Policy Gradient for Continuing Tasks in Non-stationary Markov Decision Processes

Santiago Paternain[†], Juan Andrés Bazerque[*] and Alejandro Ribeiro[‡]

***Abstract*—Reinforcement learning considers the problem of finding policies that maximize an expected cumulative reward in a Markov decision process with unknown transition probabilities. In this paper we consider the problem of finding optimal policies assuming that they belong to a reproducing kernel Hilbert space (RKHS). To that end we compute unbiased stochastic gradients of the value function which we use as ascent directions to update the policy. A major drawback of policy gradient-type algorithms is that they are limited to episodic tasks unless stationarity assumptions are imposed. Hence preventing these algorithms to be fully implemented online, which is a desirable property for systems that need to adapt to new tasks and/or environments in deployment. The main requirement for a policy gradient algorithm to work is that the estimate of the gradient at any point in time is an ascent direction for the initial value function. In this work we establish that indeed this is the case which enables to show the convergence of the online algorithm to the critical points of the initial value function. A numerical example shows the ability of our online algorithm to learn to solve a navigation and surveillance problem, in which an agent must loop between to goal locations. This example corroborates our theoretical findings about the ascent directions of subsequent stochastic gradients. It also shows how the agent running our online algorithm succeeds in learning to navigate, following a continuing cyclic trajectory that does not comply with the standard stationarity assumptions in the literature for non episodic training.**

I. Introduction

Reinforcement learning (RL) problems–which is the interest in this paper–are a special setting for the analysis of Markov decision processes (MDPs) in which both the transition probabilities are unknown. The agent interacts with the environment and observes samples of a reward random variable associated to a given state and action pair [1]. These rewards samples are used to update the policy of the agent so as to maximize the Q- function, defined as the expected cumulative reward conditioned to the current state and action. The solutions to RL problems are divided in two main approaches. On one hand, those approaches that aim to learn the Q-function to then choose the action that for the current state maximizes said function. Among these algorithms the standard solution is Q-learning [2], whose earlier formulations were applicable in scenarios where the state and the action are discrete. The aforementioned algorithms suffer from the *curse of dimensionality*, with complexity growing exponentially with the number of actions and states [3]. This is of particular concern in problems where the state and the actions are continuous spaces, and thus, any reasonable discretization leads to a large number of states and possible actions. A common approach to overcome this difficulty is to assume that the Q-function admits a finite parameterization that can be linear [4], rely on a nonlinear basis expansion [5], or be given by a neural network [6]. Alternatively one can assume that the Q-function [7], [8] belongs to a reproducing kernel Hilbert space. However, in these cases, maximizing the Q-function to select the best possible action is computationally challenging. Moreover, when using function approximations Q-learning may diverge [9].

This motivates the development of another class of algorithms that attempts to learn the optimal policy by running stochastic gradient ascent on the Q-function with respect to the policy parameters [10]–[12] or with respect to the policy itself in the case of non-parametric representations [13], [14]. These gradients involve the computation of expectations which requires knowledge of the underlying probabilistic model. With the goal of learning from data only, they provide unbiased estimates of the gradients which are used for stochastic approximation [15]. One of the classic examples of estimates of the gradients used in discrete state-action spaces is REINFORCE [10]. Similar estimates can also be computed in the case of parametric [16] and non-parametric function approximations [14]. Once these unbiased estimates have been computed convergence to the critical points can be established under a diminishing step-size as in the case in parametric optimization [17]. A drawback of said estimators is that they have high variance and therefore they suffer from slow convergence. This issue can be mitigated using Actor-Critic methods [18]–[20] to estimate the policy gradients. To compute these estimates however, one is required to re-initialize the system for every new iteration. Hence, limiting its application to episodic tasks [10], [14].

A common workaround to this hurdle is to modify the value function so to consider the average rate of reward instead of the cumulative reward [1, Chapter 13]. This formulation also requires that under every policy the MDP converges to a steady state distribution that is independent of the initial state. The convergence to a stationary distribution is restrictive in many cases, as we discus in Section II, since it prevents the agents from considering policies that result in cyclic behaviors for instance. Not being able to reproduce these behaviors is a drawback for problems like surveillance where the policy that the agent should follow is one that visits different points of interest. Moreover, even for problems where the target is a specific state, and thus the convergence to the stationary distribution is a reasonable assumption, the average reward formulation may modify the optimal policies in the sense that it is a formulation that ignores transient behaviors. We discuss

Work supported by ARL DCIST CRA W911NF-17-2-0181. [†] Dept. of Electrical and System Engineering, Univ. of Pennsylvania. Email: {spater,aribeiro} @seas.upenn.edu. [*]Univ. de la República. Email:jbazerque@fing.edu.uy

this issue in more detail also in Section II.

Given the offline policy gradient algorithm in [14], we aim to avoid the reinitialization requirement while keeping the cumulative reward as value function. In particular, we compute stochastic gradients as in [14] which are guaranteed to be unbiased estimates of the gradient of the value function at the state that systems finds itself at the beginning of the iteration. Because this estimation requires rollouts at each iteration, the agent is in fact computing estimates of the gradients of value functions at different states.

Building on our preliminary results [21], we establish in Theorem 1 that the gradients of the value function at any state are also ascent directions of the value function at the initial state. Leveraging this result we address the convergence of the online policy gradient algorithm to a neighborhood of the critical points in Theorem 2, hence dropping the assumption of the convergence to—and existence of—the stationary distribution over states for every intermediate policy. These results are backed by Proposition 3, which establishes that a critical point of the value function conditioned at the initial state is also a critical point for the value functions conditioned at states in the future, suggesting that the landscape of different value functions is still very similar. Finally, as an accessory computational refinement, we add a compression step to the online algorithm to reduce the number of kernels by trading-off a discretionary convergence error. This refinement uses Orthogonal Kernel Matching Pursuit [22]. Other than concluding remarks the paper ends with numerical experiments where we consider an agent whose goal is to surveil a region of the space while having to visit often enough a charging station. The cyclic nature of this problem evidences the ability of our algorithm to operate in a non stationary setup and carry the task by training in a fully online fashion, without the need of episodic restarts. The experiment is also useful to corroborate our theoretical findings about the ascending direction properties of stochastic gradients computed at different points of the trajectory.

II. Problem Formulation

In this work we are interested in the problem of finding a policy that maximizes the expected discounted cumulative reward of an agent that chooses actions sequentially. Formally, let us denote the time by $t \in \{\{0\}, \mathbb{N}\}$ and let $\mathcal{S} \subset \mathbb{R}^n$ be a compact set denoting the state space of the agent, and $\mathcal{A} = \mathbb{R}^p$ be its action space. The transition dynamics are governed by a conditional probability $P^{a_t}_{s_t \to s_{t+1}}(s) := p(s_{t+1} = s \big| (s_t, a_t) \in \mathcal{S} \times \mathcal{A})$ satisfying the Markov property, i.e., $p(s_{t+1} = s \big| (s_u, a_u) \in \mathcal{S} \times \mathcal{A}, \forall u \leq t) = p(s_{t+1} = s \big| (s_t, a_t) \in \mathcal{S} \times \mathcal{A})$. The policy of the agent is a multivariate Gaussian distribution with mean $h : \mathcal{S} \to \mathcal{A}$. The later map is assumed to be a vector-valued function in a vector-valued RKHS $\mathcal{H}$. We formally define this notion next, with comments ensuing.

Definition 1. *A vector valued RKHS $\mathcal{H}$ is a Hilbert space of functions $h : \mathcal{S} \to \mathbb{R}^p$ such that for all $\mathbf{c} \in \mathbb{R}^p$ and $s \in \mathcal{S}$, the following reproducing property holds*

$$< h(\cdot), \kappa(s, \cdot)\mathbf{c} >_{\mathcal{H}} = h(s)^\top \mathbf{c}. \tag{1}$$

where $\kappa(s, s')$ is a symmetric matrix-valued function that renders a positive definite matrix when evaluated at any $s, s' \in \mathcal{S}$.

If $\kappa(s, s')$ is a diagonal matrix-valued function with diagonal elements $\kappa(s, s')_{ii}$, and $\mathbf{c}$ is the i-th canonical vector in $\mathbb{R}^p$, then (1) reduces to the standard one-dimensional reproducing property per coordinate $h_i(s) =< h_i(\cdot), \kappa(s, \cdot)_{ii} >$. With the above definitions the policy of the agent is the following conditional probability of the action $\pi_h(a|s) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$, with

$$\pi_h(a|s) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{(a-h(s))^\top \Sigma^{-1}(a-h(s))}{2}}. \tag{2}$$

The latter means that given a function $h \in \mathcal{H}$ and the current state $s \in \mathcal{S}$, the agent selects an action $a \in \mathcal{A}$ from a multivariate normal distribution $\mathcal{N}(h(s), \Sigma)$. The choice of a random policy over a deterministic policy $a = h(s)$ makes the problem tractable both theoretically and numerically as it is explained in [14]. The actions selected by the agent result in a reward defined by a function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

The objective is then to find a policy $h^\star \in \mathcal{H}$ such that it maximizes the expected discounted reward

$$h^\star := \operatorname*{argmax}_{h \in \mathcal{H}} U_{s_0}(h) = \operatorname*{argmax}_{h \in \mathcal{H}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| h, s_0\right], \tag{3}$$

where the expectation is taken with respect to all states except s_0, i.e., $s_1, \ldots$ and all actions $a_0, a_1, \ldots,$. The parameter $\gamma \in (0, 1)$ is a discount factor that gives relative weights to the reward at different times. Values of γ close to one imply that current rewards are as important as future rewards, whereas smaller values of γ give origin to myopic policies that prioritize maximizing immediate rewards. It is also noticeable that $U_{s_0}(h)$ is indeed a function of the policy h, since policies affect the trajectories $\{s_t, a_t\}_{t=0}^{\infty}$.

As discussed in Section I problem (3) can be tackled using methods of the policy gradient type [1, Chapter 13]. These methods have been extended as well to non-parametric scenarios as we consider here [13], [14]. A drawback of these methods is that they require restarts which prevents them from a fully online implementation. To better explain this claim let us write down the expression of the gradient of the objective in (3) with respect to h. Before doing so, we are required to define the discounted long-run probability distribution $\rho_{s_0}(s, a)$

$$\rho_{s_0}(s, a) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p(s_t = s, a_t = a | s_0), \tag{4}$$

where $p(s_t = s, a_t = a | s_0)$ is the distribution of the MDP under a policy h

$$p(s_t = s, a_t = a | s_0) = \\ \pi_h(a_t|s_t) \int \prod_{u=0}^{t-1} p(s_{u+1}|s_u, a_u) \pi_h(a_u|s_u)\, d\mathbf{s}_{t-1} d\mathbf{a}_{t-1}, \tag{5}$$

with $d\mathbf{s}_{t-1} = (ds_1, \ldots ds_{t-1})$ and $d\mathbf{a}_{t-1} = (da_0, \ldots da_{t-1})$. We also require to define $Q(s, a; h)$, the expected discounted reward for a policy h that at state s selects action a. Formally this is

$$Q(s,a;h) := \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\Big|h, s_0=s, a_0=a\right]. \quad (6)$$

With these functions defined, the gradient of the discounted rewards with respect to h yields [11], [13]

$$\nabla_h U_{s_0}(h,\cdot) = \quad (7)$$
$$\frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\rho_{s_0}(s,a)}\left[Q(s,a;h)\kappa(s,\cdot)\Sigma^{-1}\left(a-h(s)\right)\Big|h\right],$$

where the gradient of $U_{s_0}(h)$ with respect to the continuous function h is defined in the sense of Frechet, rendering a function in $\mathcal{H}$. This is represented by the notation in $\nabla_h U_{s_0}(h,\cdot)$, where the dot substitutes the second variable of the kernel, belonging to $\mathcal{S}$, which is omitted to simplify notation. Observe that the expectation with respect to the distribution $\rho_{s_0}(s,a)$ is an integral of an infinite sum over a continuous space. Although this could have a tractable solution in some specific cases, this would require the system transition density $p(s_{t+1}|s_t,a_t)$ which is unknown in the context of RL. Thus, computing (7) in closed form becomes impractical. In fact, a large number of samples might be needed to obtain an accurate Monte Carlo approximation even if $p(s_{t+1}|s_t,a_t)$ were known. In [14] an offline stochastic gradient ascent algorithm is proposed to overcome these difficulties and it is shown to converge to a critical point of the functional U_{s_0}. Notice that to compute stochastic approximations of the gradient (7) on is required to sample from the distribution ρ_{s_0} which depends on the initial condition. This dependency results in a fundamental limitation for online implementation. We present in Section III-A the algorithm and a summary of the main results in [14] since it serves as the basis for a fully online algorithm and to understanding the aforementioned difficulties associated to the online problem in detail. Before doing so we discuss a common workout to the continuing task problem—or the problem of avoiding restarts.

A. Reinforcement learning in continuous tasks

When considering continuing tasks it is customary to modify the objective (3) and instead attempt to maximize the undiscounted objective [23, Chapter 13]

$$U'_{s_0}(h) = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[r(s_t,a_t)\Big|s_0\right]. \quad (8)$$

Consider a steady state distribution $\rho'(s)$ that is ergodic and independent of the starting point, this is, a distribution that satisfies

$$\rho'(s') = \int \rho'(s)\pi_h(a|s)p(s'|s,a)\,dsda, \quad (9)$$

for all $s'\in\mathcal{S}$. Under the assumptions that such distribution exists and that the distribution of the MDP converges to it, the limit in (8) is finite. Then, using Stolz-Cesaro's Lemma (see e.g. [24, pp. 85-88]), then (8) reduces to

$$U'_{s_0}(h) = \lim_{t\to\infty}\mathbb{E}\left[r(s_t,a_t)\Big|s_0\right], \quad (10)$$

where the expectation is with respect to the stationary distribution (9). Thus, we can rewrite the previous expression as

$$U'_{s_0}(h) = \mathbb{E}_{s\sim\rho',a\sim\pi_h}\left[r(s,a)\right]. \quad (11)$$

The advantage of this formulation is that the objective function is now independent of the initial state and therefore estimates of the gradient can be computed without requiring the reinitialization of the trajectory.

The convergence to a stationary distribution however, prevent us from achieving cyclical behaviors, for the most part. In particular, a sufficient condition for the convergence is that the Markov chain is aperiodic [25, Theorem 6.6.4]. Which hints to the fact that in some situations cycles are not achievable under these conditions. Let us consider the following scenario as an example. An agent is required to visit three different locations denoted by states s_1, s_2 and s_3 and there is a charging station s_0. In this scenario is not surprising that the optimal policy is such that it cycles in the different locations and the charging station. Consider that the resulting Markov Chain is such that with probability one we transition from s_i to s_{i+1} for $i=0,\ldots,2$ and from s_3 to s_0. In this scenario there exists a stationary distribution that places equal mass in every state, i.e = $\rho(s_i)=1/4$ for all $i=0,\ldots 3$. However, the convergence to this distribution is only guaranteed if the initial distribution is the stationary one. This assumption may not be realistic for this scenario since the agent is most likely to start in the charging station than in the other locations for instance.

Even if a stationary distribution is not attainable for the cyclic example just described, the return (8) is still well defined. Thus, we could attempt to extend the theory for continuing tasks starting from (8) and avoiding (11), without relying on a stationary distribution. However, we argue that the discounted formulation in (3) may be the preferred choice when transient behaviors are deemed important. Consider for instance the following MDP where the states are defined as $\mathcal{S}=\{0,1,\ldots,10\}$ and the actions are $\mathcal{A}=\{-1,1\}$. The transition dynamics are such that for all $s\in\mathcal{S}\setminus\{0,10\}$ we have that $s_{t+1}=s_t+a_t$, in the case of of $s_t=0$ we have that $s_{t+1}=s_t+a_t\mathbb{1}(a_t>0)$ and in the case of $s_t=10$ we have $s_{t+1}=s_t$ regardless of the action selected. All the states yield zero rewards except for $s_t=10$ whose reward is 1. Notice that under any random policy, as long as $P(a_t=1)>0$, the state converges to 10 and thus the average return (8) takes the value one. This is the same value as that of choosing the action $a_t=1$ for any state. The discounted formulation (3) allows us to distinguish between these two policies since the larger the average time to reach the state $s=10$ the smaller the value function.

Having argue the practical importance of considering problems of the form (3) for continuing tasks we proceed to describe a policy gradient based solution.

III. Online Policy Gradient

A. Stochastic Gradient Ascent

In order to compute a stochastic approximation of $\nabla_h U_{s_0}(h)$ given in (7) we need to sample from the distribution $\rho_{s_0}(s,a)$ defined in (4). The intuition behind $\rho_{s_0}(s,a)$ is that it weights by $(1-\gamma)\gamma^t$ the probability of the system being at a specific state-action pair (s,a) at time t. Notice that the weight $(1-\gamma)/\gamma^t$ is equal to the probability of a geometric random variable of parameter γ to take the value t. Thus, one can interpret the distribution $\rho_{s_0}(s,a)$ as the probability of reaching the state-action pair (s,a) after running the system for T steps, with T randomly drawn from a geometric distribution of parameter γ, and starting at state s_0. The geometric sampling transforms the discounted infinite horizon problem into an undiscounted episodic problem with random horizon (see e.g. [26, pp.39-40]). This supports steps 2-7 in Algorithm 1 which describes how to obtain a sample $(s_T,a_T)\sim\rho_{s_0}(s,a)$. Then to compute an unbiased estimate of $\nabla_h U_{s_0}(h)$ (cf., Proposition 1) one can substitute the sample (s_T,a_T) in the stochastic gradient expression

$$\hat{\nabla}_h U_{s_0}(h,\cdot)=\frac{1}{1-\gamma}\hat{Q}(s_T,a_T;h)\kappa(s_T,\cdot)\Sigma^{-1}(a_T-h(s_T)), \tag{12}$$

with $\hat{Q}(s_T,a_T;h)$ being an unbiased estimate of $Q(s_T,a_T;h)$. Algorithm 1 summarizes the steps to compute the stochastic approximation in (12). We claim that it is unbiased in Proposition 1 as long as the rewards are bounded. We formalize this assumption next as long with some other technical conditions required along the paper.

Assumption 1. There exists $B_r>0$ such that $\forall(s,a)\in\mathcal{S}\times\mathcal{A}$, the reward function $r(s,a)$ satisfies $|r(s,a)|\leq B_r$. In addition $r(s,a)$ has bounded first and second derivatives, with bounds $|\partial r(s,a)/\partial s|\leq L_{rs}$ and $|\partial r(s,a)/\partial a|\leq L_{ra}$.

Notice that these assumptions are on the reward which is user defined, as such they do not impose a hard requirement on the problem.

Algorithm 1 StochasticGradient

```
Input: $h$, $s_0$
1: Draw an integer $T$ form a geometric distribution with parameter $\gamma$, $P(T=t)=(1-\gamma)\gamma^t$
2: Select action $a_0\sim\pi_h(a|s)$
3: for $t=0,1,\ldots T-1$ do
4:     Advance system $s_{t+1}\sim P^{a_t}_{s_t\to s_{t+1}}$
5:     Select action $a_{t+1}\sim\pi_h(a|s_{t+1})$
6: end for
7: Get estimate of $Q(s_T,a_T;h)$
8: Compute the stochastic gradient $\hat{\nabla}_h U(h,\cdot)$ as in (12)
   return $\hat{\nabla}_h U(h,\cdot)$
```

Proposition 1 ((Proposition 3 [14])). *The output $\hat{\nabla}_h U_{s_0}(h,\cdot)$ of Algorithm 1 is an unbiased estimate of $\nabla_h U_{s_0}(h,\cdot)$ in* (7).

An unbiased estimate of $Q(s_T,a_T)$ can be computed considering the cumulative reward from $t=T$ until a randomly distributed horizon $T_Q\sim geom(\gamma)$ (cf., Proposition 2 [14]). The variance of this estimate may be high resulting on a slow convergence of the policy gradient algorithm (Algorithm 1). For these reasons, the literature on RL includes several practical improvements. Variance can be reduced by including batch versions of the gradient method, in which several stochastic gradients are averaged before performing the update in (12). One particular case of a batch gradient iteration in [14], averages two gradients sharing the same state s_i with stochastic actions. Other approaches include the inclusion of baselines [10] and actor critic methods [18]–[20]. Irrespective of the form selected to estimate the Q function with the estimate (12) one could update the policy iteratively running stochastic gradient ascent

$$h_{k+1}=h_k+\eta_k\hat{\nabla}_h U_{s_0}(h_k,\cdot), \tag{13}$$

where $\eta_k>0$ is the step size of the algorithm. Under proper conditions stochastic gradient ascent methods can be shown to converge with probability one to the local maxima [27]. This approach has been widely used to solve parametric optimization problems where the decision variables are vectors in R^n and in [14] these results are extended to non-parametric problems in RKHSs. Observe however, that in order to provide an estimate of $\nabla U_{s_0}(h_k,\cdot)$, Algorithm 1 requires s_0 as the initial state. Hence, it is not possible to get estimates of the gradient without resetting the system to the initial state s_0, preventing a fully online implementation. As discussed in Section II-A, this is a common challenge in continuing task RL problems and in general the alternative is to modify the objective the function and to assume the existence of a steady-state distribution to which the MDP converges (see e.g., [1, Chapter 13] or [20]), to make the problem independent of the initial state. In this work we choose to keep the objective (3) since the ergodicity assumption is not necessarily guaranteed in practice and the alternative formulation makes transient behaviors irrelevant, as it was also discussed in Section II-A. Notice that, without loss of generality, Algorithm 1 can be initialized at state s_k and its output becomes an unbiased estimate of $\nabla U_{s_k}(h_k,\cdot)$. The main contribution of this work is to show that the gradient of $U_{s_k}(h)$ is also an ascent direction for $U_{s_0}(h)$ (cf., Theorem 1) and thus, these estimates can be used to maximize $U_{s_0}(h)$ hence allowing a fully online implementation. We describe the algorithm in the next section.

B. Online Implementation

As suggested in the previous section it is possible to compute unbiased estimates of $\nabla_h U_{s_k}(h_k)$ by running Algorithm 1 with inputs h_k and s_k. The state s_k is defined for all $k\geq 1$ as the state resulting from running the Algorithm 1 with inputs h_{k-1} and s_{k-1}. This is, at each step of the online algorithm —which we summarize under Algorithm 2—the system starts from state s_k and transits to a state s_{T_k} following steps 3–6 of Algorithm 1. Then, it advances from s_{T_k} to s_{k+1} to perform the estimation of the Q-function, one that admits an online implementation, for instance by adding the rewards of the next T_Q steps with T_Q being a geometric random variable. The state s_{k+1} is the initial state for the next iteration of Algorithm 2. Notice that the update (13) —step 5 in Algorithm 2 —requires the introduction of a new element $\kappa(s_{T_k},\cdot)$ in the kernel dictionary at each iteration, thus resulting in memory explosion. To overcome this limitation we modify the stochastic gradient ascent by introducing a projection over a RKHS of lower dimension as long as the induced error remains below a given compression budget. This algorithm,

Algorithm 2 Online Stochastic Policy Gradient Ascent

```
Input: step size $\eta_0$
1: Initialize: $h_0 = 0$, and draw initial state $s_0$
2: for $k = 0 \ldots$ do
3:     Compute the stochastic gradient and next state:
4: $\left(\hat{\nabla}_h U(h_k,\cdot), s_{k+1}\right) = \text{StochasticGradient}(h_k, s_k)$
5:     Stochastic gradient ascent step
           $\tilde{h}_{k+1} = h_k + \eta_k \hat{\nabla}_h U(h_k,\cdot)$
6:     Reduce model order $h_{k+1} = \text{KOMP}(\tilde{h}_{k+1}, \epsilon_K)$
7: end for
```

which runs once after each gradient iteration, prunes the kernel expansion that describes the policy h to remove the kernels that are redundant up to an admissible error level $\epsilon_k > 0$. The subroutine is known as Kernel Orthogonal Match and Pursuit (KOMP) [28] —step 6 in Algorithm 2.

The fundamental reason to do this pruning projection over a smaller subspace is that it allow us to control the model order of the policy h_k, as it is shown in Theorem 2. However, the induced error translates into a bias on the estimate of $\nabla_h U_{s_k}(h,\cdot)$. We formalize this claim in the next proposition.

Proposition 2. *The update of Algorithm 2 is equivalent to running biased stochastic gradient ascent*

$$h_{k+1} = h_k + \eta \hat{\nabla}_h U_{s_k}(h,\cdot) + b_k, \tag{14}$$

with bias bounded by the compression budget ϵ_K for all k, i.e., $\|b_k\|_{\mathcal{H}} < \epsilon_K$.

Proof. The proof is identical to that in [14, Proposition 5]. ■

As stated by the previous proposition the effect of introducing the KOMP algorithm is that of updating the policy by running gradient ascent, where now the estimate is biased. The later will prevent the algorithm to converge to a critical point of the value function. However, we will be able to establish convergence to a neighborhood of the critical point as long as the compression is such that the error introduced is not too large. A difference between the online algorithm and the offline one presented in [14] is that even for a compression error of $\epsilon_K = 0$ we cannot achieve exact convergence to the critical points, because the directions that are being used to ascend in the function $U_{s_0}(h)$ are in fact estimates of the gradients of $U_{s_k}(h)$. In the next section we discuss this in more detail and we establish that the inner product of gradients of $U_{s_k}(h)$ and $U_{s_0}(h)$ is positive when h belongs to a properly selected Gaussian RKHS (Theorem 1).

IV. All gradients are ascent directions

As we stated in the previous section, the main difference when comparing the online —continuing task —with the offline setting [14] —episodic task —is in the gradient of the value function that we estimate. In the online setting, we have access to estimates of the gradient of the value function conditioned on the state s_k, that is $\nabla U_{s_k}(h_k,\cdot)$, where s_k changes from one iteration to another. On the other hand, in the offline setting we can restart the system to its original state s_0 or redraw it from a given distribution $P(s_0)$, so that we can compute estimates of $\nabla U_{s_0}(h_k,\cdot)$ at each iteration. Thus, in the offline case we perform ascent steps over the same function U_{s_0}, whereas in the online setting would perform gradient steps over functions U_{s_k} which are different for each k. This main difference is as well a fundamental challenge since in principle we are not guaranteed that the gradients that can be computed are ascent directions of the function of interest $U_{s_0}(h)$. Moreover, a second question is whether finding the maximum of the value function conditioned at s_0 is a problem of interest or not after we reach a new state s_k. We answer the second question in Proposition 3 by showing that if h is a critical point of $U_{s_k}(h)$ it will also be a critical point of $U_{s_l}(h)$ for all $l \geq k$. The latter can be interpreted in the following way, having a policy that is optimal at a given time, makes it optimal for the future. An in that sense, maximizing the initial objective function is a valid problem, since finding a maximum for that function means that we had found one for all $U_{s_k}(h)$. To formalize this result, we analyze the critical points of $U_{s_k}(h)$. To do so write $\nabla_h U_{s_k}(h,\cdot)$ (cf., (7)) as the following integral

$$\nabla_h U_{s_k}(h,\cdot) = \frac{1}{1-\gamma}\int Q(s,a;h)k(s,\cdot)\Sigma^{-1}(a-h(s))\rho_{s_k}(s,a)\,dsda, \tag{15}$$

where $\rho_{s_k}(s,a)$ is the distribution defined in (4). We work next towards writing ρ_{s_k} as a product of a distribution of states and a distribution of actions. To that end, write the MPD transition distribution $p(s_t = s, a_t = a|s_k)$ for any $t \geq k$ as

$$\begin{aligned} p(s_t = s, a_t = a|s_k) &= p(s_t = s|s_k)\pi_h(a_t = a|s_t, s_k) \\ &= p(s_t = s|s_k)\pi_h(a_t = a|s_t), \end{aligned} \tag{16}$$

where the last equality follows from the fact that the action depends only on the current state conditional on the policy (cf.,(2)). By substituting the previous expression in (4), $\rho_{s_k}(s,a)$ reduces to

$$\rho_{s_k}(s,a) = (1-\gamma)\sum_{t=k}^{\infty}\gamma^t \pi_h(a_t = a|s_t = s)p(s_t = s|s_k). \tag{17}$$

Notice that the density $\pi_h(a_t = a|s_t = s)$ is independent of t and thus, the previous expression yields

$$\rho_{s_k}(s,a) = \pi_h(a|s)(1-\gamma)\sum_{t=k}^{\infty}\gamma^t p(s_t = s|s_k). \tag{18}$$

Hence, defining $\rho_{s_k}(s) := (1-\gamma)\sum_{t=k}^{\infty}\gamma^t p(s_t = s|s_k)$, it follows that $\rho_{s_k}(s,a) = \rho_{s_k}(s)\pi_h(a|s)$. Having $\rho_{s_k}(s,a)$ written as a product of a function depending on the states only and a function depending on the action only, allows us to reduce the expression in (15) to

$$\nabla_h U_{s_k}(h,\cdot) = \frac{1}{1-\gamma}\int D(s)\rho_{s_k}(s)\kappa(s,\cdot)\,ds, \tag{19}$$

where the function $D(s)$ is the result of the integration of all the terms that depend on the action

$$D(s) = \int Q(s,a;h)\Sigma^{-1}\left(a - h(s)\right)\pi_h(a|s)\,da. \tag{20}$$

Writing the gradient as in (19), allows us to split the integrands in the product of a term $\rho_{s_k}(s)$ that depends on the state at time k and a term $D(s)\kappa(s,\cdot)$ that do not depend on s_k. Hence, if a policy h is such that $D(s)$ is zero for all s, then h is a critical point for all value functions. This idea suggests that the quantity $D(s)$ is of fundamental importance in the problem. Indeed, $D(s)$ is an approximation of the derivative of the Q-function with respect to a. To see why this is the case, observe that because $\pi_h(a|s)$ is Gaussian, then $\Sigma^{-1}(a-h(s))\pi_h(a|s)$ is the derivative of $\pi_h(a|s)$ with respect to a. Hence, $D(s)$ can be written as

$$D(s) = \int Q(s,a;h)\frac{\partial \pi_h(a|s)}{\partial a}da. \tag{21}$$

Notice that as the covariance matrix of $\pi_h(a|s)$ approaches the null matrix, the distribution $\pi_h(a|s)$ approaches a Dirac delta centered at $h(s)$. That being the case, $D(s)$ yields

$$D(s) \simeq \int Q(s,a;h)\delta'(a-h(s))\,da = \frac{\partial Q(s,a;h)}{\partial a}\Big|_{a=h(s)}. \tag{22}$$

In this case, the fact that $D(s)$ is identically zero means that we have found a policy h that makes every action a stationary point of the Q-function. The previous observation relates to Bellman's optimality condition, that establishes that a policy is optimal if it is such that it selects the actions that maximize the Q-function. When Σ is different than the null matrix, $D(s)$ is an approximation of the derivative. In [29] the aforementioned Gaussian smoothing is used as an approximation of the stochastic gradient in the context of zero-order optimization, and is formally established that it approximates the derivative with an error that depends linearly on the norm of the covariance matrix. The insights provided in the previous paragraphs regarding the importance of the function $D(s)$ are not enough to fully characterize the critical points of $U_{s_k}(h)$ since according to (19) its gradient depends as well on the long run discounted distribution $\rho_{s_k}(s)$. The fact that this distribution might take the value zero at different states for different s_k does not allow us to say that a policy can be a critical point of every value function. However, we will be able to prove that if a policy is a critical point for $U_{s_k}(h)$ it is also a critical point for $U_{s_l}(h)$ for every $l \geq k$. To formalize the previous statement we require the following auxiliary result.

Lemma 1. *Let S_0 be the following set*

$$\mathcal{S}_0 = \left\{s \in \mathcal{S} : \exists\, t \geq 0, p(s_t = s|s_0) > 0\right\}. \tag{23}$$

For all $s', s \in \mathcal{S}_0$ and $s'' \in \mathcal{S} \setminus \mathcal{S}_0$ we have that

$$\rho_{s_0}(s) > 0 \quad \textit{and} \quad \rho_{s'}(s'') = 0. \tag{24}$$

Proof. See appendix A. ■

The set $\mathcal{S}_0$ contains the states for which the probability measure conditioned on the policy and on the initial state s_0 is strictly positive. We term S_0 the set of reachable states from s_0. The previous result, ensures that for all reachable states $s \in \mathcal{S}_0$, the probability measure $\rho_{s_0}(s)$ is strictly positive. The latter is not surprising, since intuitively, the distribution $\rho_{s_0}(s)$ is a weighted sum of the distributions of reaching the state s starting from s_0 at different times. Moreover, and along the same lines, we establish that if a point cannot be reached starting from s_0, it cannot be reached starting from any other point that is reachable from s_0. The previous result can be summarized by saying that set of reachable points does not increase as the system evolves, i.e., $\mathcal{S}_k \subseteq \mathcal{S}_0$ for all $k \geq 0$. Building on the previous result we show that the set of critical points of $U_{s_k}(h)$ can only increase with the iterations. This means that a critical point of the functional $U_{s_k}(h)$ is also a critical point of the functional conditioned at any state visited in the future. Without loss of generality we state the result for $k=0$ and with the dimension of the action space $p=1$.

Proposition 3. *If $h \in \mathcal{H}$ is a critical point of $U_{s_0}(h)$, then it is also a critical point for $U_{s_l}(h)$ for all $l \geq 0$, with $s_l \in \mathcal{S}_0$.*

Proof. Let us start by writing the square of norm of $\nabla_h U_{s_0}(\cdot)$ according to (19) as

$$\|\nabla_h U_{s_0}(h,\cdot)\|^2 = \int\int D(s)\rho_{s_0}(s)\kappa(s,s')D(s')\rho_{s_0}(s')\,dsds'. \tag{25}$$

From Mercer's Theorem (cf., [30]) there exists $\lambda_i > 0$ and orthornormal basis $e_i(s)$ of $L^2(\mathcal{S})$ such that

$$\kappa(s,s') = \sum_{i=1}^{\infty}\lambda_i e_i(s)e_i(s'). \tag{26}$$

Using the previous result, we can decompose the expression in (25) as the following sum of squares

$$\|\nabla_h U_{s_0}(h,\cdot)\|^2 = \sum_{i=1}^{\infty}\lambda_i\left[\int D(s)\rho_{s_0}(s)e_i(s)\,ds\right]^2. \tag{27}$$

Notice that the previous expression can take the value zero if and only if for all $i = 1.\ldots$ we have that

$$\int D(s)\rho_{s_0}(s)e_i(s)ds = 0. \tag{28}$$

Because $e_i(s)$ with $i = 1\ldots$ form an orthogonal basis of $L^2(\mathcal{S})$ it means that (28) holds if and only if $D(s)\rho_{s_0}(s) \equiv 0$. To complete the proof, we are left to show that if $D(s)\rho_{s_0}(s) \equiv 0$ then it holds that $D(s)\rho_{s_l}(s) \equiv 0$ for all $l \geq 0$. The latter can be established by showing that for any $s \in S$ such that $\rho_{s_0}(s) = 0$ we also have that $\rho_{s_l}(s) = 0$ which follows by virtue of Lemma 1. ■

The previous result formalizes the idea that if we find an optimal policy at given time, then it is optimal for all future states. Moreover, the latter is true for every critical point, which suggests, that the value functions conditioned at different initial states should be similar. We formalize this intuition in Theorem 1 where we show that $\nabla_h U_{s_k}(h)$ is an ascent direction for $U_{s_0}(h)$ if the distribution $\rho_{s_k}(s)$ is bounded above and bounded away from zero. We also require some smoothness assumptions on the transition probability which we formalize next.

Assumption 2. There exists $\beta_\rho > 0$ and B_ρ such that for all $s_k, s \in \mathcal{S}_0$ and for all $h \in \mathcal{H}$ we have that

$$B_\rho \geq \rho_{s_k}(s) \geq \beta_\rho. \tag{29}$$

In addition we have that the transition probability is Lipschitz with constant L_p, i.e.,

$$|p(s_t = s|s_{t-1}, a_{t-1}) - p(s_t = s'|s_{t-1}, a_{t-1})| \leq L_p \left\| s - s' \right\|. \tag{30}$$

We require as well the following smoothness properties of the probability transition

$$p'(s,a) := \frac{\partial p(s_{t+1}|s_t, a_t)}{\partial a_t}|_{s_t=s, a_t=a} \tag{31}$$

to be Lipschitz with constants L_{ps} and L_{pa}, this is

$$|p'(s,a) - p'(s',a')| \leq L_{ps} \left\| s - s' \right\| + L_{pa} \left\| a - a' \right\|. \tag{32}$$

Notice that the lower bound on $\beta_\rho(s)$ requires that every state is reachable. The latter can be achieved with any sufficiently exploratory policy unless there are states that are attractive. The previous assumptions allow us to establish that $\nabla_h U_{s_k}(h)$ is an ascent direction for the function $U_{s_0}(h)$. Notice that for the latter to hold, we require that

$$\langle \nabla_h U_{s_0}(h), \nabla_h U_{s_k}(h) \rangle_{\mathcal{H}} \geq 0. \tag{33}$$

By writing the gradient as in (19) and using the reproducing property of the kernel it follows that the previous condition is equivalent to

$$\int D(s)^\top \rho_{s_0}(s) \kappa(s,s') D(s') \rho_{s_k}(s')\, ds ds' \geq 0. \tag{34}$$

where $(\cdot)^\top$ denotes transpose. Notice that if $\kappa(s,s')$ approaches a Dirac delta, the integral with respect to s', in the limit reduces to evaluating $D(s')\rho_{s_k}(s')$ at $s' = s$. Thus, the double integral is an approximation of

$$\int \left\| D(s) \right\|^2 \rho_{s_0}(s) \rho_{s_k}(s)\, ds \tag{35}$$

which is always non-negative. To formalize the previous argument we will consider a Gaussian Kernel and we will show that if the width of the kernel is small enough, then the previous result holds (Theorem 1). We require to establish first that $D(s)\rho_{s_0}(s)$ is bounded and Lipschitz. This is subject of the following lemma.

Lemma 2. *Let $\kappa_{\Sigma_{\mathcal{H}}}(s,s')$ be a matrix-valued Gaussian kernel with covariance matrix $\Sigma_{\mathcal{H}} \succ 0$, i.e. for all $i = 1, \ldots, p$ we have that*

$$\kappa_{\Sigma_{\mathcal{H}}}(s,s')_{ii} = e^{-(s-s')^\top \Sigma_{\mathcal{H}}^{-1} (s-s')/2}, \tag{36}$$

and $\kappa(s,s';\Sigma_{\mathcal{H}})_{ij} = 0$ for all $j = 1 \ldots p$ with $j \neq i$. Let B_r, L_{rs} and L_{ra} be the constants defined in Assumption 1. Likewise, let B_ρ, L_p, L_{ps} and L_{pa} be the constants defined in Assumption 2. Furthermore, define the following constants

$$B_D := \frac{\sqrt{2} B_r}{1-\gamma} \frac{\Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2}\right)}, \tag{37}$$

with $\Gamma(\cdot)$ being the Gamma function, $L_{Qs} = L_{rs} + \frac{B_r}{1-\gamma} L_{ps} |\mathcal{S}|$, $L_{Qa} = L_{ra} + \frac{B_r}{1-\gamma} L_{pa} |\mathcal{S}|$,

$$L_h := \left\| h \right\| \lambda_{\min}(\Sigma_{\mathcal{H}})^{-1/2}, \tag{38}$$

and $L_D := L_{Qs} + L_{Qa} L_h$. Then, we have that $D(s)\rho_{s_k}(s)$ for any $s_k \in \mathcal{S}_0$ is bounded by $B := B_\rho B_D$ and it is Lipschitz with constant $L := B_D L_p + B_\rho L_D$.

Proof. See Appendix B. ■

As it was previously discussed we require a Gaussian Kernel whose width is small enough for the inner product of gradients at different initial states to be positive. We next formalize this condition. Define the normalization factor $Z := \sqrt{\det 2\pi \Sigma_{\mathcal{H}}}$ and let

$$\sqrt{np}\left(1 + \frac{\beta_\rho}{B_\rho}\right) \left\| \Sigma_{\mathcal{H}} \right\| Z L(h, \Sigma_{\mathcal{H}}) B |\mathcal{S}| \leq \frac{\varepsilon}{2} \frac{\beta_\rho}{B_\rho}. \tag{39}$$

The previous condition in a sense defines the maximum width of the kernel. Since if the norm of $\Sigma_{\mathcal{H}}$ is large, the previous condition cannot hold. This intuition is not exact since the term Z includes the determinant of the matrix $\Sigma_{\mathcal{H}}$ and thus, it is possible to have a kernel that has some directions being wide as long as the product of the eigenvalues is small enough. Likewise the Lipschitz constant in (39) depends on the norm of the function h, and in that sense it is necessary to ensure that the norm remains bounded for said condition to hold. We are now in conditions of establishing the main result in this work, which states that as long as the norm of $\nabla_h U_{s_0}(h)$ is large, the gradient of any value function $\nabla_h U_{s_k}(h)$ is an ascent direction for $U_{s_0}(h)$. This result will be instrumental also to the proof of convergence of the online algorithm (Section V).

Theorem 1. *Under the hypotheses of Lemma 2, for every $\varepsilon > 0$ and for every $\mathcal{H}$ and $h \in \mathcal{H}$ satisfying (39) it holds that if $\left\| \nabla_h U_{s_0}(h, \cdot) \right\|_{\mathcal{H}}^2 \geq \varepsilon$ then we have that for all $k \geq 0$*

$$\langle \nabla_h U_{s_0}(h, \cdot), \nabla_h U_{s_k}(h, \cdot) \rangle_{\mathcal{H}} > \frac{\varepsilon}{2} \frac{\beta_\rho}{B_\rho}. \tag{40}$$

Proof. Consider the following integral, with the kernel covariance matrix $\Sigma_{\mathcal{H}}$ as a parameter

$$I_{\Sigma_{\mathcal{H}}} = \int D(s)^\top \rho_{s_l}(s) \kappa_{\Sigma_{\mathcal{H}}}(s,s') \rho_{s_k}(s') D(s')\, ds ds', \tag{41}$$

where $\kappa_{\Sigma_{\mathcal{H}}}(s,s')$ is a kernel of the form (36). Observe that by writing the gradients of $U_{s_0}(h)$ and $U_{s_k}(h)$ as in (19), it follows that $I_{\Sigma_{\mathcal{H}}}$ is the inner product in (40). Hence, to prove the claim, it suffices to show that for all $\Sigma_{\mathcal{H}}$ satisfying condition (39), $I_{\Sigma_{\mathcal{H}}} > \varepsilon \beta_\rho / (2B_\rho)$. To do so, apply the change of variables $u = s' - s$, and divide and multiply the previous expression by $Z := \sqrt{\det 2\pi \Sigma_{\mathcal{H}}}$ to write $I_{\Sigma_{\mathcal{H}}}$ as

$$\begin{aligned} I_{\Sigma_{\mathcal{H}}} &= \int D(s)^\top \rho_{s_0}(s) \kappa_{\Sigma_{\mathcal{H}}}(s, s+u) \rho_{s_k}(s+u) D(s+u) ds du \\ &= Z \int D(s)^\top \rho_{s_0}(s) g(u; 0, \Sigma_{\mathcal{H}}) \rho_{s_k}(s+u) D(s+u)\, ds du. \end{aligned} \tag{42}$$

where the normalization factor Z was introduced to identify $g(u; 0, \Sigma_{\mathcal{H}}) := \kappa_{\Sigma_{\mathcal{H}}}(s, s+u)_{ii}/Z$ as a Gaussian probability density function with zero mean and covariance $\Sigma_{\mathcal{H}}$ (cf. (36)). Then we write the partial integral with respect to u as the expectation of $D(s+u)\rho_{s_k}(s+u)$,

$$I_{\Sigma_{\mathcal{H}}} = Z \int D(s)^\top \rho_{s_0}(s) \mathbb{E}_{u \sim \mathcal{N}(0, \Sigma_{\mathcal{H}})} \left[D(s+u) \rho_{s_k}(s+u) \right] ds. \tag{43}$$

From Lemma 2 it follows that $D(s)\rho_{s_k}(s)$ is Lipschitz with constant L. Then, by virtue of [29, Theorem 1] we have that

$$\|\mathbb{E}\left[D(s+u)\rho_{s_k}(s+u)\right] - D(s)\rho_{s_k}(s)\| \leq \sqrt{np}\,\|\Sigma_{\mathcal{H}}\|\,L. \tag{44}$$

where again the expectation is taken with respect to the random variable $u \sim \mathcal{N}(0, \Sigma_{\mathcal{H}})$. The result in (44) allows us to lower bound $I_{\Sigma_{\mathcal{H}}}$ by

$$\begin{aligned} I_{\Sigma_{\mathcal{H}}} &\geq Z \int \|D(s)\|^2 \rho_{s_0}(s)\rho_{s_k}(s)\,ds \\ &\quad - Z\sqrt{np}\,\|\Sigma_{\mathcal{H}}\|\,L \int \|D(s)\rho_{s_0}(s)\|\,ds \\ &= \bar{I}_{\Sigma_{\mathcal{H}}} - \sqrt{np}\,\|\Sigma_{\mathcal{H}}\|\,ZL \int \|D(s)\rho_{s_0}(s)\|\,ds \end{aligned} \tag{45}$$

where $\bar{I}_{\Sigma_{\mathcal{H}}}$ was implicitly defined in (45) as

$$\bar{I}_{\Sigma_{\mathcal{H}}} := Z \int \|D(s)\|\,\rho_{s_0}(s)\rho_{s_k}(s)ds \tag{46}$$

$$= \sqrt{\det 2\pi\Sigma_{\mathcal{H}}} \int \|D(s)\|^2\,\rho_{s_0}(s)\rho_{s_k}(s)ds \tag{47}$$

Let us next define the following integrals, identical to $I_{\Sigma_{\mathcal{H}}}$ and $\bar{I}_{\Sigma_{\mathcal{H}}}$, but for ρ_{s_0} substituting ρ_{s_k}

$$J_{\Sigma_{\mathcal{H}}} := \int D(s)'\rho_{s_0}(s)\kappa_{\Sigma_{\mathcal{H}}}(s,s')D(s')\rho_{s_0}(s')\,dsds' \tag{48}$$

$$\bar{J}_{\Sigma_{\mathcal{H}}} := Z \int \|D(s)\|^2 \rho_{s_0}^2(s)ds. \tag{49}$$

and use the bounds on the probability distribution (cf., Assumption 2) to write

$$\bar{J}_{\Sigma_{\mathcal{H}}} \leq Z \int \|D(s)\|^2 \rho_{s_0}(s)\frac{B_p}{\beta_\rho}\rho_{s_k}(s)\,ds = \frac{B_p}{\beta_\rho}\bar{I}_{\Sigma_{\mathcal{H}}}. \tag{50}$$

Hence, we can write (45) as

$$I_{\Sigma_{\mathcal{H}}} \geq \frac{\beta_\rho}{B_\rho}\bar{J}_{\Sigma_{\mathcal{H}}} - \sqrt{np}\,\|\Sigma_{\mathcal{H}}\|\,ZL \int \|D(s)\rho_{s_0}(s)\|\,ds. \tag{51}$$

Repeating steps (42)-(45), after substituting ρ_{s_0} for ρ_{s_k}, we can bound the difference between $J_{\Sigma_{\mathcal{H}}}$ and $\bar{J}_{\Sigma_{\mathcal{H}}}$ as we did it for $I_{\Sigma_{\mathcal{H}}}$ and $\bar{I}_{\Sigma_{\mathcal{H}}}$ in (45). Specifically, the following inequality holds

$$\bar{J}_{\Sigma_{\mathcal{H}}} \geq J_{\Sigma_{\mathcal{H}}} - \sqrt{np}\,\|\Sigma_{\mathcal{H}}\|\,ZL \int \|D(s)\rho_{s_0}(s)\|\,ds \tag{52}$$

This allows us to further lower bound $I_{\Sigma_{\mathcal{H}}}$ by

$$I_{\Sigma_{\mathcal{H}}} \geq \frac{\beta_\rho}{B_\rho}J_{\Sigma_{\mathcal{H}}} - \sqrt{np}\left(1 + \frac{\beta_\rho}{B_\rho}\right)\|\Sigma_{\mathcal{H}}\|\,ZL \int \|D(s)\rho_{s_0}(s)\|\,ds \tag{53}$$

By virtue of Lemma 2 we have that $\|D(s)\rho_{s_0}(s)\| \leq B$. Defining $|\mathcal{S}|$ as the measure of the set $\mathcal{S}$, $I_{\Sigma_{\mathcal{H}}}$ can be further lower bounded by

$$I_{\Sigma_{\mathcal{H}}} \geq \frac{\beta_\rho}{B_\rho}J_{\Sigma_{\mathcal{H}}} - \sqrt{np}\left(1 + \frac{\beta_\rho}{B_\rho}\right)\|\Sigma_{\mathcal{H}}\|\,ZLB|\mathcal{S}|. \tag{54}$$

Notice that $J_{\Sigma_{\mathcal{H}}} = \|\nabla_h U_{s_0}(h,\cdot)\|^2 \geq \varepsilon$, hence the previous inequality reduces to

$$I_{\Sigma_{\mathcal{H}}} \geq \frac{\beta_\rho}{B_\rho}\varepsilon - \sqrt{np}\left(1 + \frac{\beta_\rho}{B_\rho}\right)\|\Sigma_{\mathcal{H}}\|\,ZLB|\mathcal{S}|. \tag{55}$$

Then, for any $\Sigma_{\mathcal{H}}$ satisfying (39), we can lower bound the right hand side of the previous expression by $\varepsilon\beta_\rho/(2B_\rho)$, obtaining

$$I_{\Sigma_{\mathcal{H}}} \geq \frac{\beta_\rho}{2B_\rho}\varepsilon, \tag{56}$$

which completes the proof of the theorem. ∎

The previous result establishes that for kernels that satisfy the condition (39) with h outside of an ε neighborhood of the critical points, i.e., for h such that $\|\nabla_h U_{s_0}(h)\| > \varepsilon$, the inner product between $\nabla_h U_{s_k}(h)$ and $\nabla_h U_{s_0}(h)$ is larger than a constant that depends on ε. The latter means that for all state $s_k \in \mathcal{S}$, $\nabla_h U_{s_k}(h)$ is an ascent direction of the function $U_{s_0}(h)$. In the next section we exploit this idea to show that the online gradient ascent algorithm proposed in Section III converges with probability one to a neighborhood of a critical point of $U_{s_0}(h)$.

V. Convergence Analysis of Online Policy Gradient

Let $(\Omega, \mathcal{F}, P)$ be a probability space and define the following sequence of increasing sigma-algebras $\{\emptyset, \Omega\} = \mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}_k \subset \ldots \subset \mathcal{F}_\infty \subset \mathcal{F}$, where for each k we have that $\mathcal{F}_k$ is the sigma algebra generated by the random variables $h_0, \ldots, h_k$. For the purpose of constructing a submartingale that will be used in the proof of convergence, we provide a lower bound on the expectation of random variables $U_{s_0}(h_{k+1})$ conditioned to the sigma field $\mathcal{F}_k$ in the next Lemma

Lemma 3. *Choosing the compression budget $\epsilon_K = K\eta$ with $K > 0$, the sequence of random variables $U(h_k)$ satisfies the following inequality*

$$\begin{aligned} \mathbb{E}\left[U_{s_0}(h_{k+1})|\mathcal{F}_k\right] &\geq U_{s_0}(h_k) - \frac{\eta^2}{Z}C_1 - \frac{\eta^3}{Z^{3/2}}C_2 \\ &- \|\nabla_h U_{s_0}(h_k)\|_{\mathcal{H}}\,K\eta + \eta\left\langle \nabla_h U_{s_0}(h_k), \nabla_h U_{s_k}(h_k)\right\rangle_{\mathcal{H}}, \end{aligned} \tag{57}$$

where C_1 and C_2 are the following positive constants

$$C_1 = L_1\left(\sigma^2 + 2K\sigma + K^2\right) \tag{58}$$

and

$$C_2 = L_2\left(\sigma^2 + 2K\sigma + K^2\right)^{3/2}, \tag{59}$$

where L_1 and L_2 are given by

$$L_1 = B_r\frac{(1-\gamma+p(1+\gamma))}{\lambda_{\min}\Sigma(1-\gamma)^2}, L_2 = B_r\frac{(1+p)\sqrt{p}}{\lambda_{\min}(\Sigma)^{3/2}(1-\gamma)^3}, \tag{60}$$

and

$$\sigma = \frac{(3\gamma)^{1/3}}{\lambda_{\min}(\Sigma^{1/2})(1-\gamma)^2}\left(4\frac{\Gamma(2+p/2)}{\Gamma(p/2)}\right)^{1/4}. \tag{61}$$

Proof. Start by writing the Taylor expansion of $U_{s_0}(h_{k+1})$ around h_k

$$U_{s_0}(h_{k+1}) = U_{s_0}(h_k) + \left\langle \nabla_h U_{s_0}(f_k,\cdot), h_{k+1} - h_k\right\rangle_{\mathcal{H}}. \tag{62}$$

where $f_k = \lambda h_k + (1-\lambda)h_{k+1}$ with $\lambda \in [0,1]$. From [14, Lemma 5] we have that

$$\left\|\nabla_h U_{s_0}(g) - \nabla_h U_{s_0}(h)\right\|_{\mathcal{H}} \leq L_1 \left\|g-h\right\|_{\mathcal{H}} + L_2 \left\|g-h\right\|_{\mathcal{H}}^2, \tag{63}$$

with L_1 and L_2 being the constants in (60). Adding and subtracting $\langle \nabla_h U_{s_0}(h_k,\cdot), h_{k+1}-h_k\rangle_{\mathcal{H}}$ to the previous expression, using the Cauchy-Schwartz inequality and (63) we can re write the previous expression as

$$\begin{aligned} U_{s_0}(h_{k+1}) &= U_{s_0}(h_k) + \langle \nabla_h U_{s_0}(h_k,\cdot), h_{k+1}-h_k\rangle_{\mathcal{H}} \\ &\quad + \langle \nabla_h U_{s_0}(f_k,\cdot) - \nabla_h U_{s_0}(h_k,\cdot), h_{k+1}-h_k\rangle_{\mathcal{H}} \\ &\geq U_{s_0}(h_k) + \langle \nabla_h U_{s_0}(h_k,\cdot), h_{k+1}-h_k\rangle_{\mathcal{H}} \\ &\quad - L_1 \left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^2 - L_2 \left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^3. \end{aligned} \tag{64}$$

Let us consider next the conditional expectation of the random variable $U_{s_0}(h_{k+1})$ with respect to the sigma-field $\mathcal{F}_k$. Combine the monotonicity and the linearity of the expectation with the fact that h_k is measurable with respect to $\mathcal{F}_k$ to write

$$\begin{aligned} \mathbb{E}\left[U_{s_0}(h_{k+1})|\mathcal{F}_k\right] &\geq U_{s_0}(h_k) \\ &+ \langle \nabla_h U_{s_0}(h_k,\cdot), \mathbb{E}\left[h_{k+1}-h_k|\mathcal{F}_k\right]\rangle_{\mathcal{H}} \\ -L_1\mathbb{E}\left[\left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^2|\mathcal{F}_k\right] &- L_2\mathbb{E}\left[\left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^3|\mathcal{F}_k\right]. \end{aligned} \tag{65}$$

Using the result of Proposition 2 we can write the expectation of the quadratic term in the right hand side of (65) as

$$\begin{aligned} & L_1\mathbb{E}\left[\left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^2|\mathcal{F}_k\right] \\ &\leq L_1\eta^2\mathbb{E}\left[\left\|\hat{\nabla}_h U_{s_k}(h_k,\cdot)\right\|_{\mathcal{H}}^2|\mathcal{F}_k\right] + L_1\epsilon_K^2 \\ &\quad +2L_1\eta\epsilon_K\mathbb{E}\left[\left\|\hat{\nabla}_h U_{s_k}(h_k,\cdot)\right\|_{\mathcal{H}}|\mathcal{F}_k\right], \end{aligned} \tag{66}$$

Using the bounds on the moments of the estimate (cf., [14, Lemma 6]), the previous expression can be upper bounded by

$$L_1\mathbb{E}\left[\left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^2|\mathcal{F}_k\right] \leq \eta^2 L_1\left(\sigma^2 + 2\frac{\epsilon_K}{\eta}\sigma + \frac{\epsilon_K^2}{\eta^2}\right). \tag{67}$$

where σ is the constant in (61). Choosing the compression budget as $\epsilon_K = K\eta$ and using the definition of C_1 in (58) it follows that

$$L_1\mathbb{E}\left[\left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^2|\mathcal{F}_k\right] = \eta^2 L_1\left(\sigma^2+2K\sigma+K^2\right) = \eta^2 C_1. \tag{68}$$

Likewise, we have that

$$\begin{aligned} L_2\mathbb{E}\left[\left\|h_{k+1}-h_k\right\|_{\mathcal{H}}^3|\mathcal{F}_k\right] &\leq \eta^3 L_2\left(\sigma^2 + 2\frac{\epsilon_K}{\eta}\sigma + \frac{\epsilon_K^2}{\eta^2}\right)^{3/2} \\ &= \eta^3 L_2\left(\sigma^2+2K\sigma+K^2\right)^{3/2} \\ &= \eta^3 C_2. \end{aligned} \tag{69}$$

Replacing the previous two bounds regarding the moments of $\|h_{k+1}-h_k\|$ in (65) reduces to

$$\begin{aligned} \mathbb{E}\left[U(h_{k+1})|\mathcal{F}_k\right] &\geq U(h_k) - \eta^2 C_1 - \eta^3 C_2 \\ &\quad + \langle \nabla_h U(h_k), \mathbb{E}\left[h_{k+1}-h_k|\mathcal{F}_k\right]\rangle_{\mathcal{H}}. \end{aligned} \tag{70}$$

Using the result of Proposition 2 and the fact that $\hat{\nabla}_h U_{s_k}(h_k)$ is unbiased (cf., Proposition 1) we can write the inner product in the previous equation as

$$\begin{aligned} \langle \nabla_h U_{s_0}(h_k), \mathbb{E}\left[h_{k+1}-h_k|\mathcal{F}_k\right]\rangle_{\mathcal{H}} &= \\ \eta\langle \nabla_h U_{s_0}(h_k), \nabla_h U_{s_k}(h_k)\rangle_{\mathcal{H}} + \langle \nabla_h U_{s_0}(h_k), b_k\rangle_{\mathcal{H}}&. \end{aligned} \tag{71}$$

The proof is then completed using the Cauchy-Schwartz inequality and fact that the norm of the bias is bounded by $\epsilon_K = K\eta$ (cf., Proposition 2). ■

The previous lemma establishes a lower bound on the expectation of $U_{s_0}(h_{k+1})$ conditioned to the sigma algebra $\mathcal{F}_k$. This lower bound however, is not enough for $U_{s_0}(h_k)$ to be a submartingale, since the sign of the term added to $U_{s_0}(h_k)$ in the right hand side of (57) is not necessarily positive. The origin of this is threefold. The first two reasons stem from algorithmic reasons. These are that we are using the estimate of $\nabla_h U_{s_k}(h_k)$ to ascend on the functionial $U_{s_0}(h)$ – which does not guarantee the inner product to be always positive – the bias that results from projecting into a lower dimension via the KOMP algorithm as stated in Proposition 2. The third reason comes from the analysis in Lemma 3 where we bounded the value of the functional using a first order approximation. To overcome the first limitation we will use the result from Theorem 1 that guarantees that the inner product in the right hand side of (57) is lower bounded by $\varepsilon\beta_\rho/(2B_\rho)$ as long as $\left\|\nabla_h U_{s_0}(h)\right\|^2 > \varepsilon$. The latter suggests that the definition of the following stopping time is necessary for the analysis

$$N = \min_{k\geq 0}\left\{\left\|\nabla_h U_{s_0}(h_k,\cdot)\right\|_{\mathcal{H}}^2 \leq \varepsilon\right\}. \tag{72}$$

We will show that by choosing the compression factor ϵ_K and the step size sufficiently small we can overcome the other two limitations and establish that $U_{s_0}(h_k)$ is a submartingale as long as $k < N$. To be able to use the result of Theorem 1 we require, condition (39) to be satisfied. As previously explained, this requires the norm of h not to grow unbounded, yet due to the stochastic nature of the update there is no guarantees that this will be the case. We assume, however, that policies with infinite norm are poor policies which leads to the conclusion that if the norm of the gradient is not too small, then it has to be the case that the norm of h is bounded. We formalize these ideas next.

Assumption 3. For every $\mathcal{H}$ it follows that $\lim_{\|h\|\to\infty} U_{s_0}(h) = \min_{h\in\mathcal{H}} U_{s_0}(h)$.

Lemma 4. *For every $\varepsilon > 0$ there exists a constant $B_h(\varepsilon)$ such that if $\left\|\nabla_h U_{s_0}(h)\right\|^2 \geq \varepsilon$ then $\|h\| \leq B_h(\varepsilon)$.*

Proof. Since the function $U_{s_0}(h)$ is bounded (cf., [14, Lemma 1]), Assumption 3 implies that $\lim_{\|h\|\to\infty}\nabla_h U_{s_0}(h) = 0$ and therefore for every $\varepsilon > 0$ there exists $B_h(\varepsilon) > 0$ such that if $\|h\| > B_h(\varepsilon)$ then $\left\|\nabla_h U_{s_0}(h)\right\| < \varepsilon$. Hence it has to be the case that if $\left\|\nabla_h U_{s_0}(h)\right\|^2 \geq \varepsilon$, then $\|h\| \leq B_h(\varepsilon)$. ■

As previously discussed to guarantee that $U_{s_0}(h_k)$ is a submartingale we need to choose the compression budget ϵ_K and the step-size η small enough. In particular observe that the compression budget multiplies a term that depends on the norm of the gradient in (57). Hence, to be able to guarantee that reducing the compression budget is enough to have a submartingale we require that the norm of the gradient of the value function is bounded. This is the subject of the following lemma.

Lemma 5. *The norm of the gradient of $\nabla_h U_{s_0}(h)$ is bounded by B_∇ where*

$$B_\nabla := \frac{\sqrt{p}B_r}{(1-\gamma)^2 \lambda_{\min}\Sigma^{1/2}}. \tag{73}$$

Proof. Use (7), the fact that $|Q(s,a;h)| \leq B_r/(1-\gamma)$ (cf., [14, Lemma 1]) and that $\|\kappa(s,\cdot)\| = 1$ to upper bound the norm of $\nabla_h U_{s_0}(h)$ by

$$\|\nabla_h U_{s_0}(h)\|^2 \leq \frac{B_r^2}{(1-\gamma)^4}\mathbb{E}\left[\left\|\Sigma^{-1}(a-h(s))\right\|^2 |h\right]. \tag{74}$$

Since the action a is drawn from a normal distribution with mean $h(s)$ and covariance matrix Σ it follows that $\left\|\Sigma^{-1/2}(a-h(s))\right\|^2$ is a χ^2 distribution and thus its expectation is p. Hence, the above expectation is bounded by $p\lambda_{\min}(\Sigma)^{-1}$. This completes the proof of the result. ■

We are now in conditions of introducing the convergence of the online policy gradient algorithm presented in Section III to a neighborhood of the critical points of the value functional $U_{s_0}(h)$. In addition, the update is such that it guarantees that the model order remains bounded for all iterations.

Theorem 2. *Let Assumptions 1–3 hold. For any $\varepsilon > 0$ chose K such that*

$$K < \frac{\varepsilon}{2B_\nabla}\frac{\beta_\rho}{B_\rho}, \tag{75}$$

algorithm step-size $\eta > 0$ such that

$$\eta \leq \frac{\sqrt{C_1^2 + 4C_2\left(\frac{\varepsilon\beta_\rho}{2B_\rho} - B_\nabla K\right)} - C_1}{2C_2}. \tag{76}$$

and compression budget of the form $\epsilon_K = K\eta$. Under the hypotheses of Lemma 2, and for any kernel such that $\Sigma_{\mathcal{H}}$ verifies (39)*, the sequence of policies that arise from Algorithm 2 satisfy that $\liminf_{k\to\infty} \|\nabla_h U_{s_0}(h_k)\|^2 < \varepsilon$. In addition, let M_k be the model order of h_k, i.e., the number of kernels which expand h_k after the pruning step KOMP. Then, there exists a finite upper bound M^∞ such that, for all $k \geq 0$, the model order is always bounded as $M_k \leq M^\infty$.*

Proof. Define the following sequence of random variables

$$V_k = (U(h^\star) - U(h_k))\,\mathbb{1}(k \leq N), \tag{77}$$

with $\mathbb{1}(\cdot)$ being the indicator function and N the stopping time defined in (72). We next work towards showing that V_k is a non-negative submartingale. Because $U(h^\star)$ maximizes $U(h)$, V_k is always non-negative. In addition $V_k \in \mathcal{F}_k$ since $U(h_k) \in \mathcal{F}_k$ and $\mathbb{1}(k \leq N) \in \mathcal{F}_k$. Thus, it remains to be shown that $\mathbb{E}[V_{k+1}|\mathcal{F}_k] \leq V_k$. Notice that for any $k > N$ it follows that $\mathbb{1}(k \leq N) = 0$ and hence $V_k = 0$. Thus we have that $V_{k+1} = V_k$ for all $k \geq N$. We are left to show that $\mathbb{E}[V_{k+1}|\mathcal{F}_k] \leq V_k$ for $k \leq N$. Using the result of Lemma 3 we can upper bound $\mathbb{E}[V_{k+1}|\mathcal{F}_k]$ as

$$\begin{aligned}\mathbb{E}[V_{k+1}|\mathcal{F}_k] \leq V_k + \eta^2 C_1 + \eta^3 C_2 + K\eta \left\|\nabla_h U_{s_0}(h_k)\right\| \\ -\eta \left\langle \nabla_h U_{s_0}(h_k), \nabla_h U_{s_k}(h_k)\right\rangle_{\mathcal{H}}.\end{aligned} \tag{78}$$

Since we have that $\|\nabla_h U_{s_0}(h_k)\|^2 \geq \varepsilon$ by virtue of Assumption 3 it follows that $\|h\|_{\mathcal{H}} \leq B_h(\varepsilon)$. Therefore, there exists some Hilbert Space for which condition (39) holds and the result of Theorem 1 implies that the inner product in the right hand side of the previous expression is lower bounded by $\varepsilon\beta_\rho/(2B_\rho)$. In addition, the norm of $\nabla_h U_{s_0}(h_k)$ is bounded by virtue of Lemma 5. Hence, the previous expression can be further upper bounded by

$$\begin{aligned}\mathbb{E}[V_{k+1}|\mathcal{F}_k] &\leq V_k + \eta^2 C_1 + \eta^3 C_2 + \eta K B_\nabla - \eta\frac{\varepsilon}{2}\frac{\beta_\rho}{B_\rho} \\ &= V_k + \eta\alpha(K,\eta),\end{aligned} \tag{79}$$

where we define $\alpha(K,\eta)$ as

$$\alpha(K,\eta) := KB_\nabla - \frac{\varepsilon}{2}\frac{\beta_\rho}{B_\rho} + \eta C_1 + \eta^2 C_2. \tag{80}$$

With the condition for the compression factor satisfying (75) we guarantee that the sum of the first two terms on the right hand side of (80) is negative. The latter is sufficient to guarantee that the expression is negative for all η satisfying (76). This completes the proof that V_k is a non-negative submartingale. Thus, V_k converges to random variable V such that $\mathbb{E}[V] \leq \mathbb{E}[V_0]$ (see e.g., [25, Theorem 5.29]). Then, by unrolling (79) we obtain the following upper bound for the expectation of V_{k+1}

$$\mathbb{E}\left[V_{k+1}\right] \leq V_0 + \alpha\eta\mathbb{E}N. \tag{81}$$

Since V_{k+1} is bounded the Dominated Convergence Theorem holds and we have that

$$\mathbb{E}[V] = \lim_{k\to\infty}\mathbb{E}\left[V_{k+1}\right] \leq V_0 + \alpha\eta\mathbb{E}N. \tag{82}$$

Since $\alpha < 0$, rearranging the terms in the previous expression we can upper bound $\mathbb{E}N$ by

$$\mathbb{E}N \leq \frac{\mathbb{E}V - V_0}{\eta\alpha}. \tag{83}$$

Therefore it must be case that $P(N = \infty) = 0$. Which implies that the event $\|\nabla_h U_{s_0}(h_k)\| < \varepsilon$ occurs infinitely often. Thus completing the proof of the result. It remains to be shown that the model order of the representation is bounded for all k. The proof of this result is identical to that in [22, Theorem 3]. ■

The previous result establishes the convergence to a neighborhood of the critical points of $U_{s_0}(h)$ of the online gradient ascent algorithm presented in Section III. For such result to hold, we require that the kernel width, the compression budget and the step size to be small enough. In addition, the compression introduced by KOMP guarantees that the model order of the function h_k remains bounded for all $k \geq 0$. In the next section we explore the implications of these theoretical results in a cyclic navigation problem.

VI. Numerical Experiments

Next we test the performance of our non-epsodic RL method in a suvelliance and navigation task. The setup includes an area in $\mathbb{R}^2$ with a point to be surveilled located at $x_g = [-1, -5]$ and a battery charger located at $x_b = [-1, 5]$. These points are depicted in green and red, respectively, in Figure 1. An agent starts moving from its initial point at $x_0 = [3, 0]$, depicted in blue in figures 1 and 2, towards its goal at x_g. The agent model consist in second order point mass acceleration dynamics, with position $x \in \mathbb{R}^2$ and velocity $v \in \mathbb{R}^2$ as state variables. It also includes second order battery charging and discharging equations with state variables $b \in \mathbb{R}$ modeling the remaining charge of the battery, and $d \in \mathbb{R}$ representing the difference between charge levels at two consecutive time instants. The battery charges at a constant rate ΔB if the agent is located within a neighborhood of the battery charger, and discharges at the same rate ΔB otherwise. Vector $s = (x, v, b, d)$ collects all the state variables of this model. The reward is shaped so that the agent is stimulated to move towards the goal x_g when b is grater than or equal to 40% of its full capacity and it is discharging $d < 0$. And towards the battery charger x_c if b is lower than 40% and discharging $d < 0$, or if b is lower than 90% and charging $d > 0$. The use of a second order model for the battery allow us to leave room for some hysteresis on the charging and discharging loop, so that the battery does not start discharging as soon as b surpasses the 40% level. Instead, it keeps charging until it reaches 90% of its capacity before moving back towards the goal. A logarithmic barrier is added to the reward for helping the agent to avoid an elliptic obstacle centered at $x_0 = [0, 0]$ with horizontal and vertical axes of length 1.8 and 0.9, respectively Under these dynamics, the agent decides its acceleration $a_k \in \mathbb{R}^2$ using the randomized Gaussian policy $a_k \sim \pi_{h_k}(.|s_k)$ where the mean of $a_k \sim \pi_{h_k}(.|s_k)$ is the kernel expansion $h_k(s_k)$ updated via (13). The Q-function in step 7 of Algorithm 1 is estimated as the sum of T_Q consecutive rewards with T_Q drawn from a geometric distribution of parameter γ. The Gaussian noise n_k that is added to $h_k(s_k)$ in step 2 of Algorithm 1 is selected at random when k is even, and equal to $n_k = -n_{k-1}$ when k is odd. This is a practical trick to improve the ascending direction of the stochastic gradient without adding bias or violating the Gaussian model for the randomized policy $\pi_h(a|s)$, see [14] for more details.

Figure 1 shows the trajectory of the agent, with its color changing gradually from blue to red as it starts from x_0 and loops between x_g and x_b. The four stages of this looping trajectory are detailed in Figure 2, with Figure 2 (left) showing the trace from x_0 to the neighborhood of x_g which includes some initial exploring swings. Then the trajectory in 2 (center left) starts when the agent's battery crosses the threshold of 40%. In this case the agent is rewarded for moving towards the charger and staying in its neighborhood until b reaches 90%. The next stage in Figure 2 (center right) starts when the battery level reaches 90% and the agent moves back to the goal. And finally the trace in Figure 2 (right) starts when the battery discharges under the safety level of 40%, moving back towards the charger and closing the loop.

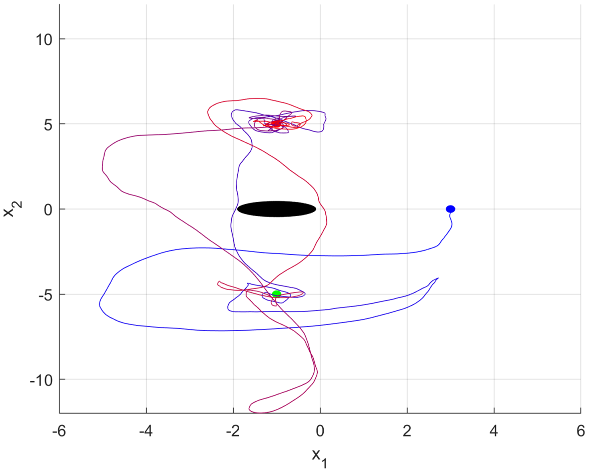

Fig. 1. Online cycling trajectory of an agent starting at $x_0 = [3, 0]$ with the goal of surveilling the location represented by a green point at $x_b = [-1, -5]$. The agent needs to recharge its battery when it discharges below 40% of its maximum capacity, at the charger location $x_b = [-1, 5]$ represented by the red point. When navigating towards x_b the agent must avoid the ellipsoidal obstacle centered at $[-1, 0]$.

This coherence between the battery level and the agent trajectories is further illustrated in figures 3 and 4. Figure 3 depicts the battery level across iterations in time, alongside the vertical position of the agent. The vertical position shows an oscillating step-response like behavior, as the agent reaches the neighborhood of the goal and the charger sequentially and hovers around them. The battery level shows the desired hysteresis, transitioning according to the thresholds at 40% and 90% and depending on the charging slope.

Such a loop is further evidenced in Figure 4, which represents the agent's vertical position versus its battery level. The horizontal dashed lines represent the full charge and low battery thresholds, and the vertical dashed lines correspond to the positions of the goal and the charger. This figure shows how starting from x_0 the agent moves towards x_g until it reaches the 40% level, and then towards the charger hovering around it until the battery charge is 90%. Then it closes the loop by moving towards the goal and the charger sequentially while its battery charges and discharges.

Figure 5 is included to corroborate the theoretical findings of Section IV. More specifically, it depicts the evolution of $U_s(h_k)$ as a function of the online iteration index k. The starting point s in $U_s(h_k)$ is the same for all k, and it is selected as the state when the battery level crosses the line of 40% for the first time. It corresponds to the location near x_g where the stage 2 starts in Figure 2(b). For completeness $s = [x, v, b, d]$ with $x = [-0.72, -4.58]$, $v = [-0.092, 0.049]$, $b = 39.99$, and $d = 3 \times 10^{-4}$. The value function $U_s(h_k)$ is estimated using rewards obtained by an episodic agent that starts at s and runs $N = 100$ sample trajectories of length T selecting actions according to the policy h_k, which is kept constant during the T state transitions. These rewards are

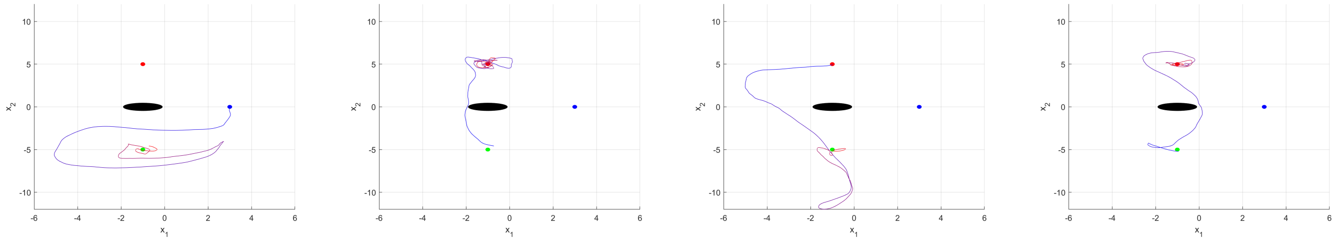

Fig. 2. Detail of the four stages of the online trajectory in Figure 1. Trace from x_0 to the neighborhood of x_g (left). Trace when the agent's battery crosses the threshold of 40% (center left). Trace when the battery level reaches 90% and the agent moves back to the goal (center right). Trace when the battery discharges under the safety level of 40% (right).

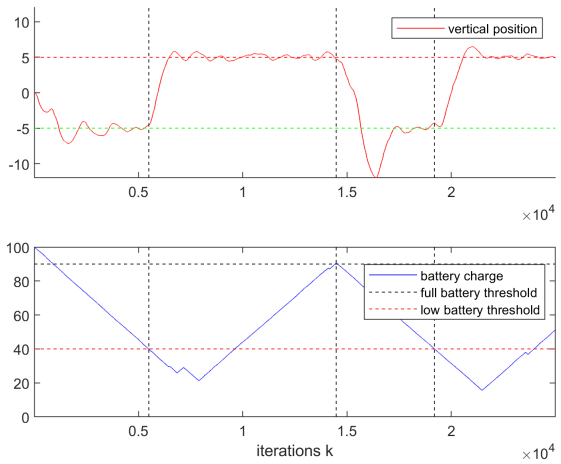

Fig. 3. Vertical position and battery charge as a function of the online gradient update index k. The horizontal dashed lines correspond to the positions of the goal and the charger, and to the battery safety and fully charged thresholds. The vertical dashed lines mark the transitions when the agent is directed to move to the charger or to the goal.

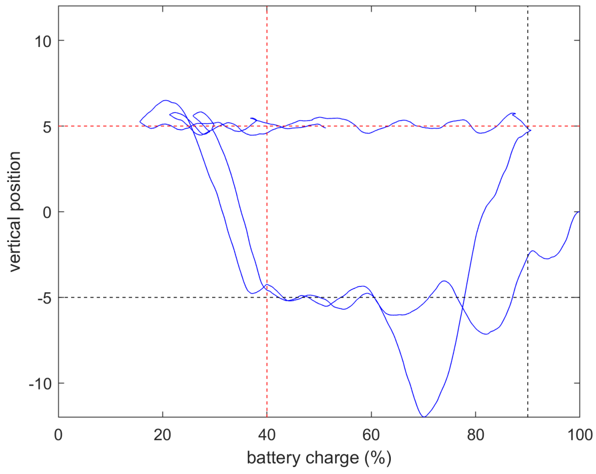

Fig. 4. Cyclic evolution of the agent's vertical position versus the battery charge. Horizontal dashed lines correspond to the positions of the goal and the charger, and vertical dashed lines represent the battery safety and fully charged thresholds.

averaged according to

$$\hat{U}_s(h_k) = \frac{1}{N}\sum_{i=1}^{N} R_{ik} \tag{84}$$

where $R_{ik} = \sum_{t=0}^{T} \gamma^t r_{itk}$ and r_{itk} is the instantaneous reward obtained by the episodic agent at time t, using policy h_k, over the sample trajectory i. These episodic trajectories are carried out for assessing performance of a fixed h_k, but the algorithm for updating these policies is non-episodic, as it evolves in the fully online fashion of (13).

The horizon $T = 100$ for these episodes was selected so that that the discarded tail of the geometric series becomes negligible, with $\gamma^T \simeq 2 \times 10^{-5}$ staying under the noise deviation. It is remarkable that when the online agent travels through s on its online journey of Figure 1, the policy figures out how to increase the reward. And such a reward will not decrease when the policy is updated in the future. This is coherent with our theoretical findings in Theorem 1, which states that gradients at future states are ascent directions for the value function at a previous state, that is s in this case.

As stated before, each point on the blue line in Figure 5 represents the mean of rewards in (84), and it is accompanied by its deviation interval. Notice that, even if the improvement in reward is relative minor, at 0.3%, it is good enough to direct the agent towards the battery charger. This can be better seen in the next figure.

Figure 6 shows five different trajectories starting at the same point $x = [-0.72, -4.58]$ represented by a blue dot. The trajectory that passes through the colored dots corresponds to an agent running our online algorithm, and coincides with part of the second stage in Figure 2 (center left). Let k_0, k_1, k_2, and k_3 be the iteration indexes when the online agent reaches the points $x_{k_0} = x$, x_{k_1}, x_{k_2}, x_{k_3}, represented by the blue, cyan, green, and purple dots, respectively. At these iterations the agent produces policies h_{k_0}, h_{k_1}, h_{k_2}, and h_{k_3}. The blue, cyan, green, and purple lines in Figure 6 represent the trajectories of an episodic agent starting at x and navigating with constant policies h_{k_0}, h_{k_1}, h_{k_2}, and h_{k_3}, respectively. Figure 6 corroborates that the policies improve as the online agent moves along its trajectory, allowing the episodic agent to navigate better. Indeed, at first the episodic agent only knows to go north west on the straight blue line, but eventually it manages to follow the purple line moving towards the charger and avoiding the obstacle. This apparent improvement

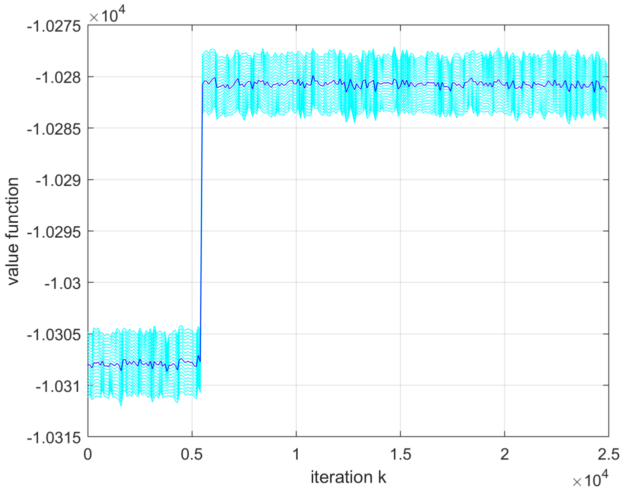

Fig. 5. Evolution of the mean accumulative reward in (84) as a function of the online iteration step k.

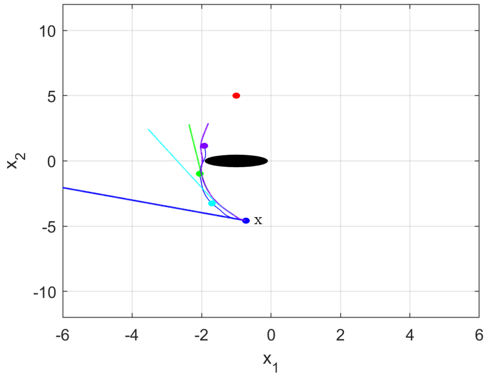

Fig. 6. Trajectories of an episodic agent using four policies that are produced by the online agent when following the cyclic trajectory of Figure 1. Each colored point represents a location x_k in which the online agent updates policy to obtain h_k. The line of the corresponding color represents the trajectory of the episodic agent that uses the fixed policy h_k to navigate from x towards the charger.

in Figure 6 is not reflected in a significant step increase in Figure 5. This is because the forgetting factor $\gamma = 0.9$ weights a few steps of the trajectory in the value function, and the fist steps are where the trajectories are not significantly separated.

Overall, this numerical example shows that the algorithm developed in this paper is capable of learning how to navigate on a loop in between to goal locations, avoiding an obstacle, and following a cyclic trajectory that does not comply with the standard stationary assumptions in the literature.

VII. Conclusion

We have considered the problem of learning a policy that belongs to a RKHS in order to maximize the functional defined by the expected discounted cumulative reward that an agent receives. In particular, we presented a fully online algorithm that accumulates at the critical points of the value function and keeps the model order of the representation of the function bounded for all iterations. The algorithm uses unbiased estimates of the gradient of the functionals conditioned at the current state that can be achieved in finite time. We establish that these gradients are also ascent directions for the initial value function for Gaussian kernels with small enough bandwidth. Therefore, by updating the policy following such gradients the value of the initial value function is increased in expectation at each iteration, when the step size and compression budget are small enough. We tested this algorithm in a navigation and surveillance problem whose cyclic nature highlights the ability to operate in a non stationary setup. The surveillance task is carried out while by training in a fully online fashion, without the need of episodic restarts. With this experiment we also corroborated our claim in Theorem 2 regarding the ascent directions of the stochastic gradients.

Appendix

A. Proof of Lemma 1

Proof. Since $s \in \mathcal{S}_0$ there exists some time $t \geq 0$ such that $p(s_t = s|s_0) > 0$. Therefore we have that

$$\begin{aligned}\rho_{s_0}(s) &= (1-\gamma)\sum_{u=0}^{\infty}\gamma^u p(s_u = s|s_0)\\ &\geq (1-\gamma)\gamma^t p(s_t = s|s_0),\end{aligned} \tag{85}$$

where the last inequality follows from the fact that $\gamma > 0$ and $p(s_u = s|s_0) \geq 0$ for all $u \geq 0$. Hence, by assumption it follows that $\rho_{s_0}(s) > 0$. To prove the second claim, start by writting $\rho_{s'}(s'')$ as

$$\begin{aligned}\rho_{s'}(s'') &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p(s_t = s''|s_0 = s')\\ &= (1-\gamma)\gamma^{-T}\sum_{u=T}^{\infty}\gamma^u p(s_u = s''|s_T = s'),\end{aligned} \tag{86}$$

where the last equality holds for any $T \geq 0$. Using the Markov property for any u we have that

$$p(s_u = s''|s_T = s') = p(s_u = s''|s_T = s', s_0). \tag{87}$$

Since $s' \in S_0$, there exists $T \geq 0$ such that $p(S_T = s'|s_0) > 0$. For that specific T, we have that

$$p(s_u = s''|s_T = s', s_0) = \frac{p(s_u = s'', s_T = s'|s_0)}{p(s_T = s'|s_0)}. \tag{88}$$

Notice next that since $s'' \in \mathcal{S} \setminus \mathcal{S}_0$ we have that $p(s_u = s''|s_0) = 0$ for all $u \geq 0$. Hence, we also have that $p(s_u = s'', S_T = s'|s_0) = 0$ for all $u \geq 0$ which completes the proof of the proposition. ■

B. Proof of Lemma 2

Without loss of generality, we prove the result for $D(s)\rho_{s_0}(s)$. We start by showing that the cumulative weighted distribution $\rho_{s_0}(s)$ is Lipschitz with constant L_p.

Lemma 6. *Under Assumption 2, the distribution $\rho_{s_0}(s)$ is Lipschitz with constant L_p, where L_p is the Lipschitz constant defined in Assumption 2.*

Proof. Let us start by writing $p(s_t = s|s_0)$ by marginalizing it

$$p(s_t = s|s_0) = \int p(s_t = s, a_{t-1}, s_{t-1}|s_0)\, ds_{t-1}da_{t-1}. \tag{89}$$

Using Bayes' rule and the Markov property of the transition probability it follows that

$$p(s_t = s|s_0) = \int p(s_t = s|a_{t-1}, s_{t-1})\pi_h(a_{t-1}|s_{t-1})p(s_{t-1}|s_0)\, ds_{t-1}da_{t-1}. \tag{90}$$

Using the Lipschitz property of the transition probability (cf., Assumption 2) we can upper bound $|p(s_t = s|s_0) - p(s_t = s'|s_0)|$ by

$$\begin{aligned}|p(s_t = s|s_0) - p(s_t = s'|s_0)| \leq L_p \left\| s - s' \right\| \\ \int \pi_h(a_{t-1}|s_{t-1})p(s_{t-1}|s_0)\, ds_{t-1}da_{t-1}\end{aligned} \tag{91}$$

Since both $\pi_h(a_{t-1}|s_{t-1})$and $p(s_{t-1}|s_0)$ are probability distributions they integrate one. Thus we have that

$$|p(s_t = s|s_0) - p(s_t = s'|s_0)| \leq L_p \left\| s - s' \right\|. \tag{92}$$

Use the definition of $\rho_{s_0}(s)$ (cf., (4)) to write the difference $|\rho_{s_0}(s) - \rho_{s_0}(s')|$ as

$$\begin{aligned}|\rho_{s_0}(s) - \rho_{s_0}(s')| = \\ (1-\gamma)\left| \sum_{t=0}^{\infty} \gamma^t \left(p(s_t = s|s_0) - p(s_t = s'|s_0)\right) \right|.\end{aligned} \tag{93}$$

Use the triangle inequality to upper bound the previous expression by

$$\begin{aligned}&|\rho_{s_0}(s) - \rho_{s_0}(s')| \\ &\leq (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \left| p(s_t = s|s_0) - p(s_t = s'|s_0) \right|.\end{aligned} \tag{94}$$

By virtue of (92), each term can be upper bounded by $\gamma^t L_p \left\| s - s' \right\|$. Thus

$$|\rho_{s_0}(s) - \rho_{s_0}(s')| \leq (1-\gamma)\sum_{t=0}^{\infty} \gamma^t L_p \left\| s - s' \right\| = L_p \left\| s - s' \right\|. \tag{95}$$

This completes the proof of the lemma. ■

Lemma 7. *Under the Assumptions of Lemma 2 $D(s)$ is bounded by B_D (cf.,* (37)*) and it is Lipschitz, i.e.,*

$$\left\| D(s) - D(s') \right\| \leq L_D \left\| s - s' \right\|, \tag{96}$$

were L_D is the constant defined in Lemma 2.

Proof. Let us start by introducing the change of variables $\zeta = \Sigma^{-1/2}(a - h(s))$ to compute $D(s)$. Hence we have that

$$D(s) = \int Q(s, h(s) + \Sigma^{1/2}\zeta)\frac{\zeta}{\sqrt{2\pi}^p}e^{-\|\zeta\|^2/2}\, d\zeta. \tag{97}$$

Notice that we can define $\phi(\zeta) = e^{-\|\zeta\|^2/2}/\sqrt{2\pi}^p$ and then, the previous expression reduces to

$$D(s) = -\int Q(s, h(s) + \Sigma^{1/2}\zeta)\nabla\phi(\zeta)\, d\zeta. \tag{98}$$

Thus, integrating each component of $D(s)$ by parts we have for each $i = 1, \ldots, n$ that

$$\begin{aligned}D(s)_i = \int Q(s, h(s) + \Sigma^{1/2}\zeta)\phi(\zeta)\Big|_{\zeta_i - \infty}^{\zeta_i = \infty} d\bar{\zeta}_i \\ + \int \frac{\partial Q(s, h(s) + \Sigma^{1/2}\zeta)}{\partial \zeta_i}\phi(\zeta)\, d\zeta,\end{aligned} \tag{99}$$

where $\bar{\zeta}_i$ denotes the integral with respect to all variables in ζ except for the i-th component. Since $Q(s,a) \leq B_r/(1-\gamma)$ [14, Lemma 3] and $\phi(\zeta)$ is a Multivariate Gaussian density the first term in the above sum is zero. Next we compute the derivative of the Q-function with respect to ζ. By the chain rule we have that

$$\frac{\partial Q(s, h(s) + \Sigma^{1/2}\zeta)}{\partial \zeta} = \Sigma^{1/2}\frac{\partial Q(s,a)}{\partial a}\Big|_{a = h(s) + \Sigma^{1/2}\zeta}. \tag{100}$$

Thus, (99) reduces to

$$D(s) = \int \frac{\partial Q(s,a)}{\partial a}\Big|_{a = h(s) + \Sigma^{1/2}\zeta}\phi(\zeta)\, d\zeta. \tag{101}$$

We claim that the first term in the above integral

$$Q'(s) := \frac{\partial Q(s,a)}{\partial a}\Big|_{a = h(s) + \Sigma^{1/2}\zeta} \tag{102}$$

is Lipschitz with constant L_D. That being the case, one has that

$$\begin{aligned}\left\| D(s) - D(s') \right\| &\leq \int \left\| Q'(s) - Q'(s') \right\| \phi(\zeta)\, d\zeta \\ &\leq L_D \left\| s - s' \right\| \int \phi(\zeta)\, d\zeta = L_D \left\| s - s' \right\|.\end{aligned} \tag{103}$$

Thus, to show that $D(s)$ is Lipschitz with constant L_D it remains to be showed that $Q'(s)$ is Lipschitz with the same constant. To do so, write the Q function as

$$Q(s,a) = r(s,a) + \sum_{t=1}^{\infty} \gamma^t \int r(s_t, a_t)p(s_t|a,s)\pi_h(a_t|s_t)\, ds_t da_t, \tag{104}$$

and compute its derivative with respect to a

$$\begin{aligned}Q'(s,a) = \frac{\partial Q(s,a)}{\partial a} = \frac{\partial r(s,a)}{\partial a} \\ + \sum_{t=1}^{\infty} \gamma^t \int r(s_t, a_t)\frac{\partial p(s_t|a,s)}{\partial a}\pi_h(a_t|s_t)\, ds_t da_t.\end{aligned} \tag{105}$$

Since $\partial r(s,a)/\partial a$ is Lipschitz in both arguments (cf., Assumption 1), to show that the derivative of Q is Lipschitz, it suffices to show that $\partial p(s_t|a,s)/\partial a$ is Lipschitz as well. To that end, write $p(s_t|s,a)$ as

$$\begin{aligned}p(s_t|s,a) = \\ \int p(s_1|s,a)\prod_{u=1}^{t-1}\pi_h(a_u|s_u)p(s_{u+1}|s_u, a_u)\, d\mathbf{s}_{t-1}d\mathbf{a}_{t-1},\end{aligned} \tag{106}$$

where $d\mathbf{s}_{t-1} = (ds_1, \cdots, ds_{t-1})$ and $d\mathbf{a}_{t-1} = (da_1, \cdots, da_{t-1})$. Let us define

$$\Delta p_t(s', s'', a', a'') := \frac{\partial p(s_t|s', a)}{\partial a}\Big|_{a=a'} - \frac{\partial p(s_t|s'', a)}{\partial a}\Big|_{a=a''}. \tag{107}$$

Using the fact that $\partial p(s_1|s, a)/\partial a$ is Lipschitz with respect to s and a with constants L_{ps} and L_{pa} (cf., Assumption 2) we have that

$$\begin{aligned}\|\Delta p_t(s, s', a, a')\| \leq \int (L_{ps}\|s' - s''\| + L_{pa}\|a' - a''\|)\\ \prod_{u=1}^{t-1} \pi_h(a_u|s_u) p(s_{u+1}|s_u, a_u)\, d\mathbf{s}_{t-1} d\mathbf{a}_{t-1}.\end{aligned} \tag{108}$$

Using the previous bound and (105), we can upper bound the norm of the difference $Q'(s, a) - Q'(s', a')$ as

$$\begin{aligned}\|Q'(s, a) - Q'(s', a')\| \leq L_{rs}\|s - s'\| + L_{ra}\|a - a'\|\\ + \sum_{t=1}^{\infty} \gamma^t \int B_r \|\Delta p(s, s', a, a')\| \pi_h(a_t|s_t)\, d\mathbf{a}_t d\mathbf{s}_t\end{aligned} \tag{109}$$

Because $p(s_{u+1}|s_u, a_u)$ and $\pi_h(a_u|s_u)$ in (108) are density functions they integrate to one. Hence, the integral in the previous expression can be upper bounded by

$$\begin{aligned}&\int \|\Delta p_t(s, s', a, a')\| \pi_h(a_t|s_t)\, d\mathbf{a}_t d\mathbf{s}_t\\ &\leq \int L_{ps}\|s - s'\| + L_{pa}\|a - a'\|\, ds_1\\ &\leq |\mathcal{S}| \left(L_{ps}\|s - s'\| + L_{pa}\|a - a'\|\right),\end{aligned} \tag{110}$$

where $|\mathcal{S}|$ is the measure of the set $\mathcal{S}$. Then, one can further upper bound (109) by

$$\begin{aligned}\|Q'(s, a) - Q'(s', a')\| \leq L_{rs}\|s - s'\| + L_{ra}\|a - a'\|\\ + \sum_{t=1}^{\infty} \gamma^t B_r |\mathcal{S}| \left(L_{ps}\|s - s'\| + L_{pa}\|a - a'\|\right).\end{aligned} \tag{111}$$

Because the sum of the geometric yields $\gamma/(1-\gamma)$, it follows that $Q'(s, a)$ satisfies

$$\|Q'(s, a) - Q'(s', a')\| \leq L_{Qs}\|s - s'\| + L_{Qa}\|a - a'\|, \tag{112}$$

with $L_{Qs} = L_{rs} + \frac{\gamma B_r}{1-\gamma} L_{ps}|\mathcal{S}|$ and $L_{Qa} = L_{ra} + \frac{\gamma B_r}{1-\gamma} L_{pa}|\mathcal{S}|$. We next show that h is Lipschitz. Using the reproducing property of the kernel, we can write the difference between $h(s)$ and $h(s')$ as

$$h(s) - h(s') = \langle h, \kappa(s, \cdot) - \kappa(s', \cdot)\rangle. \tag{113}$$

Using the Cauchy-Schwartz inequality we can upper bound the previous inner product by

$$\|h(s) - h(s')\| \leq \|h\| \sqrt{\kappa(s, s) + \kappa(s', s') - 2\kappa(s, s')} \tag{114}$$

Let us define $f(s, s') = \sqrt{\kappa(s, s) + \kappa(s', s') - 2\kappa(s, s')}$ and show that it is Lipschitz. To do so, use the following change of variables $u = \Sigma_{\mathcal{H}}^{-1/2}(s - s')$ and write $f(u)$ as follows

$$f(u) = \sqrt{2}\sqrt{1 - e^{-\|u\|^2/2}}. \tag{115}$$

Then, the gradient of $f(u)$ yields

$$\nabla f(u) = \frac{1}{\sqrt{2}} \frac{e^{-\|u\|^2/2} u}{\sqrt{1 - e^{-\|u\|^2/2}}}. \tag{116}$$

Notice that the only point where the function might not be bounded is when $u = 0$, since the limit of the denominator is zero. To show that this is not the case, observe that a second order Taylor approximation of that term yields

$$1 - e^{-\|u\|^2/2} = \frac{1}{2}\|u\|^2 + o(\|u\|^2), \tag{117}$$

where $o(\|u\|^2)$ is a function such that $\lim_{\|u\|\to 0} o(\|u\|^2)/\|u\|^2 = 0$. Thus we have that

$$\lim_{\|u\|\to 0} \|\nabla f(u)\| = \lim_{\|u\|\to 0} \frac{e^{-\|u\|^2/2}\|u\|}{\|u\|} = 1 \tag{118}$$

It can be shown that the gradient of $\|\nabla f(u)\|$ is always differentiable except at $u = 0$ and it never attains the value zero except for at the limit when $\|u\| \to \infty$. This means, that there are no critical points of $\|\nabla f(u)\|$ except at infinity. On the other hand it follows from (116) that $\lim_{\|u\|\to\infty} \|\nabla f(u)\| = 0$, so, the critical point at infinity is a minimum. Thus, the maximum norm of $\nabla f(u)$ is attained at $u = 0$ and it takes the value 1. Thus $f(u)$ is Lipschitz with constant 1. Use the fact that $f(0) = 0$ to bound

$$\begin{aligned}|f(u)| \leq \|u\| = \left\|\Sigma_{\mathcal{H}}^{-1/2}(s - s')\right\|\\ \leq \lambda_{\min}(\Sigma_{\mathcal{H}})^{-1/2}\|s - s'\|.\end{aligned} \tag{119}$$

The latter shows that $h(s)$ is Lipschitz with constant $L_h := \|h\| \lambda_{\min}(\Sigma_{\mathcal{H}})^{-1/2}$. We next use this result to complete the proof that $Q'(s)$ is Lipschitz. From its definition (cf., (102)) and the fact that its Lipschitz (cf., (112)) we have that

$$\|Q'(s) - Q'(s')\| \leq L_{Q_s}\|s - s'\| + L_{Q_a}\|h(s) - h(s')\|. \tag{120}$$

Because $h(s)$ is Lipschitz it follows that

$$\|Q'(s) - Q'(s')\| \leq (L_{Q_s} + L_{Q_a} L_h)\|s - s'\|. \tag{121}$$

This completes the proof of the first claim of the proposition. To show that $D(s)$ is bounded consider its norm. Using its expression in (98) it is possible to uper bound it as

$$\|D(s)\| \leq \mathbb{E}\left\|Q\left(s, h(s) + \Sigma^{1/2}\zeta\right)\zeta\right\|. \tag{122}$$

Since $|Q(s, a)| < B_r/(1-\gamma)$ (cf., Lemma 3 [14]) it follows that

$$\|D(s)\| \leq \frac{B_r}{1-\gamma}\mathbb{E}\|\zeta\| = \frac{\sqrt{2}B_r}{1-\gamma}\frac{\Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2}\right)}, \tag{123}$$

where Γ is the Gamma function. ■

To complete the proof of Lemma 2 observe that we can write the difference

$$\begin{aligned}\|D(s)\rho_{s_0}(s) - D(s')\rho_{s_0}(s')\| \leq \|D(s)\rho_{s_0}(s) - D(s)\rho_{s_0}(s')\|\\ + \|D(s)\rho_{s_0}(s') - D(s')\rho_{s_0}(s')\|.\end{aligned} \tag{124}$$

Using the Lipschitz continuity and the boundedness of both $D(s)$ and $\rho_{s_0}(s)$ it follows that

$$\begin{aligned}\|D(s)\rho_{s_0}(s) - D(s')\rho_{s_0}(s')\| \leq B_D L_p \|s - s'\| \\ +B_\rho L_D \|s' - s\| .\end{aligned} \tag{125}$$

This completes the proof of the lemma.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[2] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[3] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.

[4] R. S. Sutton, H. R. Maei, and C. Szepesvári, "A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation," in *Advances in neural information processing systems*, pp. 1609–1616, 2009.

[5] S. Bhatnagar, D. Precup, D. Silver, R. S. Sutton, H. R. Maei, and C. Szepesvári, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *Advances in Neural Information Processing Systems*, pp. 1204–1212, 2009.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[7] A. Koppel, G. Warnell, E. Stump, P. Stone, and A. Ribeiro, "Breaking bellman's curse of dimensionality: Efficient kernel gradient temporal difference," *arXiv preprint arXiv:1709.04221*, 2017.

[8] E. Tolstaya, A. Koppel, E. Stump, and A. Ribeiro, "Nonparametric stochastic compositional gradient descent for q-learning in continuous markov decision problems,"

[9] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*, pp. 30–37, Elsevier, 1995.

[10] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[11] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Adv. in neural information proc. sys.*, pp. 1057–1063, 2000.

[12] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[13] G. Lever and R. Stafford, "Modelling policies in mdps in reproducing kernel hilbert space," in *A. I. and Statistics*, pp. 590–598, 2015.

[14] S. Paternain, J. A. Bazerque, A. Small, and A. Ribeiro, "Stochastic policy gradient ascent in reproducing kernel hilbert spaces," *Transactions on Automatic Control*, vol. 66, p. (To appear), 8 2021.

[15] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[16] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, 2001.

[17] D. P. Bertsekas, *Nonlinear programming*. Athena Sci., Belmont, 1999.

[18] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, pp. 1008–1014, 2000.

[19] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor–critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[20] T. Degris, M. White, and R. S. Sutton, "Off-policy actor-critic," *arXiv preprint arXiv:1205.4839*, 2012.

[21] S. Paternain, J. A. Bazerque, A. Small, and A. Ribeiro, "Policy improvement directions for reinforcement learning in reproducing kernel hilbert spaces," in *IEEE 58th Conference on Decision and Control (CDC)*, pp. pp. 7454–7461, 2019.

[22] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *arXiv preprint arXiv:1612.04111*, 2016.

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 2. MIT press Cambridge, second ed., 2018.

[24] M. Muresan and M. Muresan, *A concrete approach to classical analysis*, vol. 14. Springer, 2009.

[25] R. Durrett, *Probability: Theory and Examples*. Cambridge University Press, 2010.

[26] D. P. Bertsekas and J. N. Tsitsiklis., *Neuro-dynamic programming*, vol. 5. Athena Scientific, MA, 1996.

[27] R. Pemantle, "Nonconvergence to unstable points in urn models and stochastic approximations," *The Annals of Prob.*, pp. 698–712, 1990.

[28] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1, pp. 165–187, 2002.

[29] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[30] B. J Mercer, "Xvi. functions of positive and negative type, and their connection the theory of integral equations," *Phil. Trans. R. Soc. Lond. A*, vol. 209, no. 441-458, pp. 415–446, 1909.

Santiago Paternain received the B.Sc. degree in electrical engineering from Universidad de la República Oriental del Uruguay, Montevideo, Uruguay in 2012, the M.Sc. in Statistics from the Wharton School in 2018 and the Ph.D. in Electrical and Systems Engineering from the Department of Electrical and Systems Engineering, the University of Pennsylvania in 2018. He is currently an Assistant Professor in the Department of Electrical Computer and Systems Engineering at the Rensselear Polytechnic Institute. Prior to joining Rensselear, Dr. Paternain was a postdoctoral Researcher at the University of Pennsylvania. His research interests include optimization and control of dynamical systems. Dr. Paternain was the recipient of the 2017 CDC Best Student Paper Award and the 2019 Joseph and Rosaline Wolfe Best Doctoral Dissertation Award from the Electrical and Systems Engineering Department at the University of Pennsylvania.

Juan Andrés Bazerque received the B.Sc. degree in electrical engineering from Universidad de la República (UdelaR), Montevideo, Uruguay, in 2003, and the M.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Minnesota (UofM), Minneapolis, in 2010 and 1013 respectively. Since 2015 he is an Assistant Professor with the Department of Electrical Engineering at UdelaR. 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-reciepient of the best paper award at the 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communication 2007.

Alejandro Ribeiro received the B.Sc. degree in electrical engineering from the Universidad de la República Oriental del Uruguay, Montevideo, in 1998 and the M.Sc. and Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis in 2005 and 2007. From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. After his M.Sc. and Ph.D studies, in 2008 he joined the University of Pennsylvania (Penn), Philadelphia, where he is currently the Rosenbluth Associate Professor at the Department of Electrical and Systems Engineering. His research interests are in the applications of statistical signal processing to the study of networks and networked phenomena. His focus is on structured representations of networked data structures, graph signal processing, network optimization, robot teams, and networked control. Dr. Ribeiro received the 2014 O. Hugo Schuck best paper award, and paper awards at the 2016 SSP Workshop, 2016 SAM Workshop, 2015 Asilomar SSC Conference, ACC 2013, ICASSP 2006, and ICASSP 2005. His teaching has been recognized with the 2017 Lindback award for distinguished teaching and the 2012 S. Reid Warren, Jr. Award presented by Penn's undergraduate student body for outstanding teaching. Dr. Ribeiro is a Fulbright scholar class of 2003 and a Penn Fellow class of 2015.