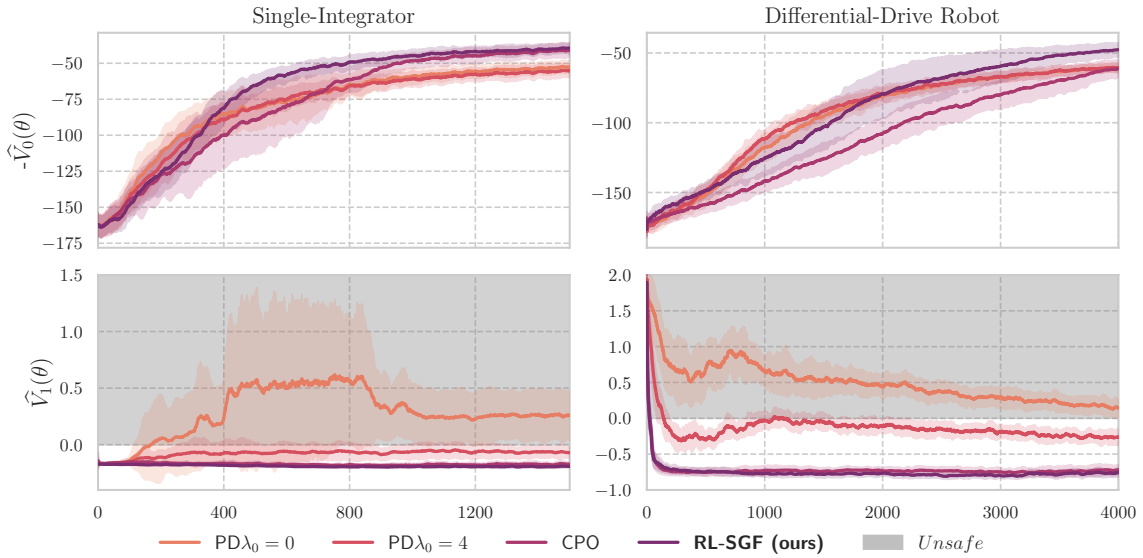

Figure 2: Comparison between RL-SGF, primal-dual approaches (PD), and Constrained Policy Optimization (CPO). Evolution of the average return $\widehat{V_0}(\theta)$ and safety measure $\widehat{V_1}(\theta)$ for single-integrator (left) and differential-drive (right) dynamics. The initial dual variable is denoted λ_0. Shaded areas represent 95% confidence intervals over 5 runs. The unsafe region ($\widehat{V_1}(\theta) > 0$) is in gray.

Results: Figure 1 shows four different snapshots describing the policies obtained during the training process of RL-SGF. Qualitatively, we observe how the vector field described by the mean of the Gaussian policy always points away from the obstacles, suggesting that all the policies are safe. We also observe how policies trained with a larger number of iterations result in trajectories that make faster progress towards the target point, as expected. Figure 2 compares RL-SGF with the primal-dual method in (Paternain et al., 2023, Algorithm 2) and CPO (Achiam et al., 2017) using the same estimators. The initial dual variable λ_0 is set to 0 and 4. In both cases, the primal-dual method results in significant constraint violations, whereas RL-SGF and CPO maintain safety during training. In Figure 2 (right), notice also how RL-SGF and CPO recover from initial unsafety and reach a safe policy much faster than the primal-dual algorithm. Simulations also show that, for the single integrator, the estimated cumulative reward $\widehat{V_0(\theta)}$ for RL-SGF becomes larger than that of the primal-dual algorithm after 300 steps and outperforms CPO throughout the entire training process. This is even more apparent in the differential-drive robot case. Table 1 presents the average performance over the last 100 training steps and the percentage of safe policies (i.e., $\widehat{V_1}(\theta) \leq 0$)

observed during the training process across different algorithms. RL-SGF displays the highest safety performance for both dynamics, while outperforming other approaches in terms of average return.

Dynamics	PD λ_0=0	PD λ_0=4	CPO	**RL-SGF (ours)**
Single-integrator	-52.99 (26.68%)	-55.53 (85.37%)	-41.33 (99.88%)	**-39.95 (99.96%)**
Differential-drive	-60.54 (2.17%)	-60.66 (83.28%)	-61.52 (99.72%)	**-48.01 (99.83%)**

Table 1: Average performance ($-\widehat{V_0}(\theta)$) over the last 100 training steps. In parentheses, we report the percentage of safe policies (i.e., $\widehat{V_1}(\theta) \leq 0$). Results are averaged over 5 random seeds.

Finally, we study the effect on the performance of RL-SGF of the number of episodes used in the estimates. For simplicity, we assume that the number of episodes at each iteration is the same, i.e., $N_i = N$ for all $i \in [k]$. Figure 3 shows that, by increasing N, we obtain faster convergence and less variance on the estimators while reducing the number of unsafe policies, as expected. In addition, all tested values of N lead to safe policies during the training process, with minimal constraint violations even for $N = 10$ (less than 0.4%). This could be an indication that, for this environment, the bounds in Proposition 5 are conservative. In the case of $N = 400$, RL-SGF achieves zero constraint violation during training.

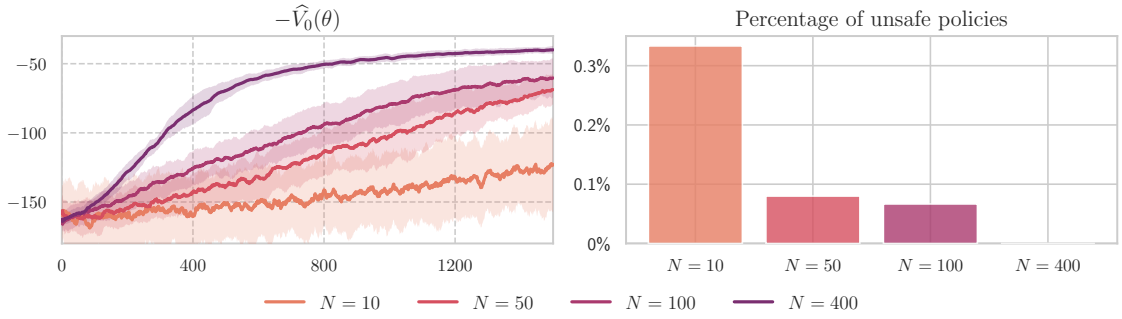

Figure 3: Illustration of the performance of RL-SGF as a function of the number of episodes N used in the estimates of the value functions and their gradients. Left plot shows the evolution of the average return $\widehat{V_0}(\theta)$ and right plot shows the safety measure $\widehat{V_1}(\theta)$ during training. Shaded areas are 95 % confidence intervals over 5 runs.

6. Conclusions

We have introduced RL-SGF, a constrained RL algorithm with anytime guarantees. At every iteration, RL-SGF uses episodic data to construct estimates of the value functions and their gradients associated with the objective and safety constraints. By deriving appropriate statistical properties of such estimates, we show that if the number of episodes at each iteration is sufficiently large, the algorithm returns a safe policy in the next iteration with arbitrarily high probability, and converges to a KKT point. We have illustrated the performance of RL-SGF in a simple 2D navigation example. As future work, we plan to improve the sample-complexity guarantees by using off-policy methods, explore schemes that adaptively tune the algorithm parameters to improve convergence, and extend this approach to other types of safety constraints, such as probabilistic or conditional value-at-risk. We also plan to apply this algorithm to more complex safety-critical systems.

Acknowledgments

This work was partially supported by ONR Award N00014-23-1-2353.

References

J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 22–31, Sydney, Australia, 2017.

A. Allibhoy and J. Cortés. Control barrier function-based design of gradient flows for constrained nonlinear programming. *IEEE Transactions on Automatic Control*, 69(6):3499–3514, 2024.

E. Altman. *Constrained Markov Decision Processes*, volume 7. CRC Press, 1999.

R. Andreani, J. M. Martinez, and B. F. Svaiter. A new sequential optimality condition for constrained optimization and algorithmic consequences. *SIAM Journal on Optimization*, 20(6): 3533–3554, 2010.

A. Auslender, R. Shefi, and M. Teboulle. A moving balls approximation method for a class of smooth constrained minimization problems. *SIAM Journal on Optimization*, 20(6):3232–3259, 2010.

Q. Bai, A. S. Bedi, M. Agarwal, A. Koppel, and V. Aggarwal. Achieving zero constraint violation for constrained reinforcement learning via primal-dual approach. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 3682–3689, Vancouver, Canada, 2022.

Q. Bai, A. S. Bedi, and V. Aggarwal. Achieving zero constraint violation for constrained reinforcement learning via conservative natural policy gradient primal-dual algorithm. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 6737–6744, Washington D. C., USA, 2023.

Q. Bai, W. U. Mondal, and V. Aggarwal. Regret analysis of policy gradient algorithm for infinite horizon average reward Markov decision processes. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 10980–10988, Vancouver, Canada, 2024.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2009. ISBN 0521833787.

L. Brunkel, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe learning in robotics: from learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A Lyapunov-based Approach to Safe Reinforcement Learning. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 8103–8112, Montreal, Canada, 2018.

D. Ding, K. Zhang, T. Basar, and M. Jovanovic. Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, volume 33, pages 8378–8390, Online Conference, 2020.

D. Ding, X. Wei, Z. Yang, Z. Wang, and M. Jovanovic. Provably efficient safe exploration via primal-dual policy optimization. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3304–3312, Online Conference, 2021.

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

S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll. A review of safe reinforcement learning: methods, theory and applications. *arXiv preprint arXiv:1809.01674v2*, 2022.

Y. Liu, J. Ding, and X. Liu. IPO: Interior-Point Policy Optimization under Constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4940–4947, New York, USA, 2020.

Y. Liu, H. Avishai, and X. Liu. Policy learning with constraints in model-free reinforcement learning: a survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4508–4515, Montreal, Canada, 2021.

P. Mestres, A. Marzabal, and J. Cortés. Anytime safe reinforcement learning. https://drive.google.com/file/d/1TT0eyd64hbari2SYsFH3lqYnEVUfYaI5/view?usp=sharing, 2024.

S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro. Constrained Reinforcement Learning Has Zero Duality Gap. In *Proceedings of the 32nd Conference in Neural Information Processing Systems*, volume 32, pages 7555–7565, Vancouver, Canada, 2019.

S. Paternain, M. Calvo-Fullana, L. F. O. Chamon, and A. Ribeiro. Safe Policies for Reinforcement Learning via Primal-Dual Methods. *IEEE Transactions on Automatic Control*, 68(3):1321–1336, 2023.

W. Suttle, V. K. Sharma, K. C. Kosaraju, S. Seetharaman, J. Liu, V. Gupta, and B. M. Sadler. Sampling-based safe reinforcement learning for nonlinear dynamical systems. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pages 4420–4428, Valencia, Spain, 2024.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

S. Zeng, T. T. Doan, and J. Romberg. Finite-time complexity of online primal-dual natural actor-critic algorithm for constrained Markov decision processes. In *IEEE Conf. on Decision and Control*, pages 4028–4033, Cancun, Mexico, 2022.

K. Zhang, A. Koppel, H. Zhu, and T.Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020.

T. Zhao, H. Hachiya, G. Niu, and M. Sugiyama. Analysis and improvement of policy gradient estimation. *Neural Networks*, 26:118–129, 2012.