Learning to Act Safely with Limited Exposure and Almost Sure Certainty

Agustin Castellano, *Student Member, IEEE*, Hancheng Min, *Student Member, IEEE*, Juan Andres Bazerque, *Member, IEEE*, and Enrique Mallada, *Senior Member, IEEE*

***Abstract*— This paper puts forward the concept that learning to take safe actions in unknown environments, even with probability one guarantees, can be achieved without the need for an unbounded number of exploratory trials. This is indeed possible, provided that one is willing to navigate trade-offs between optimality, level of exposure to unsafe events, and the maximum detection time of unsafe actions. We illustrate this concept in two complementary settings. We first focus on the canonical multi-armed bandit problem and study the intrinsic trade-offs of learning safety in the presence of uncertainty. Under mild assumptions on sufficient exploration, we provide an algorithm that provably detects all unsafe machines in an (expected) finite number of rounds. The analysis also unveils a trade-off between the number of rounds needed to secure the environment and the probability of discarding safe machines.**

We then consider the problem of finding optimal policies for a Markov Decision Process (MDP) with almost sure constraints. We show that the action-value function satisfies a barrier-based decomposition which allows for the identification of feasible policies independently of the reward process. Using this decomposition, we develop a Barrier-learning algorithm, that identifies such unsafe state-action pairs in a finite expected number of steps. Our analysis further highlights a trade-off between the time lag for the underlying MDP necessary to detect unsafe actions, and the level of exposure to unsafe events. Simulations corroborate our theoretical findings, further illustrating the aforementioned trade-offs, and suggesting that safety constraints can speed up the learning process.

***Index Terms*— Uncertain systems, randomized algorithms, Markov processes, iterative learning control, optimal control**

I. Introduction

Motivated by the success of machine learning in achieving super human performance, e.g., in vision [3], speech [4], and video games [5], there has been recent interest in developing learning-enabled technology that can implement highly complex actions for safety-critical autonomous systems, such as self-driving cars, robots, etc. However, without proper safety guarantees such systems will rarely be deployed. There is therefore the need to develop analysis tools and algorithms that can provide such guarantees during, and after, training. Efforts to provide such guarantees can be broadly grouped in two lines of work with somehow complementary success.

The first approach leverages model-based techniques, based on Lyapunov stability [6] and robust control [7], to provide (worst-case) safety guarantees based on a nominal model and assumptions on uncertainty and disturbances [8]–[11]. In such settings, safety is usually specified in terms of stability, robust stability, or the existence of some invariant or control invariant sets. Moreover, due to the worst-case approach to uncertainty, these methods tend to suffer poor performance in settings where the uncertainty is large, or the system performance is highly sensitive to model-uncertainty.

The second line of work, in which ours naturally lies , seeks to provide safety guarantees in model-free settings by adding constraints to the learning problem [12]–[19]. In this way, safety specifications can be further extended, beyond typical control notions, at the expense of introducing uncertainty and risk in the safety guarantees. More precisely, constraints in this body of work are probabilistic, either in expected value, critical value-at-risk, or high-probability, which do not allow for settings with hard constraints that need to be satisfied with probability one (w.p.1). Moreover, the satisfaction guarantees for these constraints are also generally provided probabilistically, via (expected) regret, or in high probability.

Naturally, both methodologies have their advantages. On the one hand, model-based methods can impose probability one constraints, and can also guarantee their satisfaction w.p.1. On the other hand, model-free methods can handle problems with high degree of uncertainty, and are particularly suited for cases where constraints and objectives are difficult to formalize. Our work aims for robust safety guarantees more similar to the first approach (i.e., in an almost-sure sense), while borrowing methodologies from the second one (i.e., in a model-free setting), in an attempt to "get the best of both worlds."

A. Contributions

One of the driving arguments of our work is that, due to the logical (safe/unsafe) nature of safety assessment, finding safe actions is a problem fundamentally different (and easier

A preliminary version of Section II was first presented in [1]. A more in-depth discussion of Section III-F can be found in [2].

A. Castellano, H. Min and E. Mallada are with the Department of Electrical and Computer Engineering at Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: {acaste11,hanchmin,mallada}@jhu.edu).

J. A. Bazerque is with the Department of Electrical Engineering at University of Pittsburgh, PA, 15213 (e-mail: juanbazerque@pitt.edu).

The work at Johns Hopkins was supported by NSF through grants CAREER 1752362, CPS 2136324, and TRIPODS 1934979. Bazerque's work was supported by ANII FSE_1_2019_1_159457.

to solve) than finding the best one. To that end, we model safety assessment by a *damage signal* $D_t \in \{0, 1\}$, with 1 indicating an unsafe event. We develop safety-assessment algorithms that learn which actions (or state-action pairs) satisfy safety specifications—w.p.1—both on Multi-armed Bandits (MABs) [20] and Markov Decision Processes (MDPs) [21].

The algorithms we develop for both MABs and MDPs learn the set of all feasible policies. In both cases we provide *explicit finite bounds* on the (expected) time needed to learn this set. It is important to note, though, that since our approach is model-free, unsafe events during training are unavoidable. That being said, during training our algorithms also *limit the number of constraint violations*. We refer to the latter as *exposure* throughout the paper, and define it both for MABs and MDPs. Under both settings we also provide relaxed formulations of the problems that allow for less restrictive constraints. Specifically, we consider scenarios that admit policies that endure some damage along the trajectory.

Early developments for MABs were first presented in [1]. New to this work are extensions of all the results to λ-soft strategies (Definition 4), as well as an improved bound for Theorem 2. The framework for MDPs was first introduced in [2], but with no theoretical proofs. This paper contains all the proofs that support our theory.

B. Related work

Safety in multi-armed bandits can be posed as guaranteeing that with high probability the expected cost of pulling an arm lies below a certain threshold. It has been studied in the linear setting [22], [23] and when both the reward and cost functions follow Gaussian Processes [24], yielding methods similar to the celebrated Upper Confidence Bound algorithm [25] that first explore and expand a safe set and then focus on controlling regret [26]. In contrast, in Section II we consider safety for a binary approach, where the probability of an arm failing follows a Bernoulli distribution (of unknown parameter). Instead of expanding a safe region known a priori, our algorithm sequentially eliminates unsafe arms.

Safety in control is usually specified in terms of reaching a certain region of the state space. In the context of learning, many works seek to apply Lyapunov design [9], sometimes in tandem with Gaussian processes [8], [10] to obtain safe policies. Other proposed methods are related to using control barrier functions [27], [28] to specify safety guarantees and embed them into the cost to be optimized [11]. While very powerful, these methods usually require knowledge of the system dynamics or a candidate Lyapunov/barrier function. Our framework on the other hand is model-free and thus assumes no prior knowledge. Safety specifications are implicitly captured by the damage signal and the transition kernel, yielding a safe region of the state space akin to what is achieved by computing backward reachable sets [29], [30].

Constraints in MDPs are typically formulated as expectation-based constraints [31] or as critical value-at-risk [32]. If the transitions and rewards are known, an optimal stationary policy for the first kind can be found as the solution of a linear program [33]. Under unknown dynamics, typical strategies rely on primal-dual methods [**?**], [15]–[17], [34]–[36] , or on exploring and approximating the safety region [13], [14], [18]. The main difference in the approach of Section III is that we propose an MDP problem with almost sure constraints. Its particular nature proves useful in the sense that the set of feasible policies can be characterized in (expected) finite time.

Controller Synthesis has also been used in the context of RL, where constraints are specified as high-level objectives given by Linear Temporal Logic [37]–[39]. In these works the goal is a policy that maximizes the probability of satisfying a set of tasks [39]. These methods are powerful in the sense that they can capture a broad range of objectives and in some cases, even work on continuous spaces [40]. That being said, such methods rely on reward-shaping techniques that couple feasibility/safety with optimality. Our work decouples such problems (via the computation of a barrier function B^*).

C. Outline of the paper

Section II addresses safety specifications in MABs, of the form $P(D_t = 1|A_t) \leq \mu$ for some $\mu \geq 0$. While the *"flawless"* case $\mu = 0$ is rather straightforward (Section II-A), the *"relaxed"* case $\mu > 0$ requires to trade-off between quickly discarding unsafe arms and accurately estimating $P(D_t = 1|A_t)$, which we achieve with an arm-elimination algorithm based on Sequential Probability Ratio Tests (SPRTs) [41]. We focus on rapid (finite time) detection of unsafe actions almost surely, which naturally requires to (mildly) give up optimality by possibly discarding some safe arms (Section II-B).

Section III deals with the problem of RL for Constrained Markov Decision Processes. Dealing first with the *flawless* setting, we develop a decomposition framework (Section III-B), based on hard barrier functions, that allows to decouple the safety assessment problem from the problem of maximizing rewards. This leads to a novel barrier learner algorithm, that is able to identify all state-action pairs that lead to unsafe events (Section III-C). Our analysis further shows the explicit role that the delayed consequences have in the learning process (Section III-D). Section III-F extends the RL setup to a relaxed setting in which we seek policies that allow for a finite number of unsafe events in any trajectory.

Numerical illustrations in Section IV verify our theoretical analysis for MABs (Section IV-A) and further suggest that, in the case of MDPs, learning the barrier first can aid in learning a task-oriented navigation goal later (Section IV-B).

II. Multi-armed bandits

We consider the setting of a stochastic bandit problem, with K arms indexed as $a \in \{1, \dots, K\}$. In the standard bandit problem an agent aims to devise an arm-pulling policy to optimize a reward. Here, we switch focus to the safety problem, for which we consider that pulling some the arms could be unsafe and lead to system damage or harm to the agent. Specifically, at each round $t \geq 1$ the agent pulls an arm A_t and obtains a binary-valued *damage indicator* D_t. If D_t is zero (one) this means that the action led to a safe (unsafe) result. We have that $\mathbb{E}[D_t|A_t] = \mu_{A_t}$. Each machine

is therefore characterized by its *safety parameter* μ_a. The greater this value is, the more likely it is that pulling the machine will lead to an unsafe event. The goal of the player in this setup is to identify all the machines that are μ-unsafe, which is hereby defined.

Definition 1. *Given a* safety specification $\mu \in [0, 1)$*, a machine a is said to be μ-unsafe if and only if $\mu_a > \mu$. Accordingly, a machine is μ-safe whenever $\mu_a \leq \mu$.*

The safety requirement μ is a design parameter, and is the only data that the player has access to along with the signal D_t. We will look at two distinct cases:

A) *Flawless setting* ($\mu = 0$): in this setting any machine with positive probability of giving damage (i.e. $\mu_a > 0$) is considered unsafe.
B) *Relaxed setting* ($\mu > 0$): In this setting we want to identify unsafe machines with $\mu_a > \mu$. This means that we allow "somewhat defective" machines.

We will focus first on the flawless setting, as it will allow us to build intuition on how to devise a proper Algorithm and on the values of metrics involved. The solution in this case is straightforward: let the agent pull each arm and avoid arms that have led to an unsafe event $D_t = 1$. For the second case, we will rely on building a one-sided Sequential Probability Ratio Test (SPRT) [41], that will make us try each machine a sufficient number of times; if the machine is *unsafe*, the test will eventually decide on that hypothesis.

In both cases, our goal is the same. We want to *detect* all the machines that are unsafe. To that end, let us define at each round $t \geq 1$ the candidate safe set $\mathcal{A}_t$, which contains all the arms that haven't been classified as unsafe.

Assumption 1. *Given a safety requirement μ, there are M μ-unsafe machines, where $1 \leq M \leq K$. Without loss of generality we will assume the the first M arms to be μ-unsafe (i.e. $\mu_a > \mu,\ a = 1, \dots, M$)*

Definition 2. *For each round $t \geq 1$ we define the* candidate safe set $\mathcal{A}_t$ *as the set containing all the arms that have not been classified as unsafe.*

The set $\mathcal{A}_t$ is initialized as $\mathcal{A}_0 := [K] = \{1, \dots K\}$, and sequentially trimmed down whenever an arm is found to be unsafe. To select which arm to pull at each round, we consider a *strategy* ψ, which assigns to a set of arms a probability of sampling each arm in the set. We define it next.

Definition 3 (Strategy)**.** *A strategy $\psi : 2^{[K]} \to \Delta^K$ is a function mapping sets of machines to the K-dimensional probability simplex, with the property that for any set $\mathcal{A} \subseteq [K]$ it holds that the support of $\psi(A)$ satisfies* $\operatorname{Supp}(\psi(\mathcal{A})) \subseteq \mathcal{A}$.

The requirement that $\operatorname{Supp}(\psi(\mathcal{A})) \subseteq \mathcal{A}$ ensures that (for any set $\mathcal{A}$) the strategy only assigns positive mass to arms in $\mathcal{A}$. Then, given a set $\mathcal{A}$, the probability of sampling an arm a is $\mathbb{P}_{A\sim\psi(\mathcal{A})}(A = a)$, where A is the (random) variable corresponding to the arm being sampled. We will further use the notation $\mathbb{P}_\psi(A = a)$ when the set $\mathcal{A}$ is understood from the context. Thus, given a set $\mathcal{A}_t$ at time $t \geq 0$, ψ induces a probability over a random action, A_t, to be taken with probability $\mathbb{P}_\psi(A_t = a)$.

We now define a class of strategies that are *sufficiently exploratory*, in the sense that they sample each arm in the candidate set with positive probability.

Definition 4 (λ-soft strategy)**.** *Given $0 < \lambda \leq 1$, a strategy ψ is called λ-soft if $\forall \mathcal{A} \subset [K]$*

$$\mathbb{P}_\psi(A = a) \geq \frac{\lambda}{|\mathcal{A}|} \quad \forall a \in \mathcal{A}\,.$$

As stated above, λ-softness is a condition on sufficient exploration of each arm. As a special case, the always-uniform strategy is 1-soft.

Although we recognize that detecting unsafe machines necessarily implies pulling from those unsafe arms, we want to have a notion of whether our decision rules choose machines in an efficient manner. It is with that goal in mind that we define at each round the *exposure*.

Definition 5 (Exposure)**.** *For $t \geq 1$ we define*

$$E_t = \sum_{\tau=1}^{t} \mathbb{1}(\mu_{A_\tau} > \mu)\,. \tag{1}$$

where $\mathbb{1}(x) = 1$ if x is true and 0 otherwise. This metric counts the rounds in which μ-unsafe machines have been pulled, regardless of whether they led to an unsafe event or not. Notice that the exposure inherits the randomness of the sequence of decisions A_t. Our results throughout this section will deal then with the expected value $\mathbb{E}[E_t]$. Ideally a good player would be one that attains low exposure—meaning it selects unsafe machines infrequently.

Remark 1. *Throughout the remainder of this section, we show that for λ-soft strategies $\mathbb{E}[E_t]$ is bounded. This marks a stark contrast with the notion of* regret*, typically studied in Bandit settings [20], where unbounded regret is unavoidable [42].*

At time t, the number of times arm a has been pulled is:

$$N_a(t) = \sum_{\tau=1}^{t} \mathbb{1}(A_\tau = a)\,. \tag{2}$$

The following lemma states that the expected exposure coincides with the sum of the expected number of pulls over the unsafe machines.

Lemma 1. $\mathbb{E}[E_t] = \sum_{a=1}^{M} \mathbb{E}[N_a(t)]$

Proof.

$$\mathbb{E}[E_t] = \sum_{\tau=1}^{t} \mathbb{E}\left[\mathbb{1}(\mu_{A_\tau} > \mu)\right] = \sum_{\tau=1}^{t}\sum_{a=1}^{K} P(A_\tau = a)\mathbb{1}(\mu_a > \mu)$$
$$= \sum_{\tau=1}^{t}\sum_{a=1}^{M} P(A_\tau = a) = \sum_{a=1}^{M}\sum_{\tau=1}^{t} \mathbb{E}\left[\mathbb{1}(A_\tau = a)\right] = \sum_{a=1}^{M} \mathbb{E}[N_a(t)]$$

□

Definition 6. *The* conservation ratio $C_{\varepsilon,t}$ *is the proportion of safe machines kept at time t*

$$C_{\varepsilon,t} := \frac{|\mathcal{A}_t \cap \mathcal{A}^\star_\varepsilon|}{|\mathcal{A}^\star_\varepsilon|} \tag{3}$$

where $\mathcal{A}_t$ is the candidate safe set and $\mathcal{A}_\varepsilon^\star$ is the set containing all arms that are $(\mu-\varepsilon)$-safe: $\mathcal{A}_\varepsilon^\star = \{a \in \mathcal{A} : \mu_a \leq \mu - \varepsilon\}$, where $0 \leq \varepsilon \ll 1$ is a non-negative slack parameter. We let $C_{\varepsilon,t} \equiv 0$ in the event $|\mathcal{A}_\varepsilon^\star| = 0$.

This ratio gives the proportion of $(\mu-\varepsilon)$-safe machines present in the candidate safe set at each time step. ($C_{\varepsilon,t}$ close to 1 is desirable). The need for the conservativeness given by ε is that we want to detect unsafe machines in finite time. This will become clearer when we discuss the *relaxed setting*, for now it suffices to assume $\varepsilon = 0$.

A. Flawless setting ($\mu = 0$)

We start with the simplest case imaginable, which is that of a rigorous safety requirement of $\mu = 0$. In this setting, any machine that has positive probability of giving damage $D_t = 1$ should be deemed unsafe. The strategy for discarding these machines is pretty straightforward: at each round t select an arm A_t following strategy $\psi(\mathcal{A}_t)$ and, if the resulting damage is $D_t = 1$, classify the machine as unsafe by taking it out of the candidate safe set $\mathcal{A}_t$. This decision rule, summarized in Algorithm 1 has two interesting properties: *i)* all unsafe machines are eventually found, and *ii)* no safe machines are discarded along the way, as the the following theorem states.

Algorithm 1: Flawless Inspector

Input: Number of arms K, strategy ψ.
`/* Initialize candidate safe set */`
$\mathcal{A}_0 = \{1, \dots, K\}$
for $t = 1, 2, \dots,$ **do**
  Pick arm $A_t \sim \psi(\mathcal{A}_t)$
  Observe damage D_t
  if $D_t = 1$ **then**
    `/* trim unsafe arm */`
    $\mathcal{A}_t \leftarrow \mathcal{A}_{t-1} \setminus \{A_t\}$
  end
end

Theorem 1. *Under Algorithm 1, for every strategy ψ, the following (in)equalities hold with probability 1 for all $t \geq 1$:*

$$\mathbb{E}[C_{0,t}] = 1 \tag{4}$$

$$\mathbb{E}[E_t] \leq \sum_{a=1}^{M} \frac{1}{\mu_a} \tag{5}$$

Proof. The proof for the safety ratio $C_{0,t}$ is immediate, since Algorithm 1 can never discard a safe machine (the event $D_t = 1$ has zero probability when pulling from a flawless arm). For the remaining equalities, let N_a be the number of pulls of the a-th arm needed to classify it as unsafe, which is well-defined (finite with probability one) for $a = 1, \dots, M$. We have that $N_a \sim \text{Geometric}(\mu_a)$, hence:

$$\mathbb{E}[N_a] = \sum_{n=1}^{\infty} P(N_a = n)n = \sum_{n=1}^{\infty} \mu_a(1-\mu_a)^{n-1}n = \frac{1}{\mu_a} \tag{6}$$

Furthermore, for all t we have $N_a(t) \leq N_a$. Taking expectation on both sides and using (6) yields $\mathbb{E}[N_a(t)] \leq \frac{1}{\mu_a}$. Combining this with the result of Lemma 1 gives (5). □

We now state that under a uniform strategy, Algorithm 1 finds all unsafe machines in expected finite time.

Theorem 2. *Let ψ be a λ-soft strategy, and assume all M unsafe machines satisfy $\mu_a \geq \mu_{low}$. Then Algorithm 1 finds all unsafe machines in time T, whence:*

$$\mathbb{E}[T] \leq \frac{1}{\lambda \mu_{low}}\big(M + (K-M)\log(M+1)\big). \tag{7}$$

Proof. The proof is in Appendix A. □

Corollary 1 (Sample-complexity bound)**.** *For any $\delta \in (0,1)$, with probability at least $1-\delta$, Algorithm 1 finds all unsafe machines after at most*

$$\left(1 + \log\frac{1}{\delta}\right)\frac{1}{\lambda\mu_{low}}\big(M + (K-M)\log(M+1)\big).$$

Proof. The stopping time T defined in the previous theorem is a sum of geometric random variables (see Appendix A). We use the bound on its expected value along with a tail-bound for sums of geometric random variables [43, Corollary 2.4] with mean $1 + \log 1/\delta$. □

B. Relaxed setting ($\mu > 0$)

We next consider the *relaxed* setting, in which we allow machines that give damage D_t with (possibly low) probability μ. This means that in order to identify unsafe machines we can no longer discard them at first sign of damage, but must rather pull from each arm and observe multiple unsafe events in order to be confident that the machine in question is defective. To check whether an arm a is defective or not we build the following hypothesis test $\mathbb{H}_a$:

$$(\mathbb{H}_a) \quad \begin{cases} \mathcal{H}_0: & \mu_a \leq \mu - \varepsilon \\ \mathcal{H}_1: & \mu_a > \mu \end{cases} \tag{8}$$

in which the alternative hypothesis is that the machine is μ-unsafe, and where we introduce the slack parameter $\varepsilon \in (0, \mu]$. In order to solve (8), we will devise a Sequential Probability Ratio Test (SPRT) which is based on Abraham Wald's seminal work [41] with the following properties:

1) If the machine is unsafe —meaning $\mathcal{H}_1$ is true— the test will terminate in expected finite pulls $\mathbb{E}[N_a]$.
2) If the machine is safe, the probability that the test —incorrectly— decides on $\mathcal{H}_1$ is upper bounded by α, where $\alpha \in (0,1)$ is the *failure tolerance* of the test.
3) Lowering failure tolerance α necessarily implies more pulls N_a to detect unsafe machines.
4) If $\mu_a \in (\mu - \varepsilon, \mu)$ the test is inconclusive.
5) The test is one-sided: it only decides on $\mathcal{H}_1$. Similar to Algorithm 1, whenever a machine is classified as unsafe it is not pulled any longer.

For a fixed arm a, let $\mathbf{d}_a(t) = \{D_\tau : A_\tau = a, \tau \leq t\}$ be the (binary) sequence of outcomes of the a-th machine up to time t. The sequential probability ratio test relies on computing the log-likelihood ratio at each time step:

$$\Lambda_a(t) = \log \frac{f_\mu(\mathbf{d}_a(t))}{f_{\mu-\varepsilon}(\mathbf{d}_a(t))}\,, \tag{9}$$

where f_μ and $f_{\mu-\varepsilon}$ are the likelihood that the sequence $\mathbf{d}_a(t)$ came from independent Bernoulli trials of success rate μ and $\mu-\epsilon$ respectively. The test terminates by declaring $\mathcal{H}_1$ whenever

$$\Lambda_a(t) \geq \log(1/\alpha)\,. \tag{10}$$

By means of sufficient statistics, $\Lambda_a(T)$ can be written as a function of both k, the total number of outcomes of $D_t = 1$ and $N_a(T)$, the total number of pulls up to time T. For a particular single arm the testing procedure is as follows. For each round $t \geq 1$ pull the arm and (given μ and ε) update the log-likelihood in (9). If (10) holds, then terminate the test, otherwise observe another sample D_t and repeat.

The following two lemmas state the desired behavior of the SPRT. Namely, that *i)* if the machine is unsafe, the SPRT will declare $\mathcal{H}_1$ with probability 1, *ii)* if the machine is safe, the SPRT will (incorrectly) declare $\mathcal{H}_1$ with probability less than or equal to α, and *iii)* the time of detection for unsafe machines is finite in expectation, and is well characterized in terms of the design parameters μ, ε and α.

Lemma 2. *For a fixed arm a of parameter μ_a, consider the sequential probability ratio test defined by* (8)–(10)*, where μ, ε and α are given. Then:*

i) If $\mathcal{H}_1$ is true, the test will (correctly) declare $\mathcal{H}_1$ with probability 1.

ii) If $\mathcal{H}_0$ is true, the test will keep going indefinitely with probability $\geq 1-\alpha$

Proof. The proof is in the Appendix B of [44]. □

Lemma 3. *For a fixed arm a of parameter μ_a, consider the sequential probability ratio test defined by* (8)–(10)*, where μ, ε and α are given. Then, if the alternative $\mathcal{H}_1$ is true, the test is expected to terminate after T_a steps, with:*

$$\mathbb{E}[T_a] \leq 1 + \frac{\log(1/\alpha)}{\mathrm{kl}(\mu, \mu-\varepsilon)}\,, \tag{11}$$

where $\mathrm{kl}(\mu, \mu-\varepsilon)$ *is the Kullback-Leibler divergence between Bernoulli distributions:*

$$\mathrm{kl}(\mu, \mu-\varepsilon) = \mu \log \frac{\mu}{\mu-\varepsilon} + (1-\mu)\log\frac{1-\mu}{1-\mu+\varepsilon}\,. \tag{12}$$

Proof. The proof is in the Appendix B of [44]. □

Remark 2. *The preceding lemma elucidates the need for the slack parameter ε: separating the two limiting distributions enables termination of the test under $\mathcal{H}_1$ in finite time. It also unveils two fundamental trade-offs. Firstly, if the distance between the limiting distributions increases (by enlarging ε), then the test detects unsafe machines faster. However, it becomes inconclusive over a larger proportion of the machines, since nothing can be assured in the region $\mu_a \in (\mu-\varepsilon, \mu)$. Secondly, α can be increased in order to detect unsafe machines faster, though this comes at the cost of declaring $\mathcal{H}_1$ for a larger proportion of the safe machines.*

Algorithm 2: Relaxed Inspector

Input: Number of arms K, strategy ψ, requirement μ, slack ε, tolerance α.

`/* Init. safe set and ratios */`

$\mathcal{A}_0 = \{1, \dots, K\}$, $\Lambda_a = 0 \;\; \forall a = 1, \dots, K$

for $t = 1, 2 \dots$ **do**

  Pick arm $A_t \sim \psi(\mathcal{A}_t)$

  Observe damage D_t

  `/* Update log-likelihood ratio */`

  $\Lambda_{A_t} \leftarrow \Lambda_{A_t} + \log \frac{f_\mu(D_t)}{f_{\mu-\varepsilon}(D_t)}$

  if $\Lambda_{A_t} \geq \log(1/\alpha)$ **then**

    `/* SPRT ends, trim unsafe arm */`

    $\mathcal{A}_t \leftarrow \mathcal{A}_{t-1} \setminus \{A_t\}$

  end

end

We build Algorithm 2 based on this SPRT, and state its main properties in the following theorem.

Theorem 3. *Under Algorithm 2, for every strategy psi, the following inequalities hold with probability 1 for all t:*

$$\mathbb{E}[C_{\varepsilon,t}] \geq 1-\alpha \tag{13}$$

$$\mathbb{E}[E_t] \leq M\left(1 + \frac{\log(1/\alpha)}{\mathrm{kl}(\mu, \mu-\varepsilon)}\right) \tag{14}$$

where $\mathrm{kl}(\mu, \mu-\varepsilon)$ *is the Kullback-Leibler divergence between Bernoulli distributions* (12)*.*

Proof. The proof follows from Lemma 2 and Lemma 3. □

In the same spirit as for the flawless setting, we now state that under a uniform strategy Algorithm 2 finds all unsafe machines in expected finite time.

Theorem 4. *Consider a λ-soft strategy ψ. Let T be the time it takes for Algorithm 2 to detect all the unsafe machines under ψ. Then:*

$$\mathbb{E}[T] \leq \frac{M(K-M+1)}{\lambda}\left(1 + \frac{\log(1/\alpha)}{\mathrm{kl}(\mu, \mu-\varepsilon)}\right)\,.$$

Proof. The proof can be found in the Appendix B. □

We end this section by *uniting* the flawless and relaxed settings, arguing that Algorithm 1 can be seen as a particular case of the SPRT used in Algorithm 2.

Proposition 1. *Given fixed $\mu \in (0,1)$ and $\varepsilon = \beta\mu$. When $\beta \to 1^-$, Algorithm 2 with any $\alpha > 0$ reduces to Algorithm 1.*

Proof. The flawless setting only allows for perfectly safe machines, discarding any arm at first sign of damage $D_t = 1$. We will show that this coincides with a Sequential Probability Ratio Test that compares $\mathcal{H}_0 : \mu_a \leq 0$ vs. $\mathcal{H}_1 : \mu_a > \mu$. This will hold for any $\mu \in (0,1)$ and for all $\alpha > 0$.

For a fixed μ, consider the test defined in (8) with $\varepsilon = \beta\mu$. As $\beta \to 1^-$, the null hypothesis $\mathcal{H}_0$ becomes $\mu_a = 0$. We show that the log-likelihood ratio $\Lambda_a(t)$ goes to infinity at first sign of damage, thus declaring $\mathcal{H}_1$ and terminating the SPRT. A sufficient statistic for computing the log-likelihood ratio $\Lambda_a(t)$

is counting the k outcomes of $D_\tau = 1$ in a total of t pulls. Then $\Lambda_a(t) = k \log \frac{\mu}{\mu-\varepsilon} + (t-k) \log \frac{1-\mu}{1-\mu+\varepsilon}$. Writing $\varepsilon = \beta\mu$, $\Lambda_a(T) = t \log \frac{1-\mu}{1-\mu(1-\beta)} + k \log \frac{1-(1-\beta)\mu}{(1-\beta)(1-\mu)} \underset{\beta\to 1^-}{\longrightarrow} \infty \ \forall k > 0$. Then the SPRT decides on the alternative $\mathcal{H}_1$ at first sign of damage, no matter how small α is. □

III. Assured Reinforcement Learning

This section builds on the insights given by the MABs to detect and discard unsafe policies in the context of RL. Once more, we focus on two different settings. First the *flawless setting*, which seeks policies whose probability of encountering damage at any time ($D_t = 1$) is zero. Then, the *relaxed setting*, which allows for a limited number of unsafe events in a single trajectory.

A. Problem formulation and outline

Consider a Markov Decision Process $\mathcal{M}$ with finite state space $\mathcal{S}$, finite action space $\mathcal{A}$, a reward set $\mathcal{R}$, and a damage indicator $D_t \in \{0, 1\}$. A transition kernel p specifies the conditional transition probability $p(s', r, d \mid s, a) := P(S_{t+1} = s', R_{t+1} = r, D_{t+1} = d \mid S_t = s, A_t = a)$, from state $s \in \mathcal{S}$ and under action $a \in \mathcal{A}$, to state $s' \in \mathcal{S}$ resulting in a reward $r \in \mathcal{R}$ and a damage indicator d. We define the *assured reinforcement learning problem* as follows:

$$V^*(s) := \max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s \right] \tag{15a}$$

$$\text{s.t.:} \quad \mathbb{P}_\pi \left(\sum_{t=0}^{\infty} D_{t+1} \leq M \mid S_0 = s \right) = 1 \tag{15b}$$

where $0 < \gamma < 1$ is a discount factor, $M \in \mathbb{N}$, and $\mathbb{E}_\pi$ and $\mathbb{P}_\pi$ denote the expectation and probability with respect to the distribution induced by π. We call this an assured RL problem because (15b) is a constraint that must be satisfied w.p.1. That constraint reads as: "under policy π, starting from s, no trajectory can accumulate more than M units of damage". We call this quantity M an *allowable budget*.

Analyzing this novel constraint calls for some work, and we subdivide this in two scenarios. The case $M = 0$ (the flawless setting) will be considered first, and we give great breadth to its analysis. We show how to obtain feasible policies in this case by developing a barrier function (Section III-B) that encodes for constraint satisfaction. This barrier can be learned independently of the reward process (Theorem 5), and we show that the optimal barrier function can be attained in expected finite time under a Barrier-learner algorithm (sections III-C and III-D). Since the optimal barrier characterizes the set of feasible policies (Remark 3), all feasible policies are found in finite time (similarly as in the previous section). We then present an *assured* version of the Q-learning algorithm that makes use of this barrier. Finally, in Subsection III-F we address the *relaxed setting* ($M > 0$), showing that it can be reduced to the flawless setting in a suitably augmented MDP.

B. Value function decomposition

As argued previously, we will focus firstly on the flawless setting ($M = 0$ in (15)), which amounts to:

$$V^*(s) := \max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s \right] \tag{16a}$$

$$\text{s.t.:} \quad \mathbb{P}_\pi \left(\sum_{t=0}^{\infty} D_{t+1} \leq 0 \mid S_0 = s \right) = 1 \tag{16b}$$

Notice that, since D_t only takes values 0 or 1, (16b) can be equivalently put in either of the following two ways:

$$\text{(16b)} \iff \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} D_{t+1} \mid S_0 = s \right] \leq 0 \tag{17}$$

$$\iff \mathbb{P}_\pi \left(D_{t+1} = 0 \mid S_0 = s \right) = 1 \quad \forall t. \tag{18}$$

The representation (17) is a cumulative constraint in expectation, which could be put in Lagrangian form (see e.g., [16] [34]) in order to solve a primal-dual problem. We take an alternative approach that will lead to finite time detection, and resort to (18) instead. Our goal then is to solve:

$$V^*(s) := \max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s \right] \tag{19a}$$

$$\text{s.t.:} \quad \mathbb{P}_\pi \left(D_{t+1} = 0 \mid S_0 = s \right) = 1 \quad \forall t \tag{19b}$$

With (19b) as constraint, there is a natural way to extend the definition of *exposure* for MDPs.

Definition 7 (Exposure for MDPs)**.** *Consider an algorithm that creates a sequence of state-action pairs $(s_i, a_i)_{i=1}^t$. The exposure at time t is:*

$$E_t := \sum_{i=1}^{t} \mathbb{1} \left\{ \min_\pi \mathbb{P}_\pi \left(\bigcup_{\tau=0}^{\infty} \{ D_{\tau+1} = 1 \mid S_0 = s_i, A_0 = a_i \} \right) > 0 \right\}.$$

Intuitively, the exposure at any step i is equal to one if conditioned on starting at (s_i, a_i) the constraint (19b) does not hold for any policy. As such, it indicates whether a state-action pair (s_i, a_i) is *unsafe*. As in the bandits case, we will show that this quantity can be bounded in expected value.
Although the expression above may seem cumbersome, we will show that it can be equivalently put in terms of a hard-barrier function that encodes safety, which we develop next.

With the formulation of (19), let us define the value function V^π for a specific policy π, in which the constraints are embedded inside the expectation:

$$V^\pi(s) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \left(\gamma^t R_{t+1} + \mathbb{I}\left[D_{t+1} \right] \right) \mid S_0 = s \right] \tag{20}$$

where the hard barrier index function $\mathbb{I}[\cdot]$ takes the form:

$$\mathbb{I}\left[D_{t+1} \right] = \log\left(1 - D_{t+1} \right) = \begin{cases} 0 & \text{if } D_{t+1} = 0 \\ -\infty & \text{if } D_{t+1} = 1 \end{cases} \tag{21}$$

so that it is null when the transition is safe, and takes the value $-\infty$ in the event of an unsafe transition. Being that (21) is unbounded, expectations are defined in the sense of the Lebesgue integral for functions in the extended real line [45].

The proposed value function definition (20) will prove useful in two senses: firstly, we will show that maximizing (20) is the same as (19a)–(19b). Secondly, the additional term in (20) will allow for a barrier-based decomposition of the value function, which will aid in the learning of constraints.

Lemma 4 (Equivalence)**.** *Problem* (19) *is equivalent to the maximization of* (20)*, that is*

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\left(\gamma^t R_{t+1} + \mathbb{I}\left[D_{t+1}\right]\right) \mid S_0 = s\right] \tag{22}$$

Proof. If a policy π_0 is unfeasible for Problem (19), then $\exists t : P\left(D_{t+1}=1\right) > 0$. This non-zero probability renders the expected value in (22) to $-\infty$ for π_0. Conversely, if a policy π_1 attains a finite objective for (22), then it must necessarily hold that $D_{t+1} = 0$ almost surely $\forall t$, and hence π_1 is feasible for (19). Therefore the feasible set of (19) coincides with the set of policies that obtain a finite value for (22). Lastly, any policy in any of these sets must satisfy $\log\left(1 - D_{t+1}\right) = 0\, \forall t$, almost surely, in which case the function being maximized is the same. Then the optimal sets of the two problems coincide. □

While solving (19) is of our utmost interest, we have just shown that, to this end, we can solve (22) instead. In what follows we will take this one step further, and show that (20) admits a *barrier-based decomposition* and can be cast as the sum of two value functions: one that checks only whether the policy in consideration is feasible (which will be the main focus of this work) and one that optimizes the return, provided the policy is feasible. The main idea behind this decoupling being that the search for feasible policies will be, in practice, an easier task to undergo.

To this end we define an auxiliary *hard-barrier* value function H^{π} that will relate to V^{π}:

$$H^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\log\left(1 - D_{t+1}\right) \mid S_0 = s\right] \tag{23}$$

We proceed similarly for the action-value function Q^{π} and its barrier counterpart B^{π}:

$$\begin{aligned} Q^{\pi}(s,a) &= \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\left(\gamma^t R_{t+1} + \mathbb{I}\left[D_{t+1}\right]\right) \middle| S_0 = s, A_0 = a\right] \\ B^{\pi}(s,a) &= \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\log\left(1 - D_{t+1}\right) \middle| S_0 = s, A_0 = a\right] \end{aligned} \tag{24}$$

Our original goal is to find policies that are optimal for (22) for each possible state. By contrast, maximizing (23) aims to find *safe* policies, in the sense that they achieve a finite value in (22). The main idea underpinning our work is that we can jointly work on optimizing (23), which reduces the search over the policy space, while at the same time maximizing the return present in (22). In the following Theorem we establish a fundamental separation principle between the value functions and their auxiliary counterparts.

Theorem 5 (Separation principle)**.** *Assume rewards R_{t+1} are bounded almost surely for all t and the discount factor satisfies $\gamma < 1$. Then, for every policy π:*

$$V^{\pi}(s) = V^{\pi}(s) + H^{\pi}(s) \tag{25}$$

$$Q^{\pi}(s,a) = Q^{\pi}(s,a) + B^{\pi}(s,a) \tag{26}$$

Proof. We shall prove (25) only, since the proof for (26) is alike. The following identities hold, as explained below.

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\left(\gamma^t R_{t+1} + \log\left(1 - D_{t+1}\right)\right) \mid S_0 = s\right] \\ &= \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\left(\gamma^t R_{t+1} + \log\left(1 - D_{t+1}\right)\right) \mid S_0 = s\right] \qquad (27) \\ &+ \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\log\left(1 - D_{t+1}\right) \mid S_0 = s\right] \qquad (28) \end{aligned}$$

To show that $V^{\pi}(s)$ can be separated in (27) and (28), first suppose policy π is feasible for Problem (22), in the sense that it attains a finite-valued objective. This necessarily implies that $D_{t+1} = 0 \;\; a.s. \;\; \forall t$, which makes the second term in (27) vanish. Conversely, suppose that the policy in consideration is infeasible. This together with the fact that rewards are bounded almost surely yields $V^{\pi}(s) = -\infty$, which is the same value attained by both (27) and (28). □

The preceding result implies non-trivial consequences. If the learning agent can interact with the environment and have access to rewards R_{t+1} and queries of whether a transition has been safe (i.e. $D_{t+1} = 0$), then it can separately learn both $Q^{\pi}(s,a)$ and $B^{\pi}(s,a)$. This is discussed in the next remark.

Remark 3 (Properties of the optimal barrier function B^*)**.**

$$\begin{aligned} B^*(s,a) &= \max_{\pi} B^{\pi}(s,a) \\ &= \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\log\left(1 - D_{t+1}\right) \mid S_0 = s, A_0 = a\right] \end{aligned} \tag{29}$$

By definition, the entries of B^* will either be 0 or $-\infty$. Having $B^*(s,a) = -\infty$ means that starting from (s,a), no policy is safe, in the sense that there is an (albeit small) non-zero probability of encountering an unsafe event $D_{t+1} = 1$, regardless of the actions taken later. Conversely, if $B^*(s,a) = 0$ then, upon starting from (s,a), there exists (at least one) policy that guarantees that no damage will be seen in the future. As such, the optimal barrier function B^* completely characterizes the set of feasible, stationary policies Π_{safe}:

$$\Pi_{safe} = \{\pi : \pi(a|s) = 0 \text{ whenever } B^*(s,a) = -\infty\} \tag{30}$$

In this sense, prior learning of B^* helps constrain the search of other known algorithms to only feasible policies.

Since the optimal barrier function characterizes the safety of any state-action pair, it can be used to express exposure for MDPs more succintly (c.f. Definition 7):

Remark 4 (Exposure in MDPs: alternate representation)**.** *The exposure at time $t \geq 1$ is:*

$$E_t = \sum_{\tau=1}^{t} \mathbb{1}\left(B^*(S_\tau, A_\tau) = -\infty\right).$$

C. Barrier learning

We now focus on the feasibility problem of learning B^* from data. To that end, we state the following optimality condition in the standard form of the Bellman's equations.

Theorem 6 (Bellman equation for $B^*(s,a)$)**.** *The optimal barrier function satisfies*

$$B^*(s,a) = \mathbb{E}\left[\mathbb{I}\left[D_{t+1}\right] + \max_{a' \in \mathcal{A}} B^*(S_{t+1}, a') \middle| S_t = s, A_t = a\right], \tag{31}$$

where the expectation is taken with respect to the damage and next-state transition given by the MDP.

Proof. It follows from Proposition 4.1.1 in [46, pp. 217] with the value function being *minimized*, and assuming possibly unbounded, non-negative costs $C(S_t, A_t) = -\mathbb{I}[D_{t+1}]$. □

Besides providing a certificate for optimality, the Bellman Equations for $B^*(s,a)$ hint towards a stochastic iterative algorithm to optimize (29) from data, the same way the Q-learning algorithm is derived from the standard unconstrained value function [46]. We will elaborate on this stochastic algorithm next. As introduced in (29), our goal is to learn a *safe* policy π with an associated optimal Barrier function $B^*(s,a)$ that encodes the trajectories that satisfy the constraints at all times w.p.1. For this purpose, we first devise the following iterative algorithm that attempts to reach a fixed point satisfying (31)

$$B_{k+1}(s,a) = \mathbb{E}\left[\mathbb{I}\left[D_{t+1}\right] + \max_{a' \in \mathcal{A}} B_k(S_{t+1}, a') \middle| S_t = s, A_t = a\right]. \tag{32}$$

Next, we appeal to the stochastic approximation machinery [47] to drop the unknown expectations yielding a data driven version of (32). The resulting stochastic update is given next.

Algorithm 3: `barrier_update`

Input: B-function and $(s_t, a_t, s_{t+1}, d_{t+1})$ tuple
Output: Barrier-function B

$$B(s_t, a_t) \leftarrow B(s_t, a_t) + \log(1 - d_{t+1}) + \max_{a'} B(s_{t+1}, a')$$

return B

The update in Algorithm 3 incorporates the information carried in d_{t+1}, which signals whether the constraint has been violated or not during the transition from time t to $t+1$. Moreover, the update does not only consider immediate violations, but also the future effect of the action a_t that is summarized in the second term $\max_{a'} B(s_{t+1}, a')$. This *bootstrapping* mechanism leverages on stationarity to collect damage information from all previous state transitions, and summarize it in $B(s_{t+1}, a')$ which predicts long-run future effect of the state-action pairs at time $t+1$. Thus, by repeating the update in Algorithm 3 with new data coming from successive system interactions, an agent can synthesize the whole information about all past constraint violations in the barrier function $B(s,a)$ for unveiling the set of unsafe policies.

We turn now to the details of this iterative algorithm and to its performance guarantees.

D. Performance analysis of Barrier-Learner

First, we consider a simple barrier learner algorithm where one can query on any state-action pair and sample transitions $(s,a) \to (s',d)$ according to the MDP kernel. The barrier learner is shown in Algorithm 4. Our analysis shows that the expected queries/samples required until all unsafe state-action pairs are detected is finite.

Algorithm 4: Barrier Learner Algorithm

Data: Constrained Markov Decision Process $\mathcal{M}$
Result: Optimal action-value function B^*
Initialize $B^{(0)}(s,a) = 0, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
for $t = 0, 1, \cdots$ **do**
  Draw $(s_t, a_t) \sim \mathrm{Unif}(\{(s,a) : B^{(t)}(s,a) \neq -\infty\})$
  Sample transition (s_t, a_t, s'_t, d_t) according to
  $P(S_1 = s'_t, D_1 = d_t | S_0 = s_t, A_0 = a_t)$
  $B^{(t+1)} \leftarrow$ `barrier_update` $(B^{(t)}, s_t, a_t, s'_t, d_t)$
end

In an MDP, an (s,a) pair is unsafe ($B^*(s,a) = -\infty$) if either it immediately causes damage, i.e. $P(D_1 = d | S_0 = s, A_0 = a) > 0$, or it transitions to an unsafe state, namely $P(S_1 = s' | S_0 = s, A_0 = a) > 0$ for some s' with $B^*(s', a) = -\infty, \forall a \in \mathcal{A}$. As a result, when an (s,a) is taken, one may observe the damage after several steps. We let L be the *lag* of the MDP, which is the maximum steps one need to wait until observing the potential damage by taking an unsafe (s,a) pair. The exact definition of L is given in Appendix C. Regarding Algorithm 4, we have the following result

Theorem 7. *Given an MDP, let $\rho > 0$ be a lower bound on all non-zero transition probabilities: $\rho \leq P(s'|s,a)\ \forall (s,a,s') : P(s'|s,a) > 0$. Let T be earliest time when Algorithm 4 detects all unsafe state-action pairs, i.e. $T := \min\{t : B^{(t)} = B^*\}$. Then we have:*

$$\mathbb{E}[T] \leq (L+1)\frac{|\mathcal{S}||\mathcal{A}|}{\rho} \log\left(|\mathcal{S}||\mathcal{A}| + 1\right), \tag{33}$$

where L is the lag of the MDP.

Proof Sketch. Theorem 7 is proved in three steps. We refer the readers to Appendix C for the complete proof.

First, we reformulate the algorithm so that the sampling process of (s_t, a_t) is independent of the current progress of the $B^{(t)}$-function: at iteration t, one samples an (s_t, a_t) pair uniformly from the entire $\mathcal{S} \times \mathcal{A}$ set, then the algorithm chooses to either accept or reject the sample depending on the value of $B^{(t)}(s_t, a_t)$. This way, we turn to study the number of acceptance by the reformulated algorithm before it detects all unsafe state-action pairs. Secondly, we provide a modified algorithm with more restrictions on accepting the sample compared to the reformulated algorithm. Such restrictions allow the modified algorithm to learn unsafe state-action pairs in multiple stages: at each stage, the algorithm only allows to detect a subset of unsafe state-action pairs, but the detection

process in every stage can be viewed as a safe multi-arm bandits problem discussed in Section II.

Lastly, we derive an upper bound on the expected number queries of the modified algorithm until successfully detecting all unsafe state-action pairs, based on Theorem 2, which is also an upper bound on $\mathbb{E}[T]$. □

While this bound is admittedly loose (we use a lower-bound ρ for the transition probabilities and a provably slower surrogate algorithm with more restrictions), it serves the purpose of upholding our main claim in the paper that all unsafe policies can be detected in finite time. The resulting simplicity of this bound also lets us observe the fundamental factors adding to the detection time T. Specifically, with larger spaces $\mathcal{S}$ and $\mathcal{A}$ more exploration is needed, with a smaller ρ or longer lag L unsafe actions take longer to be revealed as damaging, all three factors adding to a longer detection time. A tighter bound is presented in Appendix C when we prove Theorem 7.

This theorem has many implications. First, recalling that the optimal barrier B^* fully characterizes the set of feasible policies (Remark 3), we obtain *all* the feasible policies in finite time. Next, as a simple corollary, we have that the *exposure* (which counts the number of unsafe interactions with the environment) is also finite in expectation.

Corollary 2 (Expected exposure in MDPs is finite)**.** *The expected exposure (see Remark 4) under Algorithm 4 satisfies:*

$$\mathbb{E}[E_t] \leq (L+1)\frac{|\mathcal{S}||\mathcal{A}|}{\rho}\log\left(|\mathcal{S}||\mathcal{A}|+1\right)$$

Proof. Note that the exposure at any time t is at most t: $E_t \leq t$ (which amounts to always sampling an unsafe (s,a) pair). In particular, at the termination time T of the algorithm we have $E_T \leq T$, and therefore $\mathbb{E}[E_T] \leq \mathbb{E}[T]$. The result follows from (33). □

We now re-use Theorem 7 to derive a sample-complexity bound, and next discuss that the dimensionality dependence is far better than what appears in current works.

Corollary 3 (Sample-complexity bound)**.** *For any $0 < \delta < 1$, with probability at least $1-\delta$, Algorithm 4 learns the optimal barrier function B^* after at most T_δ steps, with*

$$T_\delta = \left(1+\log\frac{1}{\delta}\right)(L+1)\frac{|\mathcal{S}||\mathcal{A}|}{\rho}\log\left(|\mathcal{S}||\mathcal{A}|+1\right).$$

Proof. The stopping time T defined in Theorem 7 is a sum of geometric random variables (see Appendix C). We use the fact that it is bounded in expectation by (33) together with a tail-bound for sums of geometric random variables [43, Corollary 2.4] with mean $1+\log 1/\delta$. □

Remark 5. *The preceding corollary shows that the optimal barrier function—and hence the set of feasible policies—can be learned with $\mathcal{O}\left(\log\frac{1}{\delta}(L+1)\frac{|\mathcal{S}||\mathcal{A}|}{\rho}\log|\mathcal{S}||\mathcal{A}|\right)$ samples. This is much more efficient than learning an ϵ-optimal policy [48], which requires $\mathcal{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\epsilon^2}\log\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)\epsilon\delta}\right)$ samples. Learning an optimal policy requires good* estimation *of the optimal value at each state, which brings the terms $\frac{1}{\epsilon}$ and $\frac{1}{1-\gamma}$ into play. Our framework, in contrast, cares only for* detection *of unsafe state-action pairs, which can be done much faster.*

Once the optimal barrier function is known, one can optimize a policy over a smaller region, which corresponds to the reduced set of safe states $\mathcal{S}_{\texttt{safe}}$ and safe actions $\mathcal{A}_{\texttt{safe}}$. If the cardinality of these sets is much smaller than $\mathcal{S}$ and $\mathcal{A}$, the optimization process will be indeed faster. Along this line, we now illustrate how to learn both the safe region and an optimal policy by combining the barrier-learner with Q-learning.

E. Learning safely: Q-learning with a barrier

Now that we have introduced this data-driven strategy for securing the environment, let us turn back our attention to the implications of the separation principle in Theorem 5. The separation principle (26) can be used to embed the safety information provided by $B(s,a)$ in the Q-function $Q(s,a)$. Hence, we identify that the condition $Q(s,a) = -\infty$ is equivalent to $B(s,a) = -\infty$ and this propagates information about safety from successor states. For those which do satisfy the constraints we have $B(s,a) = 0$, and then $Q(s,a)$ will carry the information of the observed rewards. The following update is complementary to Algorithm 4, and amounts to the standard Q-learning algorithm for maximizing rewards at safe pairs of states and actions.

Algorithm 5: `q_update`

Data: Step size η, discount factor γ
Input: Functions Q, B, and $(s_t, a_t, s_{t+1}, r_{t+1})$
Output: Q-function

$$Q(s_t,a_t) \leftarrow (1-\eta)Q(s_t,a_t) + \eta\left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a')\right)$$
$$Q(s_t,a_t) \leftarrow B(s_t,a_t) + Q(s_t,a_t)$$

return Q

Next we specify how to use Algorithm 3 to ensure safety, and demonstrate the sample complexity of the learning process. After that, we combine algorithms 3 and 5 with the goal of safely maximizing rewards. We will borrow the well-known convergence results of Q-learning [49] together with our separation principle in Theorem 5 to provide performance guarantees for our novel Assured Q-learning algorithm.

The Barrier Learning Algorithm 4 stands alone as a data-driven method to assess safety feasibility. However, our separation principle allow us to combine it with standard existing reward optimization algorithms in order to add safety. For instance, by wrapping Algorithm 4 around the acclaimed Q-learning algorithm we obtain a Generative Assured Q-learning method to maximize the rewards over the set of safe policies, as is shown in Algorithm 6.

As a corollary of Theorem 7, and borrowing the convergence results of Q-learning from [49] we establish the following result.

Corollary 4. *With finite state space $\mathcal{S}$ and action space $\mathcal{A}$, bounded rewards $R_t \leq C$, and diminishing step-sizes satisfying $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$, the iterates $Q^{(t)}$ of the*

Algorithm 6: Generative Assured Q-Learning

Data: Constrained Markov Decision Process $\mathcal{M}$
Result: Optimal action-value functions B^* and Q^*
Initialize
$B^{(0)}(s,a)=0,\ Q^{(0)}(s,a)=0 \forall (s,a)\in\mathcal{S}\times\mathcal{A}$
for $t=0,1,\cdots$ **do**
  Draw $(s_t,a_t)\sim \text{Unif}(\{(s,a):\mathcal{B}(s,a)\neq -\infty\})$
  Sample transition (s_t,a_t,s_t',d_t) according to $P(S_1=s_t',D_1=d_t|S_0=s_t,A_0=a_t)$
  $B^{(t+1)}\leftarrow \texttt{barrier\_update}(B^{(t)},s_t,a_t,s_t',d_t)$
  $Q^{(t+1)}\leftarrow \texttt{q\_update}(B^{(t)},Q^{(t)},s_t,a_t,s_t',r_t)$
end

Algorithm 6 converge almost surely to the optimal Q-function $Q^\star$ satisfying:

$$\lim_{t\to\infty} Q^{(t)}(s,a)=Q^*(s,a)\ (w.p.1)\ \ \forall (s,a)\in\mathcal{S}\times\mathcal{A}$$

with:

$$Q^*(s,a)=\mathbb{E}\left[\log(1-D_{t+1})+\max_{a'\in\mathcal{A}} Q^*(s',a')\big|S_t=s,A_t=a\right].$$

Moreover, all unsafe state-action pairs corresponding to $Q^*(s,a)=-\infty$ are detected in expected finite time $T_{s,a}$ such that:

$$Q^{(t)}(s,a)=-\infty,\ \forall\ t\geq T_{s,a},$$

and:

$$\mathbb{E}\left[T_{s,a}\right]\leq \frac{|\mathcal{S}|^2|\mathcal{A}|}{\mu}\log\left(|\mathcal{S}||\mathcal{A}|+1\right). \tag{34}$$

Proof. In order to apply the convergence results of [49] we need iterates $Q^{(t)}$ to be finite for all t. But this is guaranteed by Theorem 7 for all safe pairs (s,a) such that $B^*(s,a)=0$. Indeed, or all values such that $B^*(s,a)=0$, we have $B^{(t)}=0$ for all t and the updates of the function $Q^{(t)}$ in Algorithm 6 amount to the asynchronous Q-learning updates with finite rewards and diminishing step-size required by [49].

For the pairs such that $B^*(s,a)=-\infty$, Theorem 5 implies $Q^*(s,a)=-\infty$ and according to Theorem 7 there must exist a time instant $T_{s,a}$ satisfying (34) such that $B^{(t)}=-\infty\ \ \forall t\geq T_{s,a}$. Since by construction the q_update in Algorithms 6 and 3 implies that $Q^{(t)}=-\infty$ whenever $B^{(t)}=-\infty$, then $\lim_{t\to\infty}Q^{(t)}(s,a)=-\infty$ for all pairs (s,a) such that $B^*(s,a)=Q^*(s,a)=-\infty$. □

The previous result applies to the specific case of diminishing step-sizes and immediate restarts after episodes of length one. However, Q-learning is widely applied with longer episodes and convergence is guaranteed provided that each state-action pair is visited infinitely often. While one step episodes in Algorithm 6 were used in order to simplify the proof of Corollary 4, the numerical experiments of the next section will demonstrate that these safe convergence results carry out to an episodic version of Assured Q-learning with ϵ-greedy exploratory modes.

F. Relaxed setting

We finish this section by analyzing the relaxed setting, which corresponds to $M>0$ in (15):

$$\max_{\pi}\ \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1}\ \middle|\ S_0=s\right] \tag{35a}$$

$$\text{s.t.: } \mathbb{P}_\pi\left(\sum_{t=0}^{\infty} D_{t+1}\leq M\mid S_0=s\right)=1;\quad M>0. \tag{35b}$$

This setting differs from the one analyzed so far, in the sense that we allow *at most* M units of damage in any given trajectory. To solve (35) we will resort to state-augmentation, tracking how much damage an agent has received so far. We define the variable to augment into the state next.

Definition 8 (Safety budget)**.** *The safety budget K_t at time t is defined and evolves as*

$$K_0:=M \tag{36a}$$

$$K_{t+1}=K_t-D_t \tag{36b}$$

Here, K_t specifies the remaining budget at time t, that is, how many more units of damage an agent can sustain and still satisfy (35b). We consider a state-augmented MDP $\tilde{\mathcal{M}}$ with state $\tilde{S}_t=(S_t,K_t)$ (notice with (36b) transitions are still Markovian), and observe that (35) can be equivalently formulated as:

$$\max_{\tilde\pi}\mathbb{E}_{\tilde\pi}\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1}\ \middle|\ S_0=s,K_0=M\right] \tag{37a}$$

$$\text{s.t.: } \mathbb{P}_{\tilde\pi}\left(K_{t+1}\geq 0\mid S_0=s,K_0=M\right)=1\ \ \forall t\geq 0 \tag{37b}$$

where the notation $\tilde\pi$ stresses that this is a policy on the augmented MDP $\tilde{\mathcal{M}}$. By defining a new binary damage signal $\tilde D_{t+1}$ on $\tilde{\mathcal{M}}$ we can rewrite the constraint (37b) as follows:

$$K_{t+1}\geq 0\iff \tilde D_{t+1}:=\mathbb{1}\left\{K_{t+1}<0\right\}=0 \tag{38}$$

Now, outstandingly, the previous state-augmented problem can be equivalently put in the following form, which *fits* the flawless setting of the beginning of the section:

$$\max_{\tilde\pi}\mathbb{E}_{\tilde\pi}\left[\sum_{t=0}^{\infty}\gamma^t R_{t+1}\ \middle|\ S_0=s,K_0=M\right] \tag{39a}$$

$$\text{s.t.: } \mathbb{P}_{\tilde\pi}\left(\tilde D_{t+1}=0\mid S_0=s,K_0=M\right)=1\ \forall t\geq 0 \tag{39b}$$

We focus then on (39), and state its main properties in the following theorem.

Theorem 8 (Stationarity and equivalence)**.**

1) *If* (39) *is feasible, there exists an optimal policy that is stationary: that is $\tilde\pi^*(\cdot|s,k)$.*
2) *In that case, $\tilde\pi^*$ is also optimal for* (35)*.*

Proof. To prove (1), swap (39b) for the equivalent constraint $\mathbb{E}_{\tilde\pi}\left[\sum_{t=0}^{\infty}\gamma^t\tilde D_{t+1}|S_0=s,K_0=M\right]\leq 0$. This fits the standard formulation for CMDPs, for which the set of stationary policies is *complete* [31]. Then, if the problem is feasible, there exists at least one stationary optimal policy. To prove (2), we

refer the reader to Lemma 5 and Lemma 6 in [2], where the equivalence between (35) and (39) is shown. □

Due to Theorem 8 and the fact that (39) fits the formulation of the beginning of the section, we can similarly define an extended action-value function $\tilde{Q}^{\tilde{\pi}}(s,k,a) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\left(\gamma^t R_{t+1} + \mathbb{I}\left[\tilde{D}_{t+1}\right]\right)\middle| S_0 = s, K_0 = k, A_0 = a\right]$, the hard-barrier $\tilde{B}^{\tilde{\pi}}(s,k,a)$, and the separation principle still holds. Then, under this setting learning feasibility would be akin to learning the optimal barrier $\tilde{B}^*(s,k,a) \quad \forall (s,k,a) \in \mathcal{S} \times [M] \times \mathcal{A}$ where $[M] = \{0,1,\dots,M\}$. This can still be done in expected finite time by applying the results of Theorem 7, along with finite exposure, defined in the extended MDP. There is a seemingly big price to pay in this case: the dimensionality increase of needing to learn $\tilde{B}^*$ for each possible budget k. This however, can be circumvented, and we refer the reader to [2] for details.

IV. Numerical Experiments

We now proceed to some numerical experiments that back up the results presented throughout the paper. We first focus on the multi-armed bandit setup and on the problem of detecting unsafe machines under a uniform strategy. Later we tackle learning the optimal barrier in a continuous control problem, and how learning this barrier—i.e. feasibility—first can make learning a task easier later.

A. Multi-armed bandits

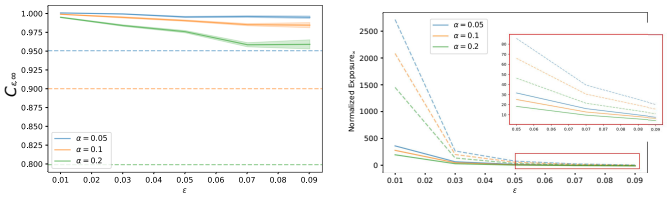
We illustrate the performance of the Relaxed Inspector (Algorithm 2) on a setup of $K = 1000$ arms, with a safety requirement of $\mu = \frac{1}{10}$. Each arm's true safety parameter is uniformly sampled between 0 and $\frac{1}{5}$. This means that if $\epsilon \approx 0$, around half of the machines are safe and the other half unsafe. We run the Relaxed Inspector on this setting under a uniform strategy, for varying levels of both the failure tolerance α and the slack ϵ. After all unsafe machines have been detected —which happens and is certified to be done in finite time in virtue of Theorem 3— we consider $C_{\varepsilon,\infty}$, the final conservation ratio and the normalized final exposure $\frac{1}{K}E_{\infty}$. Figure 1 shows both metrics for varying α and ϵ. The curves in these figures show the average value obtained after 16 independent runs. Shaded intervals correspond to $\pm\sigma/\sqrt{16}$, with σ being the sample deviation.

These figures certify the intrinsic trade-off in our methodology: if a large value of α is used, one can obtain low exposure E_t (less pulls of unsafe machines), but at the cost of discarding more safe machines (smaller conservation ratio $C_{\varepsilon,t}$). For further examples we refer the reader to [1].

B. MDPs

We consider a robot navigating in 2d-space with constant velocity module $v = 0.5m/s$. The robot can change its angle via first-order tracking dynamics, and the system evolves according to the following equations:

$$\begin{cases} \dot{x} = v.\cos\theta \\ \dot{y} = v.\sin\theta \\ \dot{\theta} = -(a-\theta) \end{cases} \tag{40}$$



(a) (b)

Fig. 1: Top: Final safety ratio $C_{\varepsilon,\infty}$ for $\mu = 0.1$ as a function of ϵ, for varying α. More safe machines are kept when using small values of ϵ and α, but this in turn implies longer training time. Bottom: Normalized final exposure for $\mu = 0.1$ as a function of ϵ, for varying tolerance level α. Larger values of α and ϵ achieve lower handicap (which implies faster detection).

where x and y are the horizontal positions, θ is the angle with respect to the horizontal, $s = [x, y, \theta]$ is the state of the system and the action a is an angle setpoint.

The (x, y)-space is a 5×5 square with an obstacle in the middle, as depicted in the top-left of Fig. 2. Bumping into either the obstacle or any of the outer walls is *unsafe*, and results in damage $D_{t+1} = 1$. At each time step, the agent decides an action $a \in [\theta_0 - \pi, \theta_0 + \pi]$ where θ_0 is the angle at the beginning of the time interval. This action is input to the system (40) for $\Delta T = 0.5s$. We uniformly discretize each component of the state-action space into $N_x = N_y = 21, N_\theta = N_a = 8$ values respectively.

1) Learning the barrier: We learn the barrier running episodes in which the initial state is picked uniformly at random, and the episode finishes either after 100 timesteps or if the agent receives damage (bumps into the object or goes out of bounds). At each step, the agent takes a random action over the presumed safe ones: those for which $B(s,a) = 0$. Figure 2 shows the learned barrier after 2×10^6 episodes.

2) Knowing the barrier accelerates learning: In this task-oriented version of the previous environment, the goal is to reach a region of the space, a circle of radius 0.5 centered at coordinates $(4.5, 0.5)$. Each episode starts in position $(0.5, 0.5)$ with heading $\theta = 0$. At each time step the reward R_t is minus the distance to the center of the goal. Bumping into the wall ends the episode with additional -100 reward. Reaching the goal ends the episode with additional $+100$ reward

We compare a standard Q-learning agent versus its *assured* counterpart: an agent that has previously learned the barrier of Fig. 2, and that only takes (presumably) safe actions. At each step, the standard agent follows an ϵ-greedy policy, while the assured agent follows a *safe* ϵ-greedy policy, only taking actions over the set of presumed-to-be safe actions ($B(s,a) = 0$). At each step t we observe a tuple (S, A, S', R, D) and update the $Q-$function as $Q_{t+1}(S,A) = (1-\alpha)Q_t(S,A) + \alpha\left(R + \gamma \max_{a'} Q_t(S', A')\right)$. Both agents use $\epsilon = 0.1$, $\alpha = 0.1$, $\gamma = 0.99$. Fig. 3 shows 100 greedy trajectories obtained with each agent during different stages of training. Assured Q-learning (on the left) rapidly learns to reach the goal, always succeeding. On the other hand, some trajectories of standard Q-learning (on the right) fail against the wall. The assured

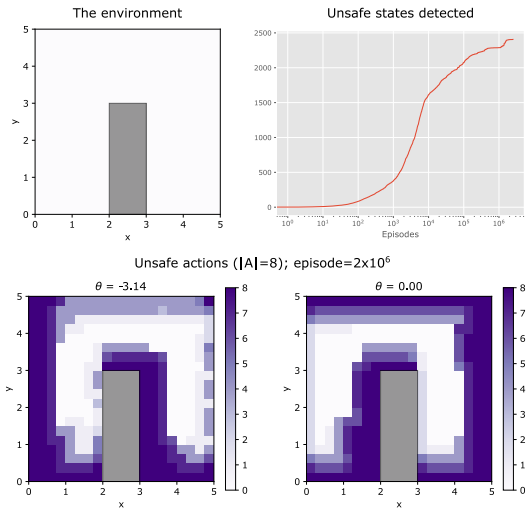


Fig. 2: Top-left: the environment. An agent navigates through it, and gets damage if it collides onto the gray obstacle or the outer walls. Top-right: learning curve for the unsafe states detected by the barrier learner (those for which $B(s,a) = -\infty \ \forall a$). Bottom: visualization of the learned barrier after two million episodes. The left figure shows the barrier when the agent's angle is $\theta = -\pi$ (heading left); the right figure shows the barrier for an agent facing right.

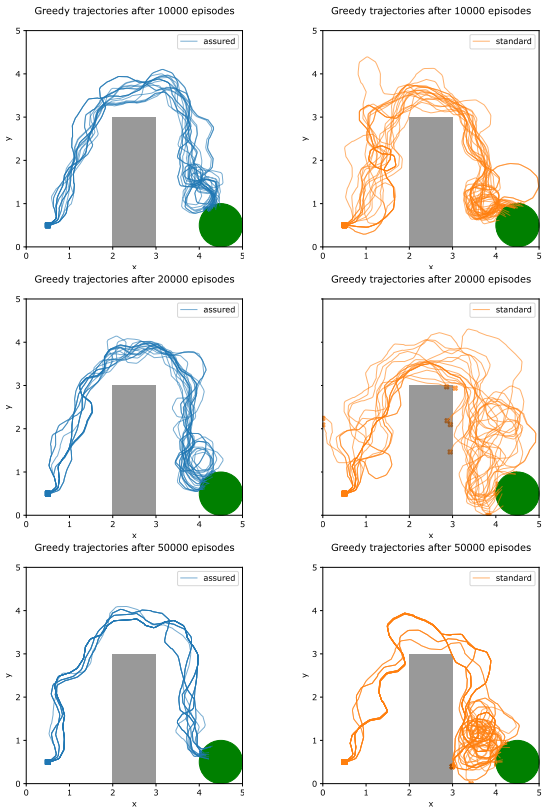


Fig. 3: Samples of greedy trajectories at different stages of training. The first column corresponds to the *assured* agent, the second one to the standard Q-learning agent.The assured agent always reaches the goal while the standard agent still fails in some trajectories. Notice the conservative nature of the assured agent, who leaves a bigger margin between itself and the obstacle.

agent is conservative, leaving a bigger gap with the obstacle.

Some quantitative metrics for these examples are shown in Fig. 4, that show the total reward and steps taken to reach the goal over these trajectories. Each column shows statistics using 100 greedy trajectories as sample points. Each box spans the first (q_1) and third (q_3) quartile, with the median shown as a solid line. Whiskers above and below each box correspond to a confidence interval depicting the inter-quartile length $1.5 \times (q_3 - q_1)$, with outliers as black circles. Green triangles represent the mean values. The value N indicates how many trajectories (out of 100) reach the goal safely.

Key takeaways from this experiment are that: *i)* the assured agent always reaches the goal safely, while the standard one sometimes fails; *ii)* when both reach the goal, the assured agent is typically faster; *iii)* the assured agent is more *conservative*, staying further away from the obstacles.

V. Conclusions

In this work we addressed the problem of learning to act safely in unknown environments. We made the case that learning safe policies is fundamentally different from learning optimal policies, and that it can be done separately and in a more efficient manner. By incorporating in the model a binary *damage signal* that indicates constraint violations, we showed that identification of all unsafe actions (MABs) and state-action pairs (MDPs) is achieved in expected finite time with probability one guarantees. These results imply that the learner is not indefinitely exposed to damage, and could aid in the design of new algorithms that rapidly learn to act safely while jointly optimizing returns. Our experimental results for MDPs suggest that our algorithm obtains good performance in a continuous-state dynamical system, making it potentially useful for control applications.

Appendix

A. Proof of Theorem 2

We will prove the following two inequalities:

$$\mathbb{E}[T] \leq \frac{1}{\lambda \mu_{\text{low}}} \sum_{i=0}^{M-1} \frac{K-i}{M-i} \leq \frac{M + (K-M)\log(M+1)}{\lambda \mu_{\text{low}}}.$$

Proof. Each iteration of Algorithm 1 can be view as doing a Bernoulli trial with success rate being the probability of detecting an unsafe machine. This probability evolves over time, and depends on the failure probability of each arm, and on the number of unsafe machines in the candidate set.

We can decompose the time T as $T = T_1 + (T_2 - T_1) + (T_3 - T_2) + \ldots + T - T_{M-1}$, where T_i is the total time taken to detect the i-th machine. Then

$$\mathbb{E}[T] = \mathbb{E}[T_1] + \mathbb{E}[T_2 - T_1] + \ldots + \mathbb{E}[T - T_{M-1}] \quad (41)$$

We first bound $\mathbb{E}[T_1]$. When all M malfunctioning machines are in play, T_1 is just the time taken to detect one of them.

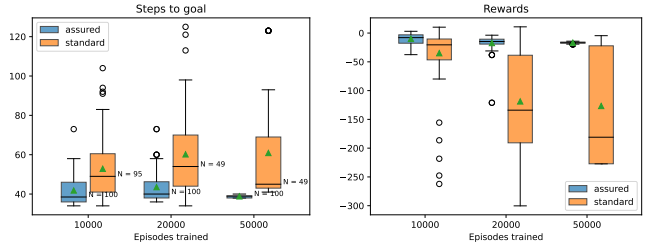


Fig. 4: Left: steps taken to reach the goal for both agents at different stages of training. The value N shows how many trajectories (out of 100) made it to the goal. The assured agent always reaches the goal ($N = 100$), usually in fewer steps; the standard agent reaches the goal half of the time ($N = 49$) after 20000 and 50000 episodes. Right: total reward along the trajectories. The assured agent's rewards are more tightly concentrated, and generally higher.

The probability of detecting the first machine is then

$$p_1 := \mathbb{P}(\text{detect first machine}) = \mathbb{P}(\text{get damage})$$
$$= \sum_{a=1}^{M} \mathbb{P}(\text{get dmg}|\text{pull } a)\mathbb{P}(\text{pull } a) \geq \frac{\lambda}{K}\sum_{a=1}^{M}\mu_a \geq \frac{\lambda M \mu_{\text{low}}}{K}$$

where in the first inequality we used the fact that the strategy ψ is λ-soft, and in the second one the lower bound on μ_a. Since $T_1 \sim Geom(p_1)$, we thus have:

$$\mathbb{E}[T_1] = \frac{1}{p_1} \leq \frac{1}{\lambda\mu_{\text{low}}} \cdot \frac{K}{M}$$

We now proceed to bound $\mathbb{E}[T_2 - T_1]$. After detecting the first machine, now p_2 is the probability of observing damage when dealing with $M-1$ malfunctioning machines over a total of $K-1$ machines. Now $T_2 - T_1 \sim Geom(p_2)$ and we get:

$$\mathbb{E}[T_2 - T_1] \leq \frac{1}{\lambda\mu_{\text{low}}} \cdot \frac{K-1}{M-1}$$

Proceeding similarly for the remaining stages, we end up bounding $\mathbb{E}[T]$ in (41) as:

$$\mathbb{E}[T] \leq \frac{1}{\lambda\mu_{\text{low}}} \sum_{i=0}^{M-1} \frac{K-i}{M-i}$$

What remains to be shown is a further upper bound on this right hand side. To that end, we manipulate the sum:

$$\sum_{i=0}^{M-1}\frac{K-i}{M-i} = M + (K-M)\sum_{i=0}^{M-1}\frac{1}{M-i}$$
$$= M + (K-M)\sum_{i=1}^{M}\frac{1}{i} \leq M + (K-M)\log(M+1)$$

where on the last inequality we used the usual bound on the harmonic series $\sum_{i=1}^{M}\frac{1}{i} < \log(M+1)$. □

B. Proof of Theorem 4

By lemma 3, we have

$$\mathbb{E}[E_T] = \sum_{a=1}^{M}\mathbb{E}[T_a] \leq M\left(1 + \frac{\log(1/\alpha)}{\text{kl}(\mu, \mu-\epsilon)}\right).$$

Then by Wald's identity [50], we have

$$\mathbb{E}[E_T] = \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}[\mathbb{1}\{\mu_{A_t} > \mu\}]\right] \geq \mathbb{E}\left[T\frac{\lambda}{K-M+1}\right],$$

where the second inequality is due to the fact that before T, the probability of sampling an unsafe machine with a λ-soft strategy is at least $\lambda/(K-M+1)$. Recalling the upper bound for $\mathbb{E}[E_T]$ in Theorem 3, one obtains

$$\mathbb{E}[T] \leq \frac{M(K-M+1)}{\lambda}\left(1 + \frac{\log(1/\alpha)}{\text{kl}(\mu, \mu-\epsilon)}\right).$$

C. Proof of Theorem 7

We prove Theorem 7 in three steps:

1) Reformulation of Algorithm 4: We reformulate Algorithm 4 as in Algorithm 7.

Algorithm 7: Barrier Learner Algorithm Reformulated

Data: Constrained Markov Decision Process $\mathcal{M}$
Initialize $B^{(0)}(s,a) = 0, \forall (s,a) \in \mathcal{S}\times\mathcal{A}$
for $\tau = 0, 1, \cdots$ **do**
  Draw $(s_\tau, a_\tau) \sim \text{Unif}(\mathcal{S}\times\mathcal{A})$
  Sample transition $(s_\tau, a_\tau, s'_\tau, d_\tau)$ according to $P(S_1 = s'_\tau, D_1 = d_\tau | S_0 = s_\tau, A_0 = a_\tau)$
  if $B^{(\tau)}(s_\tau, a_\tau) \neq -\infty$ **then**
    $B^{(\tau+1)} \leftarrow$ `barrier_update`$(B^{(\tau)}, s_\tau, a_\tau, s'_\tau, d_\tau)$
  else
    $B^{(\tau+1)} \leftarrow B^{(\tau)}$
  end
end

In the reformulated Algorithm 7, the sampling process is independent of B-function: At each iteration τ, an (s_τ, a_τ) pair is drawn uniformly from $\mathcal{S}\times\mathcal{A}$ and then a transition $(s_\tau, a_\tau, s'_\tau, d_\tau)$ is sampled according to the MDP, and the algorithm decides whether to accept such a sample depending on the value of $B(s_\tau, a_\tau)$. When we restrict ourselves to the trajectory of samples that are accepted, i.e.

$$\{(s_\tau, a_\tau, s'_\tau, d_\tau) : B^{(\tau)}(s_\tau, a_\tau) \neq -\infty,\ \tau = 0, 1, \cdots\},$$

this trajectory is also a sampled trajectory of original Algorithm 4. More importantly, for such a trajectory, the probability it appears in original Algorithm 4 is the same as the probability it appears as the accepted trajectory in Algorithm 7. With that, we define

$$T_r := \min\{\tau : B^{(\tau)} = B^*\}, \tag{42}$$

i.e. the earliest time when Algorithm 7 detects all unsafe state-action pairs, then we have

$$\mathbb{E}[T] = \mathbb{E}\left[\sum_{\tau=1}^{T_r}\mathbb{1}\{B^{(\tau)}(s_\tau, a_\tau) \neq -\infty\}\right], \tag{43}$$

where T is the earliest time when Algorithm 4 detects all unsafe state-action pairs, as defined in Theorem 7. Expectations are taken with respect to the respective sampling processes of Algorithm 4 and 7, which are different. With (43), it suffices to analyze the expected detection time of Algorithm 7.

2) Construction of modified algorithm: As discussed in Section III-D, an (s,a) pair is unsafe if either it causes damage immediately or it transitions to an unsafe state with non-zero probability. If only the latter happens for such an unsafe (s,a), then to be able to declare it unsafe, one must have already declared one of its succeeding states unsafe. To make such intuition precise, we recursively define disjoint subsets $\mathcal{S}_l, l=1,2,\cdots$ of the state space $\mathcal{S}$ as follow,

$$\mathcal{S}_1 := \{s \in \mathcal{S} : P_\pi\left(D_1 = 1 | S_0 = s\right) > 0, \forall \pi\}\,,$$
$$\mathcal{S}_l := \begin{cases} \mathcal{S}'_l, & \mathcal{S}'_l \neq \emptyset \\ \mathcal{S} \setminus \bigcup_{k<l} \mathcal{S}_k, & \mathcal{S}'_l = \emptyset \end{cases}, \tag{44}$$

where

$$\mathcal{S}'_l = \left\{ s \in \mathcal{S} \setminus \bigcup_{k<l} \mathcal{S}_k : P_\pi\left(S_1 \in \bigcup_{k<l} \mathcal{S}_k | S_0 = s\right) > 0, \forall \pi \right\}$$

Observe that for any finite MDP, its *lag* $L := \max\{l > 0 : \mathcal{S}_{l+1} \neq \emptyset\}$ is finite. Following the definition (44), $\{\mathcal{S}_l, l = 1, \cdots, L, L+1\}$ is a partition of $\mathcal{S}$. Any state $s_0 \in \bigcup_{l=1}^{L} \mathcal{S}_l$ is unsafe because starting from s_0 and under any policy π, the MDP eventually reaches a state in $\mathcal{S}_1$ with non-zero probability, then causes damage. Furthermore, any state $s_0 \in \mathcal{S}_{L+1} := \mathcal{S} \setminus \bigcup_{l=1}^{L} \mathcal{S}_l$ is safe since $\mathcal{S}'_{L+1} = \emptyset$ implies that there exists $a_0 \in \mathcal{A}$ such that $P(S_1 \in \bigcup_{k<L+1} \mathcal{S}_k | S_0 = s_0, A_0 = a_0) = 0$, i.e. taking action a_0 keeps the MDP away from the unsafe states in $\bigcup_{l=1}^{L} \mathcal{S}_l$. However, we note that a safe state can have unsafe actions and they can be detected by the Barrier Learner Algorithm. More importantly, the unsafe state sets $\mathcal{S}_l, l = 1, 2, \cdots, L$ satisfies that if all states in $\bigcup_{k<l} \mathcal{S}_k$ has been declared unsafe, any state-action pair in $\{(s,a) : s \in \mathcal{S}_l, a \in \mathcal{A}\}$, when sampled, can be declared unsafe with non-zero probability. Base on this property, we construct an modified barrier learning algorithm using the prior information on $\mathcal{S}_l, l = 1, 2, \cdots, L$. The modified algorithm is described in Algorithm 8.

The modified algorithm is similar to Algorithm 7 but it learns $\mathcal{S}_l, l = 1, 2, \cdots, L$ in order: At the beginning ($l = 1$), it only declares (s,a) pairs associated with $\mathcal{S}_1$ unsafe until all states in $\mathcal{S}_1$ are declared unsafe, after which l increases to 2. Now the algorithm only declares (s,a) pairs associated with $\mathcal{S}_2$ unsafe. Finally after all states in $\bigcup_{l=1}^{L} \mathcal{S}_l$ are declared unsafe ($l = L+1$), the algorithm starts to learn the unsafe transitions for safe states in $\mathcal{S}_{L+1}$. Similarly, we define

$$\hat{T}_r := \min\{\tau : \hat{B}^{(\tau)} = B^*\}\,, \tag{45}$$

i.e. the earliest time when Algorithm 8 detects all unsafe state-action pairs. Since the modified algorithm is more restrictive on declaring unsafe state-action pair, the expected detection time of the modified algorithm is no less than that of Algorithm 7, as stated in the following claim.

Claim 1. *Given an MDP, let T_r and $\hat{T}_r$ be the earliest times when Algorithm 7 and Algorithm 8, detect all unsafe state-action pairs in this MDP, respectively, as defined in* (42) *and* (45)*. Then, $\mathbb{E}\left[\sum_{\tau=1}^{\hat{T}_r} \mathbb{1}\{\hat{B}^{(\tau)}(s_\tau, a_\tau) \neq -\infty\}\right]$ is lower-bounded by $\mathbb{E}\left[\sum_{\tau=1}^{T_r} \mathbb{1}\{B^{(\tau)}(s_\tau, a_\tau) \neq -\infty\}\right]$, where expectations are w.r.t. $\{(s_\tau, a_\tau, s'_\tau, d_\tau), \tau = 0, 1, \cdots\}$.*

Algorithm 8: Modified Barrier Learner Algorithm with Prior Information on $\mathcal{S}_l, l = 1, 2, \cdots, L, L+1$

Data: Constrained Markov Decision Process $\mathcal{M}$, $\mathcal{S}_l, l = 1, 2, \cdots, L, L+1$ defined for $\mathcal{M}$
Initialize $\hat{B}^{(0)}(s,a) = 0, \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
Initialize $l = 1$
for $\tau = 0, 1, \cdots$ **do**
  Draw $(s_\tau, a_\tau) \sim \mathrm{Unif}(\mathcal{S} \times \mathcal{A})$
  Sample transition $(s_\tau, a_\tau, s'_\tau, d_\tau)$ according to $P(S_1 = s'_\tau, D_1 = d_\tau | S_0 = s_\tau, A_0 = a_\tau)$
  if $\hat{B}^{(\tau)}(s_\tau, a_\tau) \neq -\infty$ *and* $s_\tau \in \mathcal{S}_l$ **then**
    $\hat{B}^{(\tau+1)} \leftarrow$ `barrier_update`$(\hat{B}^{(\tau)}, s_\tau, a_\tau, s'_\tau, d_\tau)$
  else
    $\hat{B}^{(\tau+1)} \leftarrow \hat{B}^{(\tau)}$
  end
  if $\hat{B}^{\tau+1}(s,a) = -\infty, \forall s \in \mathcal{S}_l, a \in \mathcal{A}$ **then**
    $l \leftarrow l + 1$
  end
end

Proof. Condition on a fixed sample trajectory $\mathcal{T} := \{(s_\tau, a_\tau, s'_\tau, d_\tau)\}_{\tau=0}^{\infty}$, the functions $B^{(\tau)}$ and $\hat{B}^{(\tau)}$ are deterministic. We have

$$B^{(\tau)}(s,a)|\mathcal{T} \leq \hat{B}^{(\tau)}(s,a)|\mathcal{T}, \forall \tau \geq 0, \forall (s,a) \in \mathcal{S} \times \mathcal{A}\,, \tag{46}$$

proved by induction: we have

$$B^{(0)}(s,a)|\mathcal{T} \leq \hat{B}^{(0)}(s,a)|\mathcal{T}, \forall (s,a) \in \mathcal{S} \times \mathcal{A}\,,$$

at initialization. Suppose that (46) holds at time $\tau = t$. If (s_t, a_t, s'_t, d_t) is accepted by both algorithms, or rejected by both algorithms, we have

$$B^{(t+1)}(s,a)|\mathcal{T} \leq \hat{B}^{(t+1)}(s,a)|\mathcal{T}, \forall (s,a) \in \mathcal{S} \times \mathcal{A}\,. \tag{47}$$

If (s_t, a_t, s'_t, d_t) is rejected by Algorithm 7 and accepted by Algorithm 8, then we have $B^{(t)}(s_t, a_t) = -\infty, \hat{B}^{(t)}(s_t, a_t) = 0$. (47) still holds, since only $\hat{B}^{(t+1)}(s_t, a_t)$ is updated to either 0 or $-\infty$. If (s_t, a_t, s'_t, d_t) is accepted by Algorithm 7 and rejected by Algorithm 8, then we have

$$B^{(t)}(s_t, a_t) = \hat{B}^{(t)}(s_t, a_t) = 0\,.$$

Inequality (47) still holds, since only $B^{(t+1)}(s_t, a_t)$ is updated to either 0 or $-\infty$. Now from (46), we immediately know that condition on the fixed sample trajectory $\mathcal{T}$,

$$\mathbb{1}\{B(s_\tau, a_\tau) \neq -\infty\} \leq \mathbb{1}\{\hat{B}(s_\tau, a_\tau) \neq -\infty\}, \forall \tau = 0, 1, \cdots$$

Notice that T_r ($\hat{T}_r$) is the minimum t such that $B^{(t)}$ ($\hat{B}^{(t)}$) becomes exactly the same as B^*. Then $T_r|\mathcal{T} \leq \hat{T}_r|\mathcal{T}$. Therefore one have, by law of total expectation,

$$\mathbb{E}\left[\sum_{\tau=0}^{T_r} \mathbb{1}\{B(s_\tau, a_\tau) \neq -\infty\}\right]$$
$$\leq \mathbb{E}\left[\mathbb{E}\left[\sum_{\tau=0}^{\hat{T}_r} \mathbb{1}\{\hat{B}(s_\tau, a_\tau) \neq -\infty\} \middle| \mathcal{T}\right]\right]$$

$$= \mathbb{E}\left[\sum_{\tau=0}^{\hat{T}_r} \mathbb{1}\{\hat{B}(s_\tau, a_\tau) \neq -\infty\}\right]$$

□

3) Expected detection time of modified algorithm: Lastly, we prove the following Theorem regarding the expected detection time of the modified algorithm.

Theorem 9. *Given an MDP with $\mathcal{S}_l, l = 1, 2, \cdots, L, L+1$ defined as in* (44)*. Assume that exists $\rho > 0$ such that the transition probability $P(S_1 = s' | S_0 = s, A_0 = a)$, is either zero or lower bounded by ρ, for all $s, s' \in \mathcal{S}, a \in \mathcal{A}$. Let $\hat{T}_r$ be earliest time when Algorithm 8 detects all unsafe state-action pairs as defined in* (45)*, then we have*

$$\mathbb{E}\left[\sum_{\tau=1}^{\hat{T}_r} \mathbb{1}\{\hat{B}^{(\tau)}(s_\tau, a_\tau) \neq -\infty\}\right] \leq \frac{|\mathcal{S}||\mathcal{A}|}{\rho} \sum_{l=1}^{L+1} \left(\sum_{k=1}^{|\mathcal{S}_l||\mathcal{A}|} \frac{1}{k}\right).$$

Proof. Let $\hat{T}_l, l = 1, \cdots, L+1$ denote the earliest time when all unsafe state-action pairs associated with $\mathcal{S}_l$ are detected by Algorithm 8, and we let $\hat{T}_0 = 0$. Then clearly

$$\hat{T}_{l-1} < \hat{T}_l, l = 1, \cdots, L+1, \quad \hat{T}_{L+1} = \hat{T}_r,$$

and $\Delta_l := \sum_{t=\hat{T}_{l-1}}^{\hat{T}_l - 1} \mathbb{1}\{\hat{B}^{(t)}(s_t, a_t) \neq -\infty\}$ is the number of accepted samples by Algorithm 8 between $\hat{T}_{l-1}$ and $\hat{T}_l$.

Notice that we can view the barrier learning process between $\hat{T}_{l-1}$ and $\hat{T}_l$ as detecting unsafe machines in the safe multi-arm bandits problem discussed in Section II: At time $\hat{T}_{l-1}$, there are in total $K_l := \left|\left\{(s,a) : s \in \bigcup_{k=l}^{L+1} \mathcal{S}_k, a \in \mathcal{A}\right\}\right|$ machines, and the number of unsafe machines is

$$M_l := |\{(s,a) : s \in \mathcal{S}_l, a \in \mathcal{A}, B^*(s,a) = -\infty\}|.$$

We have $K_l \leq |\mathcal{S}||\mathcal{A}|$, $M_l \leq |\mathcal{S}_l||\mathcal{A}|$. Furthermore, condition on such an unsafe machine is pulled, i.e. an unsafe (s,a) in $\mathcal{S}_l$ is accepted by Algorithm 8, the probability of declaring it unsafe is at least ρ. Because the (s,a) pair either transitions to some $s \in \bigcup_{k=1}^{l-1} \mathcal{S}_k$ that has been declared unsafe or directly incurs damage with non-zero probability, and that probability is lower bounded by ρ according to our assumption.

The acceptance of sample $(s_\tau, a_\tau, s'_\tau, d_\tau)$ is equivalent to pulling a uniformly randomly drawn arm out of arms that have not been declared unsafe. Theorem 2 suggests that the expected number of such "pulling" is upper bounded as

$$\mathbb{E}[\Delta_l] \leq \frac{K_l}{\rho}\left(\sum_{k=1}^{M_l} \frac{1}{k}\right) \leq \frac{|\mathcal{S}||\mathcal{A}|}{\rho}\left(\sum_{k=1}^{|\mathcal{S}_l||\mathcal{A}|} \frac{1}{k}\right).$$

Finally, we have

$$\mathbb{E}\left[\sum_{t=0}^{\hat{T}_r} \mathbb{1}\{\hat{B}^{(t)}(s_t, a_t) \neq -\infty\}\right]$$
$$= \mathbb{E}\left[\sum_{l=1}^{L+1} \Delta_l\right] \leq \frac{|\mathcal{S}||\mathcal{A}|}{\rho} \sum_{l=1}^{L+1}\left(\sum_{k=1}^{|\mathcal{S}_l||\mathcal{A}|} \frac{1}{k}\right).$$

□

Proof of Theorem 7. Given any MDP, we have $|\mathcal{S}_l| \leq |\mathcal{S}|, \forall l = 0, 1, \cdots, L+1$, then we have

$$\mathbb{E}[T] \leq \mathbb{E}\left[\sum_{\tau=1}^{\hat{T}_r} \mathbb{1}\{\hat{B}^{(\tau)}(s_\tau, a_\tau) \neq -\infty\}\right]$$
$$\leq \frac{|\mathcal{S}||\mathcal{A}|}{\rho} \sum_{l=1}^{L+1}\left(\sum_{k=1}^{|\mathcal{S}_l||\mathcal{A}|} \frac{1}{k}\right) \leq \frac{|\mathcal{S}||\mathcal{A}|(L+1)}{\mu}\left(\sum_{k=1}^{|\mathcal{S}||\mathcal{A}|} \frac{1}{k}\right),$$

where the first equality is from (43) and Claim 1. The result follows by upper bounding the summation with the log inequality $\sum_{k=1}^{n} \frac{1}{k} \leq \log(n+1)$. □

REFERENCES

[1] A. Castellano, J. Bazerque, and E. Mallada, "Learning to be safe, in finite time," in *2021 American Control Conference (ACC)*, IEEE, 2021.

[2] A. Castellano, H. Min, E. Mallada, and J. A. Bazerque, "Reinforcement learning with almost sure constraints," in *Learning for Dynamics and Control Conference*, pp. 559–570, PMLR, 2022.

[3] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[6] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, 2011.

[7] K. Zhou and J. C. Doyle, *Essentials of Robust Control*, vol. 104. Prentice hall Upper Saddle River, NJ, 1998.

[8] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe Model-based Reinforcement Learning with Stability Guarantees," *arXiv*, 2017.

[9] T. J. Perkins and A. G. Barto, "Lyapunov design for safe reinforcement learning," *J. of Mach. Learn. Res.*, vol. 3, no. Dec, pp. 803—832, 2002.

[10] F. Berkenkamp and A. P. Schoellig, "Safe and robust learning control with Gaussian processes," in *2015 European Control Conference (ECC)*, pp. 2496—2501, 2015.

[11] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[12] J. Garcia and F. Fernandez, "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437—1480, 2015.

[13] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe Exploration in Continuous Action Spaces," *arXiv preprint arXiv:1801.08757*, 2018.

[14] A. Wachi, Y. Sui, Y. Yue, and M. Ono, "Safe Exploration and Optimization of Constrained MDPs Using Gaussian Processes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, pp. 6548–6555, 2018.

[15] T. Xu, Y. Liang, and G. Lan, "Crpo: A new approach for safe reinforcement learning with convergence guarantee," in *International Conference on Machine Learning*, pp. 11480–11491, PMLR, 2021.

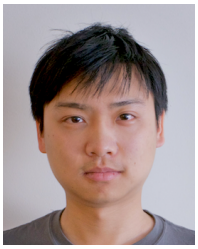
[16] S. Paternain, M. Calvo-Fullana, L. F. O. Chamon, and A. Ribeiro, "Safe policies for reinforcement learning via primal-dual methods," *IEEE Transactions on Automatic Control*, pp. 1–1, 2022.

[17] D. Ding, X. Wei, Z. Yang, Z. Wang, and M. R. Jovanović, "Provably Efficient Safe Exploration via Primal-Dual Policy Optimization," in *International Conference on Artificial Intelligence and Statistics*, pp. 3304–3312, PMLR, 2020.

[18] A. Wachi and Y. Sui, "Safe reinforcement learning in constrained markov decision processes," in *International Conference on Machine Learning*, pp. 9797–9806, PMLR, 2020.

[19] A. HasanzadeZonuzy, A. Bura, D. Kalathil, and S. Shakkottai, "Learning with safety constraints: Sample complexity of reinforcement learning for constrained MDPs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 7667–7674, 2021.
[20] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.
[21] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[22] S. Amani, M. Alizadeh, and C. Thrampoulidis, "Linear stochastic bandits under safety constraints," *Advances in neural information processing Ssystems*, vol. 32, 2019.
[23] A. Moradipari, C. Thrampoulidis, and M. Alizadeh, "Stage-wise conservative linear bandits," *Advances in neural information processing systems*, vol. 33, pp. 11191–11201, 2020.
[24] S. Amani, M. Alizadeh, and C. Thrampoulidis, "Regret bound for safe gaussian process bandit optimization," in *Learning for Dynamics and Control*, pp. 158–159, PMLR, 2020.
[25] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *J. of Machine Learning Res.*, vol. 3, no. Nov, pp. 397–422, 2002.
[26] A. Pacchiano, M. Ghavamzadeh, P. Bartlett, and H. Jiang, "Stochastic bandits with linear constraints," in *International Conference on Artificial Intelligence and Statistics*, pp. 2827–2835, PMLR, 2021.
[27] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, IEEE, 2019.
[28] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*, pp. 708–717, PMLR, 2020.
[29] D. M. Stipanović, I. Hwang, and C. J. Tomlin, "Computation of an over-approximation of the backward reachable set using subsystem level set functions," in *European Control Conference (ECC)*, pp. 300–305, 2003.
[30] A. Chakrabarty, A. Raghunathan, S. Di Cairano, and C. Danielson, "Data-driven estimation of backward reachable and invariant sets for unmodeled systems via active learning," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 372–377, 2018.
[31] E. Altman, *Constrained Markov Decision Process*, vol. 7. CRC Press, 1998.
[32] L. Prashanth, "Policy gradients for CVaR-constrained MDPs," in *Algorithmic Learning Theory: 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014. Proceedings 25*, pp. 155–169, Springer, 2014.
[33] E. Altman, "Constrained markov decision processes with total cost criteria: Lagrangian approach and dual linear program," *Mathematical methods of operations research*, vol. 48, no. 3, pp. 387–417, 1998.
[34] D. Ding, K. Zhang, T. Basar, and M. Jovanovic, "Natural policy gradient primal-dual method for constrained markov decision processes," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
[35] Y. Chen and M. Wang, "Stochastic primal-dual methods and sample complexity of reinforcement learning," *arXiv preprint arXiv:1612.02516*, 2016.
[36] H. Wei, X. Liu, and L. Ying, "Triple-q: A model-free algorithm for constrained reinforcement learning with sublinear regret and zero constraint violation," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics* (G. Camps-Valls, F. J. R. Ruiz, and I. Valera, eds.), vol. 151 of *Proceedings of Machine Learning Research*, pp. 3274–3307, PMLR, 28–30 Mar 2022.
[37] M. Hasanbeig, A. Abate, and D. Kroening, "Logically-constrained reinforcement learning," *arXiv preprint arXiv:1801.08099*, 2018.
[38] M. Hasanbeig, A. Abate, and D. Kroening, "Certified reinforcement learning with logic guidance," *arXiv preprint arXiv:1902.00778*, 2019.
[39] M. Hasanbeig, Y. Kantaros, A. Abate, D. Kroening, G. J. Pappas, and I. Lee, "Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5338–5343, IEEE, 2019.
[40] A. Lavaei, F. Somenzi, S. Soudjani, A. Trivedi, and M. Zamani, "Formal controller synthesis for continuous-space mdps via model-free reinforcement learning," in *ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 98–107, 2020.
[41] A. Wald, "Sequential Tests of Statistical Hypotheses," *The Annals of Mathematical Statistics*, vol. 2, no. 16, pp. 117—186, 1945.
[42] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.
[43] S. Janson, "Tail bounds for sums of geometric and exponential variables," *Statistics & Probability Letters*, vol. 135, pp. 1–6, 2018.
[44] A. Castellano, H. Min, J. Bazerque, and E. Mallada, "Learning to act safely with limited exposure and almost sure certainty," 2021.
[45] G. B. Folland, *Real analysis: modern techniques and their applications*, vol. 40. John Wiley & Sons, 1999.
[46] D. P. Bertsekas *et al.*, *Dynamic programming and optimal control: Vol. II*. Athena scientific, Belmont, 2012.
[47] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400 – 407, 1951.
[48] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen, "Breaking the sample size barrier in model-based reinforcement learning with a generative model," *Adv. in neural info. proc. systems*, vol. 33, pp. 12861–12872, 2020.
[49] J. N. Tsitsiklis, "Asynchronous stochastic approximation and q-learning," *Machine learning*, vol. 16, no. 3, pp. 185–202, 1994.
[50] A. Wald, "On Cumulative Sums of Random Variables," *The Annals of Mathematical Statistics*, vol. 15, no. 3, pp. 283 – 296, 1944.

Agustin Castellano is pursuing a Ph.D. at the Department of Electrical Engineering at Johns Hopkins University. He received the M.Sc. and his degree in Electrical Engineering from Universidad de la República, Uruguay in 2021 and 2017 respectively. For his M.Sc. dissertation he was awarded the first prize given by the National Academy of Engineers. His current research interests include Reinforcement Learning theory and algorithms, with applications to power system optimization.



Hancheng Min is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, Johns Hopkins University. He received the B.Eng. degree in Electrical Engineering and Automation from Tongji University in 2016, and the M.S. degree in Systems Engineering from University of Pennsylvania in 2018. His research interests include analysis and control of large-scale networks, reinforcement learning and deep learning theory.

Juan Andrés Bazerque (S'06-M'13) received the B.Sc. degree in electrical engineering from Universidad de la Republica (UdelaR), Montevideo, Uruguay, in 2003, ´ and the M.Sc. and Ph.D. degrees in Electrical Engineering from de University of Minnesota (UofM), Minneapolis, in 2010 and 2013 respectively. After his PhD studies he entered the Department of Electrical Engineering at UdelaR as an Assistant Professor. In 2022 he moved back to the US where he joined the Department of Electrical and Computer Engineering at the University of Pittsburgh. His current research interests include stochastic optimization and networked systems, focusing on reinforcement learning, graph signal processing, and power systems optimization and control. Dr. Bazerque is the recipient of the UofM's Master Thesis Award 2009-2010, and co-recipient of the best paper award at the 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communication 2007.

Enrique Mallada (S'09-M'13-SM') is an Associate Professor of Electrical and Computer Engineering at Johns Hopkins University since 2022. Prior to joining Hopkins in 2016, he was a Post-Doctoral Fellow in the Center for the Mathematics of Information at Caltech from 2014 to 2016. He received his Ingeniero en Telecomunicaciones degree from Universidad ORT, Uruguay, in 2005 and his Ph.D. degree in Electrical and Computer Engineering with a minor in Applied Mathematics from Cornell University in 2014. Dr. Mallada was awarded the Johns Hopkins Alumni Association Teaching Award in 2021, the Catalyst and Discovery Awards in 2020 and 2021, respectively, from Johns Hopkins University, the NSF CAREER award in 2018, the ECE Director's Ph.D. Thesis Research Award for his dissertation in 2014, the Center for the Mathematics of Information (CMI) Fellowship from Caltech in 2014, and the Cornell University Jacobs Fellowship in 2011. His research interests lie in the areas of control, dynamical systems, and optimization, with applications to engineering networks.