An Actor-Critic Algorithm With Second-Order Actor and Critic

Jing Wang and Ioannis Ch. Paschalidis, *Fellow, IEEE*

***Abstract*—Actor-critic algorithms solve dynamic decision making problems by optimizing a performance metric of interest over a user-specified parametric class of policies. They employ a combination of an actor, making policy improvement steps, and a critic, computing policy improvement directions. Many existing algorithms use a steepest ascent method to improve the policy, which is known to suffer from slow convergence for ill-conditioned problems. In this paper, we first develop an estimate of the (Hessian) matrix containing the second derivatives of the performance metric with respect to policy parameters. Using this estimate, we introduce a new second-order policy improvement method and couple it with a critic using a second-order learning method. We establish almost sure convergence of the new method to a neighborhood of a policy parameter stationary point. We compare the new algorithm with some existing algorithms in two applicationSand demonstrate that it leads to significantly faster convergence.**

***Index Terms*—Actor-critic algorithms, Markov decision processes, Newton's method, robotics.**

I. Introduction

MARKOV Decision Processes (MDPs) provide a general framework for sequential decision making problems. Although MDPs can be solved using *dynamic programming*, the well-known "curse of dimensionality" becomes an impediment for larger instances [1]. In addition, *dynamic programming* in a standard implementation requires explicit transition probabilities among states under each control, which are not available for many applications. To address these limitations, a number of *approximate dynamic programming* techniques have been developed, including *reinforcement learning* methods [2], a variety of techniques involving value function and policy approximations (*neuro-dynamic programming* [3]) and *actor-critic algorithms* [4].

Manuscript received April 16, 2016; accepted September 23, 2016. Date of publication; date of current version. This paper appeared in part at the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, 2014. This work was supported in part by the NSF under grants CNS-1239021, CCF-1527292, and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952. Recommended by Associate Editor Q.-S. Jia.

J. Wang is with the Center for Information & Systems Engineering, Boston University (e-mail: wangjing@bu.edu).

I. Ch. Paschalidis is with the Department of Electrical and Computer Engineering and Division of Systems Engineering, Boston University8 St. Mary's St., Boston, MA 02215, (e-mail: yannisp@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2016.2616384

This paper focuses on the latter *actor-critic algorithms*. They optimize a parametric user-designed *Randomized Stationary Policy* (RSP) using policy gradient estimation. RSPs are policies parameterized by a parsimonious set of parameters. To optimize the RSPs with respect to these parameters, *actor-critic algorithms* estimate policy gradients using learning methods that are much more efficient than computing a cost-to-go function over the entire state-action space. Many different variants of *actor-critic algorithms* have been proposed and shown to be effective for many applications such as robotics [5], biology [6], navigation [7], and optimal bidding for electricity generation [8].

In an attractive type of an *actor-critic algorithm* introduced in [4], a critic is used to estimate the policy gradient from observations on a single sample path and an actor is using this gradient to update the policy at a slower time-scale [4]. The estimate of the critic tracks the slowly-varying policy asymptotically, using first-order variants of the *Temporal Difference (TD)* learning algorithms (TD(1) and TD(λ)). However, it has been shown that second-order learning methods—Least Squares TD (LSTD)—are superior in terms of *rate of convergence* (see [9]–[14]). LSTD was first proposed for discounted cost problems in [11] and was shown to have the optimal *rate of convergence* in [12]. In [14], LSTD is used in the critic of an actor-critic algorithm, resulting in the LSTD Actor-Critic algorithm (LSTD-AC). Later, this algorithm was applied to applications of robot motion control with temporal specifications [15]–[17]. Despite faster convergence than TD-based methods, LSTD-AC exhibits slow convergence for ill-conditioned problems in which the performance metric is more sensitive to some parameters in the RSPs than others. The reason is that it uses a first order actor with an "unscaled" gradient, commonly known as steepest ascent, to update the policy. This often leads to a "zig-zagging" behavior in order to converge to a stationary point.

Several algorithms have been introduced which use a second-order method in the actor. The "natural" gradient method was originally proposed for stochastic learning [18], [19]. [20] proposed a different estimate of the natural gradient but its accuracy can be influenced by the choice of basis functions; an episodic algorithm was then proposed to guarantee the unbiasedness of the estimate. These methods use the inverse of the Fisher information matrix to scale the gradient. [21] suggested several incremental methods using the natural policy gradient. [22] presented an online natural actor-critic algorithm using a natural gradient and applied it to a road traffic optimization problem. Based on [20], [23] proposes three fully incremental natural actor-critic algorithms. It also describes a method that is based on a "vanilla" gradient and provides extensive empirical comparison of all algorithms in test problems (so called *Generic Average Reward Non-stationary Environment Testbed—GARNET* problems [23]).

Although natural gradients are very effective in stochastic learning, there are alternative ways to scale gradients. The Hessian matrix of the performance metric with respect to the parameters is commonly used to improve the *rate of convergence*. [24] proposes an estimate of the Hessian matrix for a discounted reward problem using a sample path of an MDP. Although the relationship between the Fisher information matrix and the Hessian matrix has been briefly discussed in [19] and [25], it is still not fully clear how they are related in the actor-critic framework and why natural actor-critic algorithms work well in practice.

In this work, we develop a more general estimate of the Hessian matrix for actor-critic algorithms. In Section V-C, we demonstrate that our Hessian estimate degenerates to the Fisher information matrix used in natural actor-critic algorithms if we assume no knowledge of the state-action value function and ignore second derivatives with respect to the parameter vector. In this light, natural actor-critic algorithms can be seen as equivalent to quasi-Newton methods that assume no knowledge of the state-action value function when approximating the Hessian matrix. In fact, [12] proposes a quasi-Newton actor-critic algorithm that is very similar to the methods in [20].

This paper proposes a method that uses LSTD-based critics to provide estimates of both the gradient and the Hessian and utilizes the Hessian estimate in the *actor* to update policy parameters.

We establish *almost sure* convergence in the neighborhood of a stationary point (with respect to policy parameters) of the performance metric. We remark that a subset of the results appeared in a preliminary conference paper in [1]. The present paper contains all proofs concerning the Hessian estimate, the convergence analysis which was absent from [1], and a much more extensive numerical evaluation of our method both in GARNET problems and in an application from robotics.

The remainder of the paper is organized as follows: Section II provides background on MDPs and establishes some of our notation. Section III presents the estimation of the policy gradient. Section IV develops the estimate of the policy Hessian, which is the foundation of the new algorithm. Section V describes our method and Section VI proves its convergence. Section VII presents two case studies.

Notation: Bold letters are used to denote vectors and matrices; typically vectors are lower case and matrices upper case. Vectors are column vectors, unless explicitly stated otherwise. Prime denotes transpose. For the column vector $\mathbf{x} \in \mathbb{R}^n$ we write $\mathbf{x} = (x_1, \ldots, x_n)$ for economy of space, while $\|\mathbf{x}\|$ denotes the Euclidean norm. The expressions $\succ 0$ and $\succeq 0$ denote positive-definiteness and positive-semi-definiteness, respectively. Vectors or matrices with all zeroes are written as $\mathbf{0}$ and the identity matrix as $\mathbf{I}$. For any set $\mathscr{S}$, $|\mathscr{S}|$ denotes its cardinality. $\boldsymbol{\theta}$ denotes the parameters in parameterized policies. If not explicitly specified, ∇ and ∇^2 denote the gradient and Hessian w.r.t. $\boldsymbol{\theta}$. To simplify the notation, a lot of equations in this paper are represented using functional notation and the domain of these functions is assumed to be $\mathbb{X} \times \mathbb{U}$, where $\mathbb{X}$ and $\mathbb{U}$ are the state and the action space, respectively, of the MDP. Vector-valued functions are denoted using bold letters while scalar-valued functions are denoted using normal letters. $\underline{0}$ and $\underline{1}$ are functions that assign the value 0 and 1 to all state-action pairs, respectively.

II. Markov Decision Processes

Consider a discrete-time *Markov Decision Process (MDP)* with a finite state space $\mathbb{X}$ and an action space $\mathbb{U}$. Let $\mathbf{x}_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$ be the state of the system and the action taken at time k, respectively. Let $g(\mathbf{x}_k, u_k)$ be the one-step reward of applying action u_k when the system is at state $\mathbf{x}_k$. We will use $\mathbf{x}_0$ to denote the initial state and $p(\mathbf{x}_{k+1}|\mathbf{x}_k, u_k)$ for the state transition probabilities, which are typically not explicitly known. We assume that $\{\mathbf{x}_k\}$ and $\{\mathbf{x}_k, u_k\}$ are ergodic Markov chains [12].

This paper considers policies that belong to a parameterized family of RSPs $\{\mu_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^n\}$. That is, given a state $\mathbf{x} \in \mathbb{X}$ and an n-dimensional parameter vector $\boldsymbol{\theta}$, the policy applies action $u \in \mathbb{U}$ with probability $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$. Given a fixed policy $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$, the history of $g(\mathbf{x}_k, u_k)$ can be represented by a random process. Let $E_{\boldsymbol{\theta}}\{\cdot\}$ be the expectation with respect to this random process; the long-term average reward for a policy $\mu_{\boldsymbol{\theta}}$ is $\bar{\alpha}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\{\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} [g(\mathbf{x}_k, u_k)]\}$.

In average reward MDP optimization problems, the performance metric is the long-term average reward $\bar{\alpha}(\boldsymbol{\theta})$ and the objective is to optimize $\bar{\alpha}(\boldsymbol{\theta})$. Similar problems can be defined by using discounted reward or total reward as performance metrics [12]. Note that the discounted reward and the total reward can be treated as the average reward of an artificial MDP (See Chapter 2 of [12]). Without loss of generality, this paper focuses on the average reward case. Corresponding results for the other cases can be obtained with modifications similar to Sec. 2.4 and 2.5 of [12].

III. Estimation of Policy Gradient

The *state-action value function* $Q_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ (sometimes referred to as the Q-value function) of a policy $\mu_{\boldsymbol{\theta}}$ is defined as the expected future reward given the current state $\mathbf{x}$ and the action u. $Q_{\boldsymbol{\theta}}$ is the unique solution of the Poisson equation with parameter $\boldsymbol{\theta}$ [26], [12] (written as a functional relationship)

$$Q_{\boldsymbol{\theta}} = g - \bar{\alpha}(\boldsymbol{\theta})\underline{1} + P_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}, \tag{1}$$

where $P_{\boldsymbol{\theta}}$ is the operator of taking expectation after one transition. More precisely, for any real-valued or vector-valued function f defined on $\mathbb{X} \times \mathbb{U}$,

$$(P_{\boldsymbol{\theta}} f)(\mathbf{x}, u) = \sum_{\mathbf{y},\nu} p(\mathbf{y}|\mathbf{x}, u)\mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) f(\mathbf{y}, \nu) \tag{2}$$

for all $(\mathbf{x}, u) \in \mathbb{X} \times \mathbb{U}$.

Let now

$$\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x}), \tag{3}$$

where $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \mathbf{0}$ when $\mathbf{x}, u$ are such that $\mu_{\boldsymbol{\theta}}(u|\mathbf{x}) \equiv 0$ for all $\boldsymbol{\theta}$'s. It is assumed that $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is bounded and continuously differentiable. Since $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$ is the probability of action u at state $\mathbf{x}$ for $\boldsymbol{\theta}$, $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is the gradient of the *log-likelihood* $\ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})$. We write $\boldsymbol{\psi}_{\boldsymbol{\theta}} = (\psi^1_{\boldsymbol{\theta}}, \dots, \psi^n_{\boldsymbol{\theta}})$ where n is the dimensionality of $\boldsymbol{\theta}$.

For each $\boldsymbol{\theta} \in \mathbb{R}^n$, let $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u)$ be the stationary probability of state-action pair $(\mathbf{x}, u)$ in the Markov chain $\{\mathbf{x}_k, u_k\}$. For any $\boldsymbol{\theta} \in \mathbb{R}^n$, we define the inner product operator $\langle \cdot, \cdot \rangle_{\boldsymbol{\theta}}$ of two real-valued or vector-valued functions Q_1, Q_2 on $\mathbb{X} \times \mathbb{U}$ by

$$\langle Q_1, Q_2 \rangle_{\boldsymbol{\theta}} = \sum_{\mathbf{x}, u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) Q_1(\mathbf{x}, u) Q_2(\mathbf{x}, u). \tag{4}$$

A key fact underlying actor-critic algorithms is that the policy gradient of $\bar{\alpha}(\boldsymbol{\theta})$ can be expressed as [27], [12]

$$\frac{\partial \bar{\alpha}(\boldsymbol{\theta})}{\partial \theta_i} = \left\langle Q_{\boldsymbol{\theta}}, \psi^i_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}, \; i = 1, \dots, n. \tag{5}$$

IV. Estimation of the Policy Hessian

Earlier work in actor-critic methods has used critics based on TD(1), TD(λ), and LSTD methods to estimate the policy gradient $\nabla \bar{\alpha}(\boldsymbol{\theta})$ [4], [28]. Since we are interested in a Newton-like gradient ascent update in the actor, in this section we develop an estimate for the policy Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$.

Applying the operator ∇ on the real-valued function $g_{\boldsymbol{\theta}}(\mathbf{x}, u)$ parameterized by $\boldsymbol{\theta}$, we obtain a vector-valued function, abbreviated as $\nabla g_{\boldsymbol{\theta}}$, which maps $(\mathbf{x}, u)$ to $\nabla g_{\boldsymbol{\theta}}(\mathbf{x}, u)$. For a vector-valued function $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^m$ parameterized by $\boldsymbol{\theta}$, which can be denoted as $\mathbf{f}_{\boldsymbol{\theta}} = (f^1_{\boldsymbol{\theta}}, \dots, f^m_{\boldsymbol{\theta}})$, we define $\nabla \mathbf{f}_{\boldsymbol{\theta}}$ to be an $n \times m$ matrix-valued function whose ith column is $\nabla f^i_{\boldsymbol{\theta}}$.

Lemma IV.1: For any vector-valued function $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^m$, we have

$$\nabla \left(P_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}} \right) = P_{\boldsymbol{\theta}} \left(\nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right).$$

Proof: For all state-action pairs $(\mathbf{x}, u) \in \mathbb{X} \times \mathbb{U}$, we have

$$\begin{aligned} \nabla (P_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}})(\mathbf{x}, u) = {} & \nabla \left(\sum_{\mathbf{y}, \nu} p(\mathbf{y}|\mathbf{x}, u) \mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \nu) \right) \\ = {} & \sum_{\mathbf{y}, \nu} p(\mathbf{y}|\mathbf{x}, u) \nabla \left(\mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \nu) \right). \end{aligned} \tag{6}$$

In the above, $\mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \nu)$ is a function defined on $\mathbb{X} \times \mathbb{U}$, which is abbreviated as $\mu_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}}$. Using the chain rule and the definition of $\boldsymbol{\psi}_{\boldsymbol{\theta}}$, we obtain

$$\begin{aligned} \nabla \left(\mu_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}} \right) &= \mu_{\boldsymbol{\theta}} \nabla \mathbf{f}_{\boldsymbol{\theta}} + \nabla \mu_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \\ &= \mu_{\boldsymbol{\theta}} \left(\nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right). \end{aligned} \tag{7}$$

The lemma can be proved by substituting (7) to (6). ■

Lemma IV.1 provides a way to interchange the $P_{\boldsymbol{\theta}}$ and ∇ operators. Similar to the definition of $\boldsymbol{\psi}_{\boldsymbol{\theta}}$, we define

$$\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \nabla^2 \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x}), \tag{8}$$

where $\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \mathbf{0}$ when $\mathbf{x}, u$ are such that $\mu_{\boldsymbol{\theta}}(u|\mathbf{x}) \equiv 0$ for all $\boldsymbol{\theta}$. $\boldsymbol{\varphi}_{\boldsymbol{\theta}}$ is the Hessian matrix of the *log-likelihood* $\ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})$.

The following theorem establishes a similar result to (5) for the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$.

Theorem IV.2 (Hessian Matrix of Average Reward): Let $\varphi^{ij}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ be the scalar-valued (i, j)-th component of $\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$. The second-order partial derivative of $\bar{\alpha}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ can be represented as:

$$\begin{aligned} \frac{\partial^2 \bar{\alpha}(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} = {} & \left\langle Q_{\boldsymbol{\theta}}, \psi^i_{\boldsymbol{\theta}} \psi^j_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle Q_{\boldsymbol{\theta}}, \varphi^{ij}_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \\ & + \left\langle \frac{\partial Q_{\boldsymbol{\theta}}}{\partial \theta_i}, \psi^j_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle \frac{\partial Q_{\boldsymbol{\theta}}}{\partial \theta_j}, \psi^i_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \end{aligned} \tag{9}$$

for all $i, j = 1, \dots, n$, where $\langle \cdot, \cdot \rangle_{\boldsymbol{\theta}}$ is the inner product operator defined in (4).

Proof: Applying the ∇ operator on both sides of (1) and using Lemma IV.1 with $\mathbf{f}_{\boldsymbol{\theta}}$ being the scalar function $Q_{\boldsymbol{\theta}}$, we obtain

$$\nabla \bar{\alpha}(\boldsymbol{\theta}) \underline{1} + \nabla Q_{\boldsymbol{\theta}} = P_{\boldsymbol{\theta}} \left(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + \nabla Q_{\boldsymbol{\theta}} \right). \tag{10}$$

Defining the vector-valued function $\mathbf{f}_{\boldsymbol{\theta}} = \boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + \nabla Q_{\boldsymbol{\theta}}$ and applying again the ∇ operator on both sides of (10), we have

$$\nabla (\nabla \bar{\alpha}(\boldsymbol{\theta}) \underline{1} + \nabla Q_{\boldsymbol{\theta}}) = \nabla (P_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}}),$$

which due to Lemma IV.1 implies

$$\nabla^2 \bar{\alpha}(\boldsymbol{\theta}) \underline{1} + \nabla^2 Q_{\boldsymbol{\theta}} = P_{\boldsymbol{\theta}} \left(\nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right). \tag{11}$$

Take now the inner product with $\underline{1}$ on both sides of (11) and notice that because $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is the stationary probability under θ, it holds $\langle \underline{1}, \mathbf{h} \rangle_{\boldsymbol{\theta}} = \langle \underline{1}, P_{\boldsymbol{\theta}} \mathbf{h} \rangle_{\boldsymbol{\theta}}$ for any function $\mathbf{h}$ defined on $\mathbb{X} \times \mathbb{U}$. We have

$$\nabla^2 \bar{\alpha}(\boldsymbol{\theta}) + \left\langle \underline{1}, \nabla^2 Q_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} = \left\langle \underline{1}, \nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}.$$

Using the definition of $\mathbf{f}_{\boldsymbol{\theta}}$ and the fact $\nabla \mathbf{f}_{\boldsymbol{\theta}} = \nabla (\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}) + \nabla^2 Q_{\boldsymbol{\theta}}$, we obtain

$$\begin{aligned} \nabla^2 \bar{\alpha}(\boldsymbol{\theta}) + \left\langle \underline{1}, \nabla^2 Q_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} = {} & \left\langle \underline{1}, \nabla (\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}) + \nabla^2 Q_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \\ & + \left\langle \underline{1}, Q_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}'_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \nabla Q'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \end{aligned} \tag{12}$$

Applying the chain rule, noticing that $\nabla \boldsymbol{\psi}_{\boldsymbol{\theta}} = \boldsymbol{\varphi}_{\boldsymbol{\theta}}$, and reorganizing the terms in (12) it follows

$$\begin{aligned} \nabla^2 \bar{\alpha}(\boldsymbol{\theta}) = {} & \left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\varphi}_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \\ & + \left\langle \nabla Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \nabla Q'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}. \end{aligned} \tag{13}$$

■

Corresponding results for the discounted reward and the total reward cases can be derived based on the relationship between these three problems we discussed earlier. Intuitively, the discounted and total rewards can be considered as average rewards in some artificial MDPs. More detailed information about constructing the artificial MDPs is available at Sec. 2.4 and Sec. 2.5 of [12].

Theorem IV.2 states that the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$ can be decomposed into four terms, all of which take the form of inner products. The first two terms are the inner products of the state-action value function $Q_{\boldsymbol{\theta}}$ with $\psi^i_{\boldsymbol{\theta}} \psi^j_{\boldsymbol{\theta}}$ and $\varphi^{ij}_{\boldsymbol{\theta}}$. Because of the similarity between the first two terms and (5), we can use similar techniques as in the LSTD-AC to estimate them.

For the last two terms in (13) we need an estimate of $\nabla Q_{\boldsymbol{\theta}}$. Note that (10) is the counterpart of the Poisson equation (1) for $\nabla Q_{\boldsymbol{\theta}}$, where $P_{\boldsymbol{\theta}}(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}})$ plays the role of the one-step reward. However, this equation can not be directly used to estimate $\nabla Q_{\boldsymbol{\theta}}$ because it is quite hard to obtain $P_{\boldsymbol{\theta}}(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}})$. To address this problem, we present the following theorem.

Theorem IV.3: Let the function $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$ be the solution of the equation

$$\nabla \bar{\alpha}(\boldsymbol{\theta})\underline{1} + \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} = \boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \tag{14}$$

and $\nabla Q_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$ be the solution of (10). Then,

$$\left\langle \nabla Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} - \left\langle \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} = - \left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}}. \tag{15}$$

Proof: Applying the $P_{\boldsymbol{\theta}}$ operator on both sides of (14) and using the fact that $P_{\boldsymbol{\theta}}\underline{1} = \underline{1}$, we obtain

$$\nabla \bar{\alpha}(\boldsymbol{\theta})\underline{1} + P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} = P_{\boldsymbol{\theta}}(\psi_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}). \tag{16}$$

Comparing (10) and (16), it follows $P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} = \nabla Q_{\boldsymbol{\theta}}$. As a result,

$$\begin{aligned}
&\left\langle \nabla Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} - \left\langle \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} = \left\langle \nabla Q_{\boldsymbol{\theta}} - \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&= \left\langle P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} - \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&= \left\langle -\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + \nabla \bar{\alpha}(\boldsymbol{\theta})\underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&= - \left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} + \nabla \bar{\alpha}(\boldsymbol{\theta}) \left\langle \underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}},
\end{aligned} \tag{17}$$

where the third equality above used (14).

Let now $\pi_{\boldsymbol{\theta}}(\cdot)$ be the stationary probability of the Markov chain $\{\mathbf{x}_k\}$ under RSP $\boldsymbol{\theta}$. Then, $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u) = \pi_{\boldsymbol{\theta}}(\mathbf{x})\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$, and

$$\begin{aligned}
\left\langle \underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} &= \sum_{\mathbf{x},u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)^{'} \\
&= \sum_{\mathbf{x},u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) \nabla \mu_{\boldsymbol{\theta}}(u|\mathbf{x})^{'} / (\mu_{\boldsymbol{\theta}}(u|\mathbf{x})) \\
&= \sum_{\mathbf{x}} \pi_{\boldsymbol{\theta}}(\mathbf{x}) \sum_{u} \nabla \mu_{\boldsymbol{\theta}}(u|\mathbf{x})^{'} \\
&= \mathbf{0},
\end{aligned} \tag{18}$$

where in the second equality we used (3) and the last equality follows from the fact that $\sum_u \mu_{\boldsymbol{\theta}}(u|\mathbf{x}) = 1$ for all $\boldsymbol{\theta}$. Eq. (15) follows by combining (17) and (18). ■

By symmetry to Eq. (15), it also holds that

$$\left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \nabla Q_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} - \left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} = - \left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}}. \tag{19}$$

Substituting (15) and (19) into (13), we obtain a new estimate of the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$ given in the following Corollary.

Corollary IV.4: With $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ being a solution of (14), the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$ can be expressed as:

$$\begin{aligned}
\nabla^2 \bar{\alpha}(\boldsymbol{\theta}) = &\left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\varphi}_{\boldsymbol{\theta}} - \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} + \left\langle \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&+ \left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}}.
\end{aligned} \tag{20}$$

A. Function Approximation

We can calculate $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ by solving (1) and (14). However, when $\mathbb{X} \times \mathbb{U}$ is very large, the computational cost becomes prohibitive. This problem can be addressed using *function approximation* techniques. One popular type of function approximation is to express $Q_{\boldsymbol{\theta}}$ and each component of $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ with a linear combination of feature functions. We choose a set of feature functions $\boldsymbol{\phi}_{\boldsymbol{\theta}} = (\psi_{\boldsymbol{\theta}}^i, \varphi_{\boldsymbol{\theta}}^{ij}, \psi_{\boldsymbol{\theta}}^i \psi_{\boldsymbol{\theta}}^j;\ i, j = 1, \ldots, n)$, where $\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is an N-dimensional vector for $\forall \mathbf{x}, u \in \mathbb{X} \times \mathbb{U}$ with $N = (2n^2 + n)$ and n being the dimensionality of $\boldsymbol{\theta}$. Similar to other actor-critic algorithms, the basis functions $\boldsymbol{\phi}_{\boldsymbol{\theta}}$ need to be uniformly linearly independent [4], [12], which can be enforced by choosing a suitable structure of policies. Some additional features can be added depending on the particular application. This added flexibility could be useful in a number of ways as it has been discussed in [4].

Similar to [12], we consider the following linear approximation for $Q_{\boldsymbol{\theta}}$

$$Q_{\boldsymbol{\theta}}^{\mathbf{r}}(\mathbf{x}, u) = \boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}(\mathbf{x}, u)\mathbf{r}, \qquad \mathbf{r} \in \mathbb{R}^N. \tag{21}$$

Let us now view the inner product operator in (4) for real-valued functions in $\mathbb{X} \times \mathbb{U}$ as an inner product between vectors in $\mathbb{R}^{|\mathbb{X}||\mathbb{U}|}$ and denote by $\|\cdot\|_{\boldsymbol{\theta}}$ the induced norm. Also denote by $\Phi_{\boldsymbol{\theta}}$ the low-dimensional subspace spanned by $\boldsymbol{\phi}_{\boldsymbol{\theta}}$. If we define

$$\mathbf{r}^* = \arg\min_{\mathbf{r} \in \mathbb{R}^N} ||Q_{\boldsymbol{\theta}}^{\mathbf{r}} - Q_{\boldsymbol{\theta}}||_{\boldsymbol{\theta}}, \tag{22}$$

then $Q_{\boldsymbol{\theta}}^{\mathbf{r}^*}$ is the projection of $Q_{\boldsymbol{\theta}}$ on $\Phi_{\boldsymbol{\theta}}$. Similar to (2.2) of [4],

$$\begin{aligned}
\langle Q_{\boldsymbol{\theta}}^{\mathbf{r}^*}, \psi_{\boldsymbol{\theta}}^i \rangle_{\boldsymbol{\theta}} &= \langle Q_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}}^i \rangle_{\boldsymbol{\theta}}, \\
\left\langle Q_{\boldsymbol{\theta}}^{\mathbf{r}^*}, \varphi_{\boldsymbol{\theta}}^{ij} - \psi_{\boldsymbol{\theta}}^i \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}} &= \left\langle Q_{\boldsymbol{\theta}}, \varphi_{\boldsymbol{\theta}}^{ij} - \psi_{\boldsymbol{\theta}}^i \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}},
\end{aligned} \tag{23}$$

for all $i, j = 1, \ldots, n$.

Define the linear approximation of $\tilde{Q}_{\boldsymbol{\theta}}^i$, the ith component of $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$, as

$$\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i}(\mathbf{x}, u) = \boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}(\mathbf{x}, u)\mathbf{t}^i, \qquad \mathbf{t}^i \in \mathbb{R}^N. \tag{24}$$

Again, for all $i, j = 1, \ldots, n$ and

$$\mathbf{t}^{i*} = \arg\min_{\mathbf{t} \in \mathbb{R}^N} ||\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i} - \tilde{Q}_{\boldsymbol{\theta}}^i||_{\boldsymbol{\theta}}, \tag{25}$$

$\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^{i*}}$ is the projection of $\tilde{Q}_{\boldsymbol{\theta}}^i$ on $\Phi_{\boldsymbol{\theta}}$. Similar to (2.2) of [4], we have

$$\left\langle \tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^{i*}}, \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}} = \left\langle \tilde{Q}_{\boldsymbol{\theta}}^i, \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}}. \tag{26}$$

Equations (23) and (26) state that the projections of $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ on the low-dimensional space $\Phi_{\boldsymbol{\theta}}$ are sufficient for estimating (20). This reduces the computational cost for obtaining $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ since we only have to compute the relative parsimonious vectors $\mathbf{r}^*$ and $\mathbf{t}^{i*}$, $i = 1, \ldots, n$, while it does not alter the inner products needed to compute the gradient $\nabla\bar{\alpha}(\boldsymbol{\theta})$ (cf. (5)) and the Hessian $\nabla^2\bar{\alpha}(\boldsymbol{\theta})$ (cf. (20)).

V. A Second-Order Actor-Critic Algorithm

A. Critic Step

We use the Least Squares Temporal Difference (LSTD) (see, e.g., [14]) with parameter λ to estimate $\mathbf{r}^*$ and $\mathbf{t}^{i*}$, $i = 1\dots,n$, defined in (22) and (25), respectively. Recall that $\mathbf{x}_k$ and u_k denote the state and the action of the system at time k, respectively. Let α_k denote an estimate of the average reward at time k. $\mathbf{z}_k \in \mathbb{R}^N$ denotes Sutton's eligibility trace and $\mathbf{A}_k \in \mathbb{R}^{N\times N}$ a sample estimate of the matrix formed by $\mathbf{z}_k(\boldsymbol{\phi}^{'}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \boldsymbol{\phi}^{'}_{\boldsymbol{\theta}_k}(\mathbf{x}_{k+1},u_{k+1}))$, which can be viewed as a sample observation of the scaled difference of the features between time k and time $k+1$. $\mathbf{b}_k \in \mathbb{R}^N$ refers to a statistical estimate of the single period relative reward with eligibility trace $\mathbf{z}_k$. Let also use the initial values: $\mathbf{A}_0$ is an identity matrix, α_0 is zero, and $\mathbf{b}_0$ and $\mathbf{z}_0$ are column vectors with all zeros. To estimate $\mathbf{r}^*$, we use the following *Q-critic* update

$$
\begin{aligned}
\alpha_{k+1} &= \alpha_k + \gamma_k(g(\mathbf{x}_k,u_k) - \alpha_k), \qquad (27)\\
\mathbf{z}_{k+1} &= \lambda\mathbf{z}_k + \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k),\\
\mathbf{A}_{k+1} &= \mathbf{A}_k + \gamma_k(\mathbf{z}_k\mathbf{w}_k^{'} - \mathbf{A}_k),\\
\mathbf{b}_{k+1} &= \mathbf{b}_k + \gamma_k[(g(\mathbf{x}_k,u_k) - \alpha_k)\mathbf{z}_k - \mathbf{b}_k],
\end{aligned}
$$

where $\mathbf{w}_k = \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_{k+1},u_{k+1})$ and γ_k is a stepsize. Let $\mathbf{r}_k$ be the estimate of $\mathbf{r}^*$ at time k; we set

$$
\mathbf{r}_{k+1} = \begin{cases} \mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1}, & \text{if } \det(\mathbf{A}_{k+1}) \geq \epsilon,\\ \mathbf{r}_k, & \text{otherwise,}\end{cases} \qquad (28)
$$

where ϵ is a small positive constant used to judge whether $\mathbf{A}_{k+1}$ is "ill-conditioned" or not. $\mathbf{A}_k$ should be invertible when k is large enough [29], [30]. Our *Q-critic* (27) is the same with the critic update of the LSTD-AC algorithm in [14] and (28) estimates the same $\mathbf{r}^*$. In addition, we add another critic, named as $\tilde{\mathbf{Q}}$-critic, to estimate $\mathbf{t}^{i*}, \forall i$.

Let now $\mathbf{v}_0^i$, $i = 1,\dots,n$, be a column vector with all zeros. Let also η_0^i, $i = 1,\dots,n$, be a scalar set to zero. Notice the relationship between Eq. (1) for the Q-function and Eq. (14) for the $\tilde{\mathbf{Q}}$-function. To estimate $\mathbf{t}^{i*}$, $i = 1,\dots,n$, defined in (25), we use the following LSTD $\tilde{\mathbf{Q}}$-*critic* update

$$
\begin{aligned}
\eta_{k+1}^i &= \eta_k^i + \zeta_k(q_k^i - \eta_k^i), \qquad i = 1,\dots,n, \qquad (29)\\
\mathbf{v}_{k+1}^i &= \mathbf{v}_k^i + \zeta_k[(q_k^i - \eta_k^i)\mathbf{z}_k - \mathbf{v}_k^i], \qquad i = 1,\dots,n,
\end{aligned}
$$

where $q_k^i = \Gamma(\mathbf{r}_k)\mathbf{r}_k^{'}\boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k)\psi^i_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k)$ is an estimate of the ith component of $\boldsymbol{\psi}_{\boldsymbol{\theta}}Q_{\boldsymbol{\theta}}$ which plays the role of the one-step reward in (14). ζ_k is the stepsize of the $\tilde{\mathbf{Q}}$-*critic* and $\Gamma(\mathbf{r}_k)$ is a function that restricts the influence of the error in the estimate $\mathbf{r}_k$. Let $\mathbf{t}_k^i$ be the estimate of $\mathbf{t}^{i*}$ at time k. Similar to the *Q-critic*, we set

$$
\mathbf{t}_{k+1}^i = \begin{cases} \mathbf{A}_{k+1}^{-1}\mathbf{v}_{k+1}^i, & \text{if } \det(\mathbf{A}_{k+1}) \geq \epsilon,\\ \mathbf{t}_k^i, & \text{otherwise,}\end{cases} \qquad (30)
$$

for $i = 1,\dots,n$. Note that the Sherman-Morrison update of a matrix inverse [22] and the matrix determinant lemma [31] can be applied to reduce the computational cost of calculating $\mathbf{A}_{k+1}^{-1}$ and $\det(\mathbf{A}_{k+1})$ in (28) and (30).

B. Actor Step

Let $Q_{\boldsymbol{\theta}}^{\mathbf{r}}(\mathbf{x},u) = \Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x},u)$ and $\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i} = \Gamma(\mathbf{t}^i)\mathbf{t}^{i'}\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x},u)$ be our estimates for $Q_{\boldsymbol{\theta}}$ and $\tilde{Q}_{\boldsymbol{\theta}}^i$ given $\mathbf{r}$ and $\mathbf{t}^i$, $i = 1,\dots,n$. As mentioned above, the function $\Gamma(\cdot)$ restricts the influence of the error in $\mathbf{r}$ and $\mathbf{t}^i$, respectively (cf. (21) and (24)). For convenience of notation, let $\mathbf{T} = (\mathbf{t}^1,\dots,\mathbf{t}^n)$ and denote by $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{\mathbf{T}} = (\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^1},\dots,\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^n})$ a vector-valued function mapping $\mathbb{X}\times\mathbb{U}$ onto $\mathbb{R}^n$ with ith element equal to $\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i}$. Motivated by (20) and using just a single sample to estimate the expectation (in a standard stochastic approximation fashion), we also define $\hat{\mathbf{U}}_{\boldsymbol{\theta},\mathbf{r},\mathbf{T}}$ to be an $n\times n$ matrix-valued function defined on $\mathbb{X}\times\mathbb{U}$ and parameterized by $(\boldsymbol{\theta},\mathbf{r},\mathbf{T})$ as follows

$$
\hat{\mathbf{U}}_{\boldsymbol{\theta},\mathbf{r},\mathbf{T}} = Q_{\boldsymbol{\theta}}^{\mathbf{r}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}} - \boldsymbol{\psi}_{\boldsymbol{\theta}}\boldsymbol{\psi}_{\boldsymbol{\theta}}^{'}) + \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{\mathbf{T}}\boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} + \boldsymbol{\psi}_{\boldsymbol{\theta}}(\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{\mathbf{T}})^{'}. \qquad (31)
$$

Let $\mathbf{H}_k$ be the estimate of $-\nabla^2\bar{\alpha}(\boldsymbol{\theta})$ at time k with initial condition $\mathbf{H}_0 = \mathbf{I}$. The update rule for $\mathbf{H}_k$ is:

$$
\mathbf{H}_{k+1} = \begin{cases} \mathbf{H}_k + \mathbf{U}_k, & \text{if } \mathbf{U}_k \succ 0,\\ \mathbf{H}_k, & \text{otherwise,}\end{cases} \qquad (32)
$$

where $\mathbf{U}_k = -\hat{\mathbf{U}}_{\boldsymbol{\theta}_k,\mathbf{r}_k,\mathbf{T}_k}(\mathbf{x}_k,u_k)$. Note that $\mathbf{H}_k \succ 0$ because it is updated only when $\mathbf{U}_k \succ 0$. Let χ_k be the number of times the top branch in (32) is executed by iteration k and define

$$
\hat{\mathbf{H}}_k = \begin{cases} \mathbf{I}, & \text{if } \chi_k < \chi_{\min},\\ \mathbf{H}_k, & \text{otherwise,}\end{cases} \qquad (33)
$$

which will be used to avoid a noisy estimate in the initial updates. The actor update takes the form:

$$
\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \beta_k\Gamma(\mathbf{r}_k)\mathbf{r}_k^{'}\boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k)\hat{\mathbf{H}}_k^{-1}\boldsymbol{\psi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k), \qquad (34)
$$

where β_k is a stepsize.

In the update (32), we make sure that our scaling matrix is always positive definite. Notice that $\mathbf{H}_k$ is the estimate of the negative Hessian matrix because we are dealing with a maximization problem. In particular, the Hessian matrix will generally be negative definite in the vicinity of a local maximum and we expect that the upper branch of the update (32) will be used as we approach such a point. The iteration (34) takes a scaled gradient ascent step, with the scaling matrix being positive definite.

The sequences $\{\gamma_k\}$ and $\{\zeta_k\}$ correspond to the stepsizes used by the critics, while β_k and $\Gamma(\mathbf{r}_k)$ control the stepsize for the actor. The function $\Gamma(\mathbf{r}_k)$ is selected such that for some positive constants $C_1 < C_2$:

$$
\begin{aligned}
&\|\mathbf{r}\|\Gamma(\mathbf{r}) \in [C_1, C_2], \qquad \forall \mathbf{r}\in\mathbb{R}^N, \qquad (35)\\
&\|\Gamma(\mathbf{r}) - \Gamma(\hat{\mathbf{r}})\| \leq \frac{C_2\|\mathbf{r}-\hat{\mathbf{r}}\|}{1+\|\mathbf{r}\|+\|\hat{\mathbf{r}}\|}, \qquad \forall \mathbf{r},\hat{\mathbf{r}}\in\mathbb{R}^N.
\end{aligned}
$$

An example that satisfies these requirements is $\Gamma(\mathbf{r}) = \min(1, D/\|\mathbf{r}\|)$ for some positive constant D.

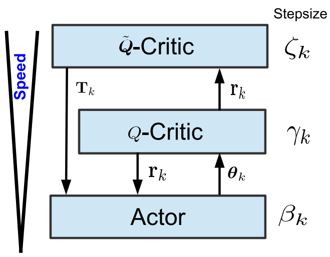

Fig. 1. Relationships between the critics and the actor.

We say a stepsize sequence $\{f_k\}$ is *Square Summable but Not Summable (SSNS)* if $f_k > 0$, $\sum_{k=0}^{\infty} f_k^2 < \infty$ and $\sum_{k=0}^{\infty} f_k = \infty$. For the algorithm to converge, $\{\zeta_k\}$, $\{\gamma_k\}$, and $\{\beta_k\}$ should be SSNS and satisfy

$$\sum_k (\beta_k/\gamma_k)^{d_1} < \infty, \quad \sum_k (\gamma_k/\zeta_k)^{d_2} < \infty, \tag{36}$$

for some $d_1, d_2 > 0$.

The relationships between the two critics and the actor are shown in Fig. 1. The Q*-critic* and the $\tilde{\mathbf{Q}}$*-critic* generate estimates $\mathbf{r}_k$ and $\mathbf{T}_k = (\mathbf{t}_k^1, \ldots, \mathbf{t}_k^n)$ which yield linear approximations of $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$, respectively. Both critics need to converge faster than the actor in order to track the changes in $\boldsymbol{\theta}$. Moreover, because the observed derivative q_k^i used in the $\tilde{\mathbf{Q}}$*-critic* depends on $\mathbf{r}_k$, the $\tilde{\mathbf{Q}}$*-critic* is updated faster than the Q*-critic* so that it can track changes in $\mathbf{r}_k$. We next present a result establishing a relationship between the stepsize sequences.

Proposition V.1: Suppose $\{\zeta_k\}$ and $\{\beta_k\}$ are two SSNS stepsize sequences that satisfy

$$\sum_k (\beta_k/\zeta_k)^d < \infty, \qquad \text{for some } d > 0. \tag{37}$$

Let $\gamma_k = (\zeta_k \beta_k)^{1/2}$. Then, $\{\gamma_k\}$ is also SSNS and $\{\gamma_k\}$, $\{\beta_k\}$, $\{\zeta_k\}$ satisfy (36).

Proof: Due to the assumption in (37), $\lim_{k\to\infty}(\beta_k/\zeta_k) = 0$, which implies that there exists a positive constant K such that for $\forall k > K$, $\beta_k \le \zeta_k$. Since $\{\beta_k\}$ is SSNS, it follows

$$\sum_k \gamma_k = \sum_k (\zeta_k \beta_k)^{1/2} \ge C_1 + \sum_{k=K+1}^{\infty} \beta_k = \infty,$$

where $C_1 = \sum_{k=0}^{K} \gamma_k$. Furthermore, since $\{\zeta_k\}$ is SSNS

$$\sum_k \gamma_k^2 = \sum_k \zeta_k \beta_k \le C_2 + \sum_{k=K+1}^{\infty} \zeta_k^2 < \infty,$$

where $C_2 = \sum_{k=0}^{K} \gamma_k^2$. Finally, letting $d_1 = d_2 = 2d$ and due to (37) we have

$$\sum_k (\beta_k/\gamma_k)^{d_1} = \sum_k (\gamma_k/\zeta_k)^{d_2} = \sum_k (\beta_k/\zeta_k)^d < \infty.$$

■

Proposition V.1 simplifies the selection of stepsizes. We just need to select β_k and ζ_k first and let $\gamma_k = (\zeta_k\beta_k)^{1/2}$. An example of $\{\zeta_k\}$, $\{\gamma_k\}$, and $\{\beta_k\}$ that are SSNS and satisfy (36) is: $\zeta_k = 1/k$, $\beta_k = c/(k \ln k)$, where $k > 1$ and $c > 0$, and $\gamma_k = (\zeta_k \beta_k)^{1/2} = (1/k)\sqrt{c/\ln k}$.

C. Relationship With Natural Actor-Critic Algorithms

In our approach, we use the Hessian matrix to scale the gradient in order to improve the convergence rate. A similar idea is to use the Fisher information matrix to scale the gradient. It was first proposed by [19] and several related algorithms followed [20], [23], [21]. This section discusses the relationship of the Fisher information matrix with the Hessian matrix for actor-critic algorithms.

Suppose $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is the stationary state-action distribution when the RSP parameter equals $\boldsymbol{\theta}$. [20] states that the Fisher information matrix is equal to

$$F_{\boldsymbol{\theta}} = \sum_{\mathbf{x},u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x}) \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})', \tag{38}$$

which can also be written as $\left\langle \underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}' \right\rangle_{\boldsymbol{\theta}}$, where $\boldsymbol{\psi}_{\boldsymbol{\theta}} = \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})$ (cf. (3)).

Let us now compare this expression with the true Hessian matrix (cf. (9)). If we set $Q_\theta \equiv \underline{1}$, hence, $\nabla Q_{\boldsymbol{\theta}} \equiv \underline{0}$, and ignore second derivatives with respect to $\boldsymbol{\theta}$, then the Hessian matrix degenerates to the Fisher information matrix in (38). In this sense, natural actor-critic algorithms are quasi-Newton methods that approximate the Hessian without utilizing the state-action value function $Q_{\boldsymbol{\theta}}$. In contrast, our method takes advantage of the state-action value function.

VI. Convergence

A. Linear Stochastic Approximation Driven by a Slowly Varying Markov Chain

Our Q-critic in (27) has the same form as in [14] so its convergence can be proved in a similar way. In the $\tilde{Q}$-critic (29), the increment q_k^i depends on the parameter vector $\mathbf{r}_k$. To facilitate the convergence proof of the $\tilde{Q}$-critic, this section generalizes the theory of linear stochastic approximation driven by a slowly varying Markov chain developed in [12] to the case where the objective is affected by some additional parameters $\mathbf{r}$.

Let $\{\mathbf{y}_k\}$ be a finite Markov chain whose transition probabilities depend on a parameter $\boldsymbol{\theta} \in \mathbb{R}^n$. Let $\{\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot) : \boldsymbol{\theta} \in \mathbb{R}^n, \mathbf{r} \in \mathbb{R}^N\}$ be a family of m-vector-valued functions parameterized by $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\mathbf{r} \in \mathbb{R}^N$. Let $\boldsymbol{\Xi}_k$ be some $m \times m$ matrix. Consider the following iteration to update a vector $\mathbf{s} \in \mathbb{R}^m$:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \zeta_k(\mathbf{h}_{\boldsymbol{\theta}_k,\mathbf{r}_k}(\mathbf{y}_k) - \mathbf{G}_{\boldsymbol{\theta}_k}(\mathbf{y}_k)\mathbf{s}_k) + \zeta_k \boldsymbol{\Xi}_k \mathbf{s}_k. \tag{39}$$

In the above iteration, $\mathbf{s}_k \in \mathbb{R}^m$ is the approximation vector. $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ and $\mathbf{G}_{\boldsymbol{\theta}}(\cdot)$ are m-vector-valued and $m \times m$-matrix-valued functions parameterized by $\boldsymbol{\theta}, \mathbf{r}$ and $\boldsymbol{\theta}$, respectively. Let $\mathbf{E}[\cdot]$ denote expectation. In order to establish the convergence results, we make the following assumptions.

Assumption A:

1) The sequence $\{\zeta_k\}$ is deterministic, non-increasing and SSNS.

2) The random sequence $\{\boldsymbol{\theta}_k\}$ satisfies $\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\| \leq \beta_k F_k$ for some process $\{F_k\}$ with bounded moments, where $\{\beta_k\}$ is a positive deterministic sequence such that $\sum_k (\beta_k/\zeta_k)^d < \infty$ for some $d > 0$.
3) $\boldsymbol{\Xi}_k$ is an $m \times m$-matrix valued martingale difference with bounded moments.
4) The (random) sequence $\{\mathbf{r}_k\}$ satisfies $\|\mathbf{r}_{k+1} - \mathbf{r}_k\| \leq \gamma_k F_k^r$ for some nonnegative process $\{F_k^r\}$ with bounded moments, where $\{\gamma_k\}$ is a positive sequence such that $\sum_k (\gamma_k/\zeta_k)^d < \infty$ for some $d > 0$.
5) $\mathbf{r}_k$ converges to $\bar{\mathbf{r}}(\boldsymbol{\theta}_k)$ when $k \to \infty$, namely, $\lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\| = 0$, w.p.1.
6) *(Existence of solution to the Poisson Equation.)* For each $\boldsymbol{\theta}$ and $\mathbf{r}$, there exists $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) \in \mathbb{R}^m$, $\bar{\mathbf{G}}(\boldsymbol{\theta}) \in \mathbb{R}^{m\times m}$, and corresponding m-vector and $m \times m$-matrix function $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$, $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\cdot)$ that satisfy the Poisson equation. That is, for each $\mathbf{y}$,

$$\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) = \mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) + (P_{\boldsymbol{\theta}} \hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}})(\mathbf{y}),$$

$$\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{y}) = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{y}) - \bar{\mathbf{G}}(\boldsymbol{\theta}) + (P_{\boldsymbol{\theta}} \hat{\mathbf{G}}_{\boldsymbol{\theta}})(\mathbf{y}).$$

7) *(Boundedness.)* For all $\boldsymbol{\theta}$ and $\mathbf{r}$, we have $\max(\|\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})\|, \|\bar{\mathbf{G}}(\boldsymbol{\theta})\|) \leq C$ for some constant C.
8) *(Boundedness in expectation.)* For any $d > 0$, there exists $C_d > 0$ such that $\sup_k \mathbf{E}[\|\mathbf{f}_{\boldsymbol{\theta}_k}(\mathbf{y}_k)\|^d] \leq C_d$ and $\sup_k \mathbf{E}[\|\mathbf{g}_{\boldsymbol{\theta}_k,\mathbf{r}_k}(\mathbf{y}_k)\|^d] \leq C_d$, where $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$ represents $\mathbf{G}_{\boldsymbol{\theta}}(\cdot)$ and $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\cdot)$, and $\mathbf{g}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ represents $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ and $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$.
9) *(Lipschitz continuity.)* For some constant $C > 0$, and for all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$, $\|\bar{\mathbf{G}}(\boldsymbol{\theta}) - \bar{\mathbf{G}}(\bar{\boldsymbol{\theta}})\| \leq C\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|$. For all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ and $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, $\|\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) - \bar{\mathbf{h}}(\bar{\boldsymbol{\theta}}, \bar{\mathbf{r}})\| \leq C(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|)$.
10) *(Lipschitz continuity in expectation.)* There exists a positive measurable function $C(\cdot)$ such that for every $d > 0$, $\sup_k \mathbf{E}\left[C(\mathbf{y}_k)^d\right] < \infty$. In addition, for all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$, $\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}) - \mathbf{f}_{\bar{\boldsymbol{\theta}}}(\mathbf{y})\| \leq C(\mathbf{y})\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|$, where $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$ represents $\mathbf{G}_{\boldsymbol{\theta}}(\cdot)$ and $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\cdot)$. For all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ and $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, $\|\mathbf{g}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \mathbf{g}_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}(\mathbf{y})\| \leq C(\mathbf{y})(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|)$, where $\mathbf{g}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ represents $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ and $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$.
11) There exists $a > 0$ such that for all $\mathbf{s} \in \mathbb{R}^m$ and $\boldsymbol{\theta} \in \mathbb{R}^n$, $\mathbf{s}'\bar{\mathbf{G}}(\boldsymbol{\theta})\mathbf{s} \geq a\|\mathbf{s}\|^2$.

Lemma VI.1: If Assumptions A.(1–11) are satisfied, then $\lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k)\| = 0$ w.p.1.

Proof: See Appendix A. ■

Theorem VI.2: If Assumptions A.(1–11) are satisfied, then $\lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| = 0$ w.p.1.

Proof: We have

$$\begin{aligned}&\|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \\ &\leq \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k)\| + \|\bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\|.\end{aligned}$$

Due to Assumption A.(9), we have

$$\lim_{k\to\infty} \|\bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \leq C \lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\|,$$

where C is a constant. Combining the above, we have

$$\begin{aligned}0 &\leq \lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \\ &\leq 0 + \lim_{k\to\infty} \|\bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \\ &\leq 0 + C \lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\| \\ &= 0, \qquad \text{w.p.1},\end{aligned}$$

where the second inequality follows from Lemma VI.1 and the equality is due to Assumption A.(5). We conclude that $\lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| = 0$, w.p.1.

B. Critic Convergence

In this section, we will use the results in Section VI-A to prove the convergence of the *Q-critic* and the $\tilde{\mathbf{Q}}$*-critic* presented in Section V-A. Before presenting the convergence results, we first state the following assumptions and definitions.

Assumption B: There exists a function $\tilde{L} : \mathbb{X} \to [1, \infty)$ and constants $0 \leq \rho < 1$, $b > 0$ such that for each $\boldsymbol{\theta} \in \mathbb{R}^n$,

$$\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\tilde{L}(\mathbf{x}_1)] \leq \rho\tilde{L}(\mathbf{x}) + bI_{\mathbf{x}^*}(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathbb{X}, \tag{40}$$

where $\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\cdot]$ denotes expectation under $\boldsymbol{\theta}$ with initial state $\mathbf{x}$, $I_{\mathbf{x}^*}(\cdot)$ is the indicator function for the initial state $\mathbf{x}^*$ being equal to the argument of the function, and $\mathbf{x}_1$ is the (random) state of the MDP after one transition from the initial state.

The assumption above is identical to [12, Assumption 2.5]. We call a function satisfying the inequality (40) a stochastic Lyapunov function. Let $L : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ be a function that satisfies the following assumption.

Assumption C: For each $d > 0$ there is $K_d > 0$ such that

$$\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[L(\mathbf{x}, U_0)^d] \leq K_d\tilde{L}(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathbb{X}, \boldsymbol{\theta} \in \mathbb{R}^n,$$

where U_0 is the random variable of the action at state $\mathbf{x}$.

Note that if any function is upper bounded by a function L as described in Assumption C, then all its steady-state moments are finite.

Lemma VI.3: If two functions $L_f : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ and $L_g : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ satisfy Assumption C, then so does $L_f L_g$.

Proof: For any two random variables A and B, $\mathbf{E}[AB] \leq (1/2)(\mathbf{E}[A^2] + \mathbf{E}[B^2])$. As a result, we have

$$\begin{aligned}&\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}\left[L_f(\mathbf{x}, U_0)^d L_g(\mathbf{x}, U_0)^d\right] \\ &\quad\leq \frac{1}{2}\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}\left[L_f(\mathbf{x}, U_0)^{2d}\right] + \frac{1}{2}\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}\left[L_g(\mathbf{x}, U_0)^{2d}\right] \\ &\quad\leq \frac{1}{2}(K_{2d}^f + K_{2d}^g)\tilde{L}(\mathbf{x}),\end{aligned}$$

where K_{2d}^f and K_{2d}^g are the bounding constants of f and g appearing in Assumption C. ■

Definition 1: We define $\mathcal{D}^{(2)}$ to be the family of all functions $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ that satisfy: for all $\mathbf{x} \in \mathbb{X}$ and $u \in \mathbb{U}$, there exists a constant $K > 0$ such that

$$\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, u)\| \leq KL(\mathbf{x}, u),\ \forall \boldsymbol{\theta} \in \mathbb{R}^n, \tag{41}$$

$$\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, u) - \mathbf{f}_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\| \leq K\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| L(\mathbf{x}, u),\ \forall \boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n, \tag{42}$$

where the bounding function L satisfies Assumption C.

Lemma VI.4: If $\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{g}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, then $\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{g}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ and $\mathbf{f}_{\boldsymbol{\theta}}\mathbf{g}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$.

Proof: The proof for $\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{g}_{\boldsymbol{\theta}}$ is immediate; we focus on $\mathbf{f}_{\boldsymbol{\theta}}\mathbf{g}_{\boldsymbol{\theta}}$. Inequality (41) can be proved using Lemma 4.3(f) of [4]. To prove inequality (42),

$$\begin{aligned}
\|\mathbf{f}_{\boldsymbol{\theta}}\mathbf{g}_{\boldsymbol{\theta}} - \mathbf{f}_{\bar{\boldsymbol{\theta}}}\mathbf{g}_{\bar{\boldsymbol{\theta}}}\| &= \|\mathbf{f}_{\boldsymbol{\theta}}\mathbf{g}_{\boldsymbol{\theta}} + \mathbf{f}_{\boldsymbol{\theta}}\mathbf{g}_{\bar{\boldsymbol{\theta}}} - \mathbf{f}_{\boldsymbol{\theta}}\mathbf{g}_{\bar{\boldsymbol{\theta}}} - \mathbf{f}_{\bar{\boldsymbol{\theta}}}\mathbf{g}_{\bar{\boldsymbol{\theta}}}\| \\
&\leq \|\mathbf{f}_{\boldsymbol{\theta}}\|\|\mathbf{g}_{\boldsymbol{\theta}} - \mathbf{g}_{\bar{\boldsymbol{\theta}}}\| + \|\mathbf{g}_{\bar{\boldsymbol{\theta}}}\|\|\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{f}_{\bar{\boldsymbol{\theta}}}\| \\
&\leq 2K_f K_g L_f L_g \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|,
\end{aligned}$$

where K_f and L_f are the bounding constant and the bounding function for f in (41) and (42), while K_g and L_g are the corresponding quantities for $\mathbf{g}$. According to Lemma VI.3, $L_f L_g$ also satisfies Assumption C, which completes the proof. ■

We assume $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, which is the same with Assumption 4.1 of [12]. This assumption ensures that the feature vector $\boldsymbol{\phi}_{\boldsymbol{\theta}} = (\phi^1_{\boldsymbol{\theta}}, \ldots, \phi^N_{\boldsymbol{\theta}})$, as a function of the policy parameter $\boldsymbol{\theta}$, is "well behaved." Given our feature vector definition, notice that this assumption requires that the RSP function family $\mu_{\boldsymbol{\theta}}$ is twice continuously differentiable for all $\boldsymbol{\theta}$ with bounded first and second derivatives that belong to $\mathcal{D}^{(2)}$. We also assume that the one-step reward function $g \in \mathcal{D}^{(2)}$.

The critic consists of two parts: a Q-critic that estimates $Q_{\boldsymbol{\theta}}$ (cf. (27), (28)) and a $\tilde{\mathbf{Q}}$-critic that estimates $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ (cf. (29), (30)). The Q-critic is exactly the same with the LSTD-AC algorithm [14], whose convergence has already been proved in [14] under the assumptions imposed. For the $\tilde{\mathbf{Q}}$-critic, denote by $\mathbf{V}(\mathbf{A})$ a column vector stacking all columns in a matrix $\mathbf{A}$. The $\tilde{\mathbf{Q}}$-critic can be written as in (39) if we let

$$\mathbf{s}_k = \left[M\eta^1_k \cdots M\eta^n_k\ (\mathbf{v}^1_k)' \cdots (\mathbf{v}^n_k)'\right]', \tag{43}$$

$$\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) = \begin{bmatrix} M\Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u) \\ \Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)\mathbf{V}(\mathbf{z}\boldsymbol{\psi}'_{\boldsymbol{\theta}}(\mathbf{x}, u)) \end{bmatrix},$$

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{y}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathrm{diag}(\mathbf{z}, \ldots, \mathbf{z})/M & \mathbf{I} \end{bmatrix}$$

$$\boldsymbol{\Xi}_k = \mathbf{0},$$

where $\mathrm{diag}(\mathbf{z}, \ldots, \mathbf{z})$ denotes an $nN \times n$ block diagonal matrix with every diagonal element being equal to $\mathbf{z}$, $\mathbf{y} = (\mathbf{x}, u, \mathbf{z})$, M is an arbitrary (large) positive constant whose role is to facilitate the convergence proof, and at any iteration k of (39) $\mathbf{r}_k$ iterates as in (28). The stochastic process $\{\mathbf{z}_k\}$ is the eligibility trace iterating as in (27).

To prove the convergence of the $\tilde{\mathbf{Q}}$-*critic*, we just need to verify Assumptions A.(1–11). It is easy to verify that $\mathbf{z}_k = \sum_{l=0}^{k-1} \lambda^{k-l-1}\boldsymbol{\phi}_{\boldsymbol{\theta}_l}(\mathbf{x}_l, u_l)$. First, we establish the following lemma.

Lemma VI.5: For every $d > 0$, we have $\sup_k \mathbf{E}[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d] < \infty$, where $L : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ is a bounded function that satisfies Assumption C.

Proof: According to the triangle inequality, we have

$$\begin{aligned}
\|\mathbf{z}_k\|^d &= \|\sum_{l=0}^{k-1} \lambda^{k-l-1}\boldsymbol{\phi}_{\boldsymbol{\theta}_l}(\mathbf{x}_l, u_l)\|^d \\
&\leq \sum_{l=0}^{k-1} \lambda^{d(k-l-1)}\|\boldsymbol{\phi}_{\boldsymbol{\theta}_l}(\mathbf{x}_l, u_l)\|^d \\
&\leq K_1 \sum_{l=0}^{k-1} \lambda^{d(k-l-1)} L_1(\mathbf{x}_l, u_l)^d,
\end{aligned}$$

for some bounded function L_1 that satisfies Assumption C and some positive constant K_1, where the last inequality is due to $\boldsymbol{\phi}_{\boldsymbol{\theta}_k} \in \mathcal{D}^{(2)}$. In addition, we can multiply with $L(\mathbf{x}_k, u_k)^d$ and take expectation on both sides of the above, which yields

$$\begin{aligned}
&\mathbf{E}[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d] \\
&\quad \leq K_1 \sum_{l=0}^{k-1} \lambda^{d(k-l-1)} \mathbf{E}[L(\mathbf{x}_k, u_k)^d L_1(\mathbf{x}_l, u_l)^d].
\end{aligned} \tag{44}$$

Similar to the proof of Lemma VI.3,

$$\begin{aligned}
&\mathbf{E}[L(\mathbf{x}_k, u_k)^d L_1(\mathbf{x}_l, u_l)^d] \\
&\quad \leq \frac{1}{2}\mathbf{E}[L(\mathbf{x}_k, u_k)^{2d}] + \frac{1}{2}\mathbf{E}[L_1(\mathbf{x}_l, u_l)^{2d}] < \infty.
\end{aligned} \tag{45}$$

Combining (44) and (45), we establish that $\mathbf{E}[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d]$ is bounded. ■

Theorem VI.6: Under iterations (27) and (28),

$$\|\mathbf{r}_{k+1} - \mathbf{r}_k\| \leq \gamma_k F^r_k, \qquad \text{w.p.1}, \tag{46}$$

for some random sequence $\{F^r_k\}$ that has bounded moments, where $\{\gamma_k\}$ is the stepsize in (27).

Proof: SeeAppendix B. ■

Using SSNS stepsizes according to (36), Assumptions A.(1) and (4) will be satisfied because of Theorem VI.6. Now, $\|\mathbf{r}\|\Gamma(\mathbf{r})$ is bounded because of (35). According to (31), $\mathbf{U}_k$ has bounded moments because $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$, $\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$, $Q_{\boldsymbol{\theta}}$, and $\tilde{Q}^i_{\boldsymbol{\theta}}$, $\forall i$, have bounded moments. $\mathbf{H}_k$ and $\hat{\mathbf{H}}_k$ should also have bounded moments because the update in (32) is applied only when $\mathbf{U}_k$ is positive definite. As a result, $\Gamma(\mathbf{r}_k)\mathbf{r}'_k\boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k)\hat{\mathbf{H}}_k\boldsymbol{\psi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k)$ should have bounded moments, thus, Assumption A.(2) holds. Assumption A.(3) is trivially satisfied. In addition, because the *Q-critic* converges, we have

$$\lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\| = 0, \qquad \text{w.p.1},$$

which is Assumption A.(5).

For $i = 1, \ldots, n$, define the function $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} = \boldsymbol{\phi}_{\boldsymbol{\theta}}\psi^i_{\boldsymbol{\theta}}$. Because $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ and $\boldsymbol{\psi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, we obtain $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ according to Lemma VI.4. Notice that for any fixed $\mathbf{r}$ and $\boldsymbol{\theta}$, the $\tilde{\mathbf{Q}}$-critic (43) is equivalent to the Q-critic of an artificial Markov decision process with reward function $g^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{x}, u) = \Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u)$, $i = 1, \ldots, n$. As a result, the Poisson equations of Assumption A.(6)

should be satisfied with appropriately defined average steady-state quantities $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})$ and $\bar{\mathbf{G}}(\boldsymbol{\theta})$. More specifically, similar to [4, Sec. 5.2], we have

$$\bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r}) = \left\langle \underline{1}, g^i_{\boldsymbol{\theta},\mathbf{r}} \right\rangle_{\boldsymbol{\theta}},$$

$$\bar{\mathbf{z}}(\boldsymbol{\theta}) = (1-\lambda)^{-1} \left\langle \underline{1}, \phi_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}},$$

$$\mathbf{h}^i_1(\boldsymbol{\theta}, \mathbf{r}) = \sum_{k=0}^{\infty} \lambda^k \left\langle P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} - \bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})\underline{1}, \phi_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}},$$

$$\begin{aligned}\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) = (&M\bar{\kappa}^1(\boldsymbol{\theta}, \mathbf{r}), \ldots, M\bar{\kappa}^n(\boldsymbol{\theta}, \mathbf{r}),\\ &(\mathbf{h}^1_1(\boldsymbol{\theta}, \mathbf{r}) + \bar{\kappa}^1(\boldsymbol{\theta}, \mathbf{r})\bar{\mathbf{z}}(\boldsymbol{\theta}), \ldots,\\ &(\mathbf{h}^n_1(\boldsymbol{\theta}, \mathbf{r}) + \bar{\kappa}^n(\boldsymbol{\theta}, \mathbf{r})\bar{\mathbf{z}}(\boldsymbol{\theta})),\end{aligned}$$

$$\bar{\mathbf{G}}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathrm{diag}(\bar{\mathbf{z}}(\boldsymbol{\theta}), \ldots, \bar{\mathbf{z}}(\boldsymbol{\theta}))/M & \mathbf{I} \end{bmatrix},$$

where $P^k_{\boldsymbol{\theta}}$ denotes the application of the operator $P_{\boldsymbol{\theta}}$ k times. We can interpret $\bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})$ as the steady-state expectation of the "observed reward" function $g^i_{\boldsymbol{\theta},\mathbf{r}}$.

Let now $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) = \Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u)\mathbf{z}$, $i = 1, \ldots, n$. It can be seen that if $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}$ are bounded and Lipschitz continuous in expectation for all $i = 1, \ldots, n$, then $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}$ should also be bounded and Lipschitz continuous in expectation. Recall that $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$. For $i = 1, \ldots, n$ and each $d > 0$,

$$\begin{aligned}&\sup_k \mathbf{E}\left[\|\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}_k)\|^d\right]\\ &\leq (\Gamma(\mathbf{r})\|\mathbf{r}\|)^d \sup_k \mathbf{E}\left[\|\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}_k, u_k)\|^d \|\mathbf{z}_k\|^d\right]\\ &\leq (\Gamma(\mathbf{r})\|\mathbf{r}\|)^d K^d \sup_k \mathbf{E}\left[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d\right],\end{aligned}$$

for some function L that satisfies Assumption C and some positive constant K. According to (35), $\Gamma(\mathbf{r})\|\mathbf{r}\|$ is bounded. Using Assumption C and Lemma VI.5, it follows that $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}$ satisfies Assumption A.(8). Using Lemma VI.5 it also follows that $\mathbf{G}_{\boldsymbol{\theta}}$ satisfies the same assumption.

It is easy to verify that the function $f(\mathbf{r}) = \Gamma(\mathbf{r})\mathbf{r}$ is Lipschitz continuous and suppose its Lipschitz constant is C_Γ. We will next prove that $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y})$ is Lipschitz continuous in expectation. For all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$, $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, and $i = 1, \ldots, n$, we have

$$\begin{aligned}&\|\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \tilde{\mathbf{h}}^i_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}(\mathbf{y})\|\\ &\leq \|\Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u)\mathbf{z} - \Gamma(\bar{\mathbf{r}})\bar{\mathbf{r}}'\boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\mathbf{z}\|\\ &\leq \|\mathbf{z}\|\Gamma(\mathbf{r})\|\mathbf{r}'\left(\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u) - \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\right)\|\\ &\quad + \|\mathbf{z}\|\|\left(\Gamma(\mathbf{r})\mathbf{r} - \Gamma(\bar{\mathbf{r}})\bar{\mathbf{r}}\right)'\boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\|\\ &\leq \|\mathbf{z}\|\Gamma(\mathbf{r})\|\mathbf{r}\|\|\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u) - \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\|\\ &\quad + \|\mathbf{z}\|\|\boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\|C_\Gamma\|\mathbf{r} - \bar{\mathbf{r}}\|. \end{aligned} \tag{47}$$

Recall that $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$. Let K and L be the bounding constant and the bounding function for $\boldsymbol{\xi}^i_{\boldsymbol{\theta}}$; then

$$\|\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \tilde{\mathbf{h}}^i_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}(\mathbf{y})\| \leq C(\mathbf{y})\left(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|\right),$$

where $C(\mathbf{y}) = (\Gamma(\mathbf{r})\|\mathbf{r}\| + C_\Gamma)KL(\mathbf{x}, u)\|\mathbf{z}\|$ and $\mathbf{y} = (\mathbf{x}, u, \mathbf{z})$. Using the fact that $\Gamma(\mathbf{r})\|\mathbf{r}\|$ is bounded and Lemma VI.5, it follows that $\mathbf{E}[C(\mathbf{y})^d] < \infty$ for each $d > 0$. As a result, $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}$ satisfies Assumption A.(10). Moreover, replicating an argument from [4, Sec. 5.2] it can also be shown that $\mathbf{G}_{\boldsymbol{\theta}}$ satisfies the same assumption. Furthermore, defining

$$\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) = \sum_{k=0}^{\infty} \mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})|\mathbf{y}_0 = \mathbf{y}],$$

$$\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{y}) = \sum_{k=0}^{\infty} \mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{y}_k) - \bar{\mathbf{G}}(\boldsymbol{\theta})|\mathbf{y}_0 = \mathbf{y}],$$

we can use similar arguments as above to establish that these functions satisfy Assumption A.(8) and (10).

Lemma VI.7: Let $\hat{\boldsymbol{\theta}} = (\boldsymbol{\theta}, \mathbf{r})$. Let also $\hat{\mathcal{D}}^{(2)}$ be the counterpart of $\mathcal{D}^{(2)}$ for functions parameterized by $\hat{\boldsymbol{\theta}}$. Then $P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}}$ belongs to $\hat{\mathcal{D}}^{(2)}$ for all nonnegative integers k.

Proof: A simple observation is that $\mathcal{D}^{(2)} \subseteq \hat{\mathcal{D}}^{(2)}$ and that Lemma VI.4 still holds for $\hat{\mathcal{D}}^{(2)}$. Namely, a product function $f_{\hat{\boldsymbol{\theta}}} g_{\hat{\boldsymbol{\theta}}} \in \hat{\mathcal{D}}^{(2)}$ if $f_{\hat{\boldsymbol{\theta}}} \in \hat{\mathcal{D}}^{(2)}$ and $g_{\hat{\boldsymbol{\theta}}} \in \hat{\mathcal{D}}^{(2)}$.

$P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}}$ can be written as $P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} = \Gamma(\mathbf{r})\mathbf{r}' P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}}$. We first observe that $P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ according to [32, Corollary 2.4]. To verify (41), we have (in functional relationships)

$$\|P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}}\| \leq \Gamma(\mathbf{r})\|\mathbf{r}\|\|P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}}\| \leq \Gamma(\mathbf{r})\|\mathbf{r}\|KL.$$

To verify (42), for $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ and $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, we have

$$\begin{aligned}&\|P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} - P^k_{\bar{\boldsymbol{\theta}}} g^i_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}\|\\ &\leq \Gamma(\mathbf{r})\|\mathbf{r}\|\|P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}} - P^k_{\bar{\boldsymbol{\theta}}} \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}\| + \|P^k_{\bar{\boldsymbol{\theta}}} \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}\|C_\Gamma\|\mathbf{r} - \bar{\mathbf{r}}\|\\ &\leq \Gamma(\mathbf{r})\|\mathbf{r}\|KL\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + KLC_\Gamma\|\mathbf{r} - \bar{\mathbf{r}}\|\\ &\leq \left(\Gamma(\mathbf{r})\|\mathbf{r}\| + C_\Gamma\right)KL\left(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|\right),\end{aligned}$$

where K and L are the bounding constant and function of $P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}}$, respectively. ∎

Using the fact that $g^i_{\boldsymbol{\theta},\mathbf{r}}, \boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, $\bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})$ and $\bar{\mathbf{z}}(\boldsymbol{\theta})$ are bounded and Lipschitz continuous with respect to $\hat{\boldsymbol{\theta}}$ due to [32, Corollary 5.3]. It can be easily verified that $(P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} - \bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})\underline{1})\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \hat{\mathcal{D}}^{(2)}$ using Lemma VI.7 and Lemma VI.4. Again, using [32, Corollary 5.3], we can obtain that $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})$ is bounded and Lipschitz continuous with respect to $\hat{\boldsymbol{\theta}}$. As a result, $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})$ satisfies Assumption A.(7) and (9). Similarly, it can also be shown that $\bar{\mathbf{G}}(\boldsymbol{\theta})$ satisfies the same assumptions. Finally, it can also be verified that $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y})$ and $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{y})$ satisfy the same assumptions using similar arguments.

The final step in verifying all parts of Assumption A is part (11). That follows from [4, Lemma 5.3]. Having established all parts of Assumption A, the convergence of the $\mathbf{Q}$-critic follows.

C. Actor Convergence

The actor update defined in (34) is similar to the actor update using the unscaled gradient. The difference is that the gradient estimate is multiplied by a positive definite matrix. This section will present the convergence results for this type of actors.

Define

$$\mathbf{S}_{\boldsymbol{\theta}}(\mathbf{x},u) = \mathbf{H}_{\boldsymbol{\theta}}\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x},u)\boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}(\mathbf{x},u),$$

where $\mathbf{H}_{\boldsymbol{\theta}}$ is a positive definite matrix for all $\boldsymbol{\theta}$. Let $\bar{\mathbf{S}}(\boldsymbol{\theta}) = \langle \underline{1}, \mathbf{S}_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}}$ and let $\bar{\mathbf{r}}(\boldsymbol{\theta})$ be the limit of the critic parameter $\mathbf{r}$ if the policy parameter is held fixed to $\boldsymbol{\theta}$. Similar to [12], the *actor* update can be written as

$$\begin{aligned}\boldsymbol{\theta}_{k+1} =\ & \boldsymbol{\theta}_k + \beta_k \mathbf{S}_{\boldsymbol{\theta}}(\mathbf{x}_k,u_k)\mathbf{r}_k\Gamma(\mathbf{r}_k)\\ =\ & \boldsymbol{\theta}_k + \beta_k\bar{\mathbf{S}}(\boldsymbol{\theta}_k)\bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\\ & + \beta_k\left(\mathbf{S}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\right)\mathbf{r}_k\Gamma(\mathbf{r}_k)\\ & + \beta_k\bar{\mathbf{S}}(\boldsymbol{\theta}_k)\left(\mathbf{r}_k\Gamma(\mathbf{r}_k) - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\right).\end{aligned}$$

Define

$$\begin{aligned}\mathbf{f}(\boldsymbol{\theta}_k) &= \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\bar{\mathbf{r}}(\boldsymbol{\theta}_k),\\ \mathbf{e}_k^{(1)} &= \left(\mathbf{S}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\right)\mathbf{r}_k\Gamma(\mathbf{r}_k),\\ \mathbf{e}_k^{(2)} &= \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\left(\mathbf{r}_k\Gamma(\mathbf{r}_k) - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\right).\end{aligned}$$

Then, the actor update becomes:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \beta_k\left(\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\mathbf{f}(\boldsymbol{\theta}_k) + e_k^{(1)} + e_k^{(2)}\right).$$

$\mathbf{f}(\boldsymbol{\theta}_k)$ is the expected actor update, while $e_k^{(1)}$ and $e_k^{(2)}$ are two error terms due to the fact that the update is performed on a sample path of the MDP. Using Taylor's series expansion,

$$\begin{aligned}\bar{\alpha}(\boldsymbol{\theta}_{k+1}) \geq\ & \bar{\alpha}(\boldsymbol{\theta}_k) + \beta_k\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\nabla\bar{\alpha}(\boldsymbol{\theta}_k)^{'}\mathbf{f}(\boldsymbol{\theta}_k)\\ & + \beta_k\nabla\bar{\alpha}(\boldsymbol{\theta}_k)^{'}\mathbf{e}_k^{(1)} + \beta_k\nabla\bar{\alpha}\left(\boldsymbol{\theta}_k\right)^{'}\mathbf{e}_k^{(2)}.\end{aligned}$$

Lemma VI.8: (Convergence of the noise terms). It holds:
- $\sum_{k=0}^{\infty}\beta_k\nabla\bar{\alpha}(\boldsymbol{\theta}_k)^{'}\mathbf{e}_k^{(1)}$ converges w.p.1.
- $\lim_k \mathbf{e}_k^{(2)} = 0$ w.p.1.

Proof: Let $\hat{\mathbf{e}}_k^{(1)} = \left(\boldsymbol{\xi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \bar{\boldsymbol{\xi}}(\boldsymbol{\theta}_k)\right)\mathbf{r}_k\Gamma(\mathbf{r}_k)$ and $\hat{\mathbf{e}}_k^{(2)} = \bar{\boldsymbol{\xi}}(\boldsymbol{\theta}_k)\left(\mathbf{r}_k\Gamma(\mathbf{r}_k) - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\right)$, where $\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{x},u) = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x},u)\boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}(\mathbf{x},u)$ and $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta}) = \langle\underline{1},\boldsymbol{\xi}_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}} = \langle\boldsymbol{\psi}_{\boldsymbol{\theta}},\boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}\rangle_{\boldsymbol{\theta}}$. Then, $\hat{\mathbf{e}}_k^{(1)}$ and $\hat{\mathbf{e}}_k^{(2)}$ are the two error terms for the actor update using the unscaled gradient [4]. It easily follows that $\mathbf{e}_k^{(1)} = \mathbf{H}_{\boldsymbol{\theta}_k}\hat{\mathbf{e}}_k^{(1)}$ and $\mathbf{e}_k^{(2)} = \mathbf{H}_{\boldsymbol{\theta}_k}\hat{\mathbf{e}}_k^{(2)}$. Furthermore, $\mathbf{S}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) = \mathbf{H}_{\boldsymbol{\theta}_k}^{-1}\boldsymbol{\xi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k)$. The lemma can be proved by combining these facts with [4, Lemma 6.2]. ∎

Lemma VI.8 shows that $\mathbf{e}_k^{(1)}$ can be averaged out and $\mathbf{e}_k^{(2)}$ goes to zero. As a result, the two error terms are negligible and the update is determined by the expected direction $\mathbf{f}(\boldsymbol{\theta})$ in the long run.

Lemma VI.9: We have $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{\theta}) + \varepsilon(\lambda,\boldsymbol{\theta})$, where $\mathbf{g}(\boldsymbol{\theta})$ is a function such that $\nabla\bar{\alpha}(\boldsymbol{\theta})^{'}\mathbf{g}(\boldsymbol{\theta}) \geq 0$, and $\sup_{\boldsymbol{\theta}}|\varepsilon(\lambda,\boldsymbol{\theta})| < C(1-\lambda)$ for some constant $C > 0$ independent of λ.

Proof: According to (5), $\nabla\bar{\alpha}(\boldsymbol{\theta}) = \langle\boldsymbol{\psi}_{\boldsymbol{\theta}}, Q_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}} = \langle\boldsymbol{\psi}_{\boldsymbol{\theta}},\boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}\bar{\mathbf{r}}(\boldsymbol{\theta})\rangle_{\boldsymbol{\theta}} = \bar{\boldsymbol{\xi}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta})$. For $\lambda = 1$, we have

$$\begin{aligned}\nabla\bar{\alpha}(\boldsymbol{\theta})^{'}\mathbf{f}(\boldsymbol{\theta}) =\ & \nabla\bar{\alpha}(\boldsymbol{\theta})^{'}\bar{\mathbf{S}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta})\\ =\ & \bar{\mathbf{r}}(\boldsymbol{\theta})^{'}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})^{'}\bar{\mathbf{S}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta}).\end{aligned}$$

Notice that $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})^{'}\bar{\mathbf{S}}(\boldsymbol{\theta}) \succeq 0$. Specifically,

$$\begin{aligned}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})^{'}\bar{\mathbf{S}}(\boldsymbol{\theta}) =\ & \langle\boldsymbol{\psi}_{\boldsymbol{\theta}}^{'},\boldsymbol{\phi}_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}}\left\langle\mathbf{H}_{\boldsymbol{\theta}},\boldsymbol{\psi}_{\boldsymbol{\theta}}\boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}\right\rangle_{\boldsymbol{\theta}}\\ =\ & \mathbf{H}_{\boldsymbol{\theta}}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})^{'}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta}),\end{aligned}$$

where $\mathbf{H}_{\boldsymbol{\theta}} \succeq 0$ and $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})^{'}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta}) \succeq 0$ by construction. As a result, $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})^{'}\bar{\mathbf{S}}(\boldsymbol{\theta}) \succeq 0$, which implies that $\nabla\bar{\alpha}(\boldsymbol{\theta})^{'}\mathbf{f}(\boldsymbol{\theta}) \geq 0$.

The proof for $\lambda < 1$ follows the proof in [4]. Let us write $\bar{\mathbf{r}}^{\lambda}(\boldsymbol{\theta})$ for the steady-state expectation of $\mathbf{r}_k$. Following the proof of [4], we have $\|\bar{\mathbf{r}}^{\lambda}(\boldsymbol{\theta}) - \mathbf{r}(\boldsymbol{\theta})\| \leq C_0(1-\lambda)$ for some positive constant C_0. Let $\mathbf{g}(\boldsymbol{\theta}) = \bar{\mathbf{S}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta})$, where $\bar{\mathbf{r}}(\boldsymbol{\theta})$ is the steady-state expectation of $\mathbf{r}_k$ when $\lambda = 1$. Then we can still obtain $\nabla\bar{\alpha}(\boldsymbol{\theta})^{'}\mathbf{g}(\boldsymbol{\theta}) \geq 0$. In addition, $\|\mathbf{f}(\boldsymbol{\theta}) - \mathbf{g}(\boldsymbol{\theta})\| = \|\bar{\mathbf{S}}(\boldsymbol{\theta})(\bar{\mathbf{r}}^{\lambda}(\boldsymbol{\theta}) - \bar{\mathbf{r}}(\boldsymbol{\theta}))\| \leq C(1-\lambda)$ for some C. ∎

Lemma VI.9 shows that the expected direction $\mathbf{f}(\boldsymbol{\theta})$ is always a gradient ascent direction for λ sufficiently close to 1. We arrive at the following convergence result whose proof is similar to [4, Thm. 6.3].

Theorem VI.10 Actor Convergence: For any $\epsilon > 0$, there exists some λ sufficiently close to 1 such that the second-order Actor-Critic algorithm satisfies $\lim_{k\to\infty}\inf_k|\nabla\bar{\alpha}(\theta_k)| < \epsilon$ w.p.1. That is, $\boldsymbol{\theta}_k$ visits an arbitrary neighborhood of a stationary point infinitely often.

VII. Case Study

A. Garnet Problem

This section reports empirical results from our method applied to GARNET problems introduced in [23]. GARNET problems do not correspond to any particular application; they are meant to be generic, yet, representative of MDPs one encounters in practical applications [23]. As we mentioned earlier, GARNET stands for "Generic Average Reward Non-stationary Environment Testbed."

A GARNET problem is characterized by 5 parameters and can be written as GARNET(n,m,b,σ,τ). The parameters n and m are the number of states and actions, respectively. For each state-action pair, there are b possible next states, and each next state is chosen randomly without replacement. The transition probabilities to these b states are generated as follows: we divide a unit-length interval into b segments by choosing $b-1$ breaking points according to a uniform random distribution. The lengths of these segments represent the transition probabilities and they are randomly assigned to the b states we have already selected.

The expected reward for each transition is a normally distributed random variable with zero mean and unit variance. The actual reward is a normally distributed random variable whose mean is the expected reward and whose variance is 1.

The parameter $\tau, 0 \leq \tau \leq 1/n$, determines the degree of non-stationarity in the problem. If $\tau = 0$, the GARNET problem is stationary. Otherwise, if $\tau > 0$, one of the states will be selected with probability $n\tau$ at each time step and randomly reconstructed as described above.

To apply the actor-critic algorithm, the key step is to design an RSP $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$. In this case study, we define the RSP to be the Boltzmann distribution that is based on some state-action features. Good state-action features should be interpretable and could help reduce the number of parameters in the RSP.

We first define the state feature $\mathbf{f}_{\mathsf{S}}(\mathbf{x})$ to be a binary vector of length d, i.e., $\mathbf{f}_{\mathsf{S}}(\mathbf{x}) \in \{0,1\}^d$, for each state $\mathbf{x}$. There is a parameter l specifying the number of components in the state feature that are equal to 1. State features are randomly generated and we make sure no two states have the same state feature.

In [23], the state-action feature is constructed by padding zeros to state features so that the features for different actions are orthogonal. As a result, the dimensionality of the state-action feature constructed in this manner is equal to $d|\mathbb{U}|$. This approach significantly increases the feature dimensionality, especially when the action space is very large. In this paper, we use the state-action feature described below. For each state $\mathbf{x}_0$ and action u, the state-action feature is:

$$\mathbf{f}_{\mathsf{SA}}(\mathbf{x}_0, u) = \mathbf{E}[\mathbf{f}_{\mathsf{S}}(\mathbf{x}_1)|u] - \mathbf{f}_{\mathsf{S}}(\mathbf{x}_0), \tag{48}$$

where $\mathbf{E}[\mathbf{f}_{\mathsf{S}}(\mathbf{x}_1)|u] = \sum_{\mathbf{x}_1} p(\mathbf{x}_1|\mathbf{x}, u)\mathbf{f}_{\mathsf{S}}(\mathbf{x}_1)$ is the expected feature at the next state after applying action u.

With the state-action feature as in (48), the probability of taking action u in state $\mathbf{x}$ is set to

$$\mu_{\boldsymbol{\theta}}(u|\mathbf{x}) = \frac{e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}{\sum_{u\in\mathbb{U}} e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}, \tag{49}$$

which is a typical Boltzmann distribution with T being the temperature of the distribution. With the state-action feature described above, we can interpret $-\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}$ as the "energy" and the distribution prefers actions that lead to lower energy.

A common consideration in RSP design is the so-called exploitation-exploration tradeoff [2]. An RSP exhibits higher exploitation if it is more greedy, i.e., it is more likely to only pick the most desirable action. However, sometimes the exploration of undesirable actions is necessary because they may be desirable in the long run. High exploitation and low exploration may result in a sub-optimal solution. On the contrary, low exploitation and high exploration may reduce the convergence rate of the actor-critic algorithm. Our RSP defined in (49) is flexible because tuning T in (49) can effectively adjust the degree of exploration. High temperature T implies more exploration while low temperature T implies more exploitation.

In this empirical study, we compare our algorithm with the LSTD-AC algorithm in [14], and the four algorithms in [23], which are henceforth referred to as BSGL1 to BSGL4, in a GARNET problem GARNET$(50, 4, 5, 0.1, 0)$. BSGL1 is based on a "vanilla" gradient ascent and BSGL2-BSGL4 are based on natural gradients. Henceforth, for state features we let $d = 8$ and $l = 3$. The state-features are randomly assigned and we make sure no two states have the same state-feature. For all algorithms, the critic step-size is $\alpha_k = \frac{\alpha_0 \cdot \alpha_c}{\alpha_c + k^{2/3}}$ and the actor stepsize $\beta_c = \frac{\beta_0 \cdot \beta_c}{\beta_c + k}$, where $\alpha_c = \beta_c = 1000$. For the LSTD actor-critic and our method $\alpha_0 = 0.1$ and $\beta_0 = 0.1$. For BSGL1 and BSGL2, $\alpha_0 = 0.1$ and $\beta_0 = 0.01$. For BSGL3 and BSGL4, we choose $\alpha_0 = 0.01$ and $\beta_0 = 0.001$. For all algorithms, the initial parameters

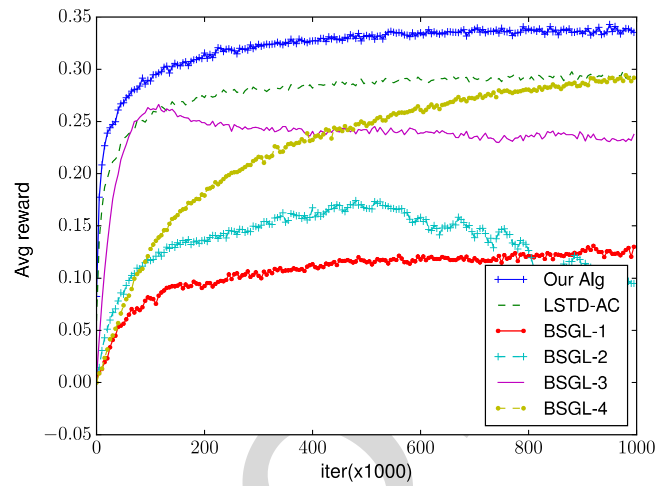

Fig. 2. Comparison of our algorithm with LSTD and natural actor-critic algorithms.

$\boldsymbol{\theta}_0$ are zero and the temperature in (49) is set to $T = 1$. For our algorithm, we choose $\chi_{\min} = 100$ (cf. (33)).

We run each algorithm 50 times independently and Fig. 2 displays the mean of the average reward for the first 1,000,000 iterations. Table I summarizes the convergence time and converged average reward for each algorithm. For each problem, the first two columns of Table I show the mean and standard deviation of the reward achieved. The third and fourth columns list the time (mean and standard deviation) it takes to convergence. The last column shows the average CPU time per iteration (TPI). The results are based on 50 independent runs for the GARNET problem and 100 independent runs for the robot control problem. Note that BSGL2 becomes numerically unstable after 500,000 iterations, so the reward of BSGL2 in Table I is the maximal reward before numerical instability occurs and the time is the time it takes to reach the maximal reward.

Compared to the LSTD-AC method, our method adds a second-order critic update and takes advantage of the Hessian estimate in the actor update. For this problem, the average CPU time of one LSTD-AC iteration is 1288 μs. In comparison, the average CPU time for one iteration of our algorithm is 1818 μs, which means that computing the second-order critic and the inverse of the Hessian adds about 41% to the computational cost. Despite the larger CPU time per iteration, our algorithm still converges faster than LSTD-AC because fewer iterations are needed. The CPU time per iteration of both our algorithm and LSTD-AC is larger than BSGL1-4. This is likely because both our algorithm and LSTD-AC use a state-action feature vector, whose dimensionality is larger than the one used in BSGL1-4 for value function approximations.

Among the four algorithms in [23], BSGL3 converges faster, which is consistent with the empirical study in [23]. Compared to BSGL3, although our algorithm uses longer time to converge, it converges to higher value (0.33) than BSGL3 (0.24). On average our algorithm takes only 43 seconds to reach an average reward of 0.24 vs. 122 seconds needed by BSGL3 to reach the same value.

TABLE I
COMPARISON OF ALL ALGORITHMS IN A GARNET AND A ROBOT CONTROL PROBLEM.

Alg. Name	GARNET					Robot Control				
	Reward		Conv. Time (s)		TPI(μ s)	Reward		Conv. Time (s)		TPI(μ s)
	Mean	Std	Mean	Std		Mean	Std	Mean	Std	
Our Alg.	0.33	0.070	727	10.9	1818	0.0916	0.00109	118	3.0	3281
LSTD-AC	0.29	0.091	773	9.9	1288	0.0851	0.0235	187	23	2837
BSGL-1	0.11	0.083	540	7.5	601	0.0819	0.000731	217	2.9	2173
BSGL-2	0.16	0.078	342	4.4	684	0.0909	0.00136	231	9.8	2313
BSGL-3	0.24	0.093	122	1.6	678	0.0927	0.000936	142	6.4	2372
BSGL-4	0.28	0.082	686	11.6	686	0.0916	0.000860	209	5.0	2319

For BSGL2, the Table Displays the Maximal Average Reward Before Numerical Instability Happens and the Time to Reach the Reward

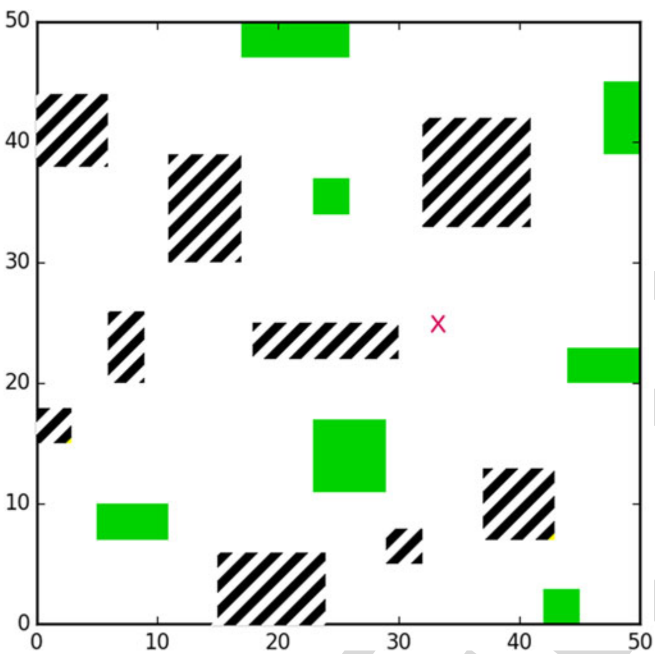

Fig. 3. View of the mission environment, where the initial region is marked by ``x'', the goal regions are marked by green colors, and the unsafe regions are displayed in black stripes.

B. Robot Control Problem

In this section we compare the performance of our algorithm with other algorithms in a robotics application. Fig. 3 shows the mission environment, which is a 50×50 grid. We model the motion of the robot in the environment as the following MDP $\mathbb{M}$:

- **State space**. Each state $\mathbf{x} \in \mathbb{X}$ corresponds to a region in the mission environment and can be represented by a coordinate (i, j), where i is the row number and j is the column number.
- **Action space**. The action space $\mathbb{U} = \{u_1, u_2, u_3, u_4\}$ corresponds to four control primitives (actions): "North," "East," "South," and "West," which represent the directions in which the robot intends to move. Depending on the location of a region, some of these actions may not be enabled, for example, in the lower-left corner, only actions "North" and "East" are enabled. For each state $\mathbf{x}$, let $\mathbb{U}_e(\mathbf{x})$ denote the enabled actions in this state.
- **Transitional model**. A control action does not necessarily lead the robot to the intended direction because the outcome is subject to noise in actuation and possible surface roughness in the environment. In this problem, a robot can only move to the adjacent state in one step. For each enabled control, the robot moves to the intended direction with probability 0.7 and moves to other allowed directions with equal probabilities.
- **Initial state**. The robot starts from state $\mathbf{x}_0$, which is labeled as 'x' in Fig. 3.
- **Reward function**. There are some *unsafe* regions $\mathbb{X}_U$, which should be avoided, in the mission environment. There are also some *goal* states $\mathbb{X}_G$ that should be visited as often as possible. The *unsafe* and *goal* states are displayed as black stripes and green solid colors in Fig. 3, respectively. The objective is to find an optimal policy that maximizes the *expected average reward* with an one-step reward function defined by

$$g(\mathbf{x}, u) = \begin{cases} 1, & \text{if } x \in \mathbb{X}_G, \\ -1, & \text{if } x \in \mathbb{X}_U, \\ 0, & \text{otherwise.} \end{cases}$$

This problem is the foundation of many complex robot motion control problems in which MDPs are defined in more complex ways, i.e., using temporal logic [15]–[17].

In this problem, we consider two state features that represent the *safety* and *affinity* of the state, respectively. For each pair of states $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}$, we define $d(\mathbf{x}_i, \mathbf{x}_j)$ to be the minimum number of transitions from $\mathbf{x}_i$ to $\mathbf{x}_j$. We say $\mathbf{x}_j \in \mathscr{N}(\mathbf{x}_i)$—a neighborhood of $\mathbf{x}_i$—if and only if $d(\mathbf{x}_i, \mathbf{x}_j) \leq r_n$, for some fixed integer r_n given *a priori*. For each state $\mathbf{x} \in \mathbb{X}$, the safety score is defined as the ratio of the safe neighboring states over all neighboring states of $\mathbf{x}$. Namely,

$$\mathsf{safety}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \mathscr{N}(\mathbf{x})} I_s(\mathbf{y})}{|\mathscr{N}(\mathbf{x})|}, \tag{50}$$

where $I_s(\mathbf{y})$ is an indicator function such that $I_s(\mathbf{y}) = 1$ if and only if $\mathbf{y} \in \mathbb{X} \setminus \mathbb{X}_U$ and $I_s(\mathbf{y}) = 0$ otherwise. A higher safety score for the current state of the robot means it is less likely for

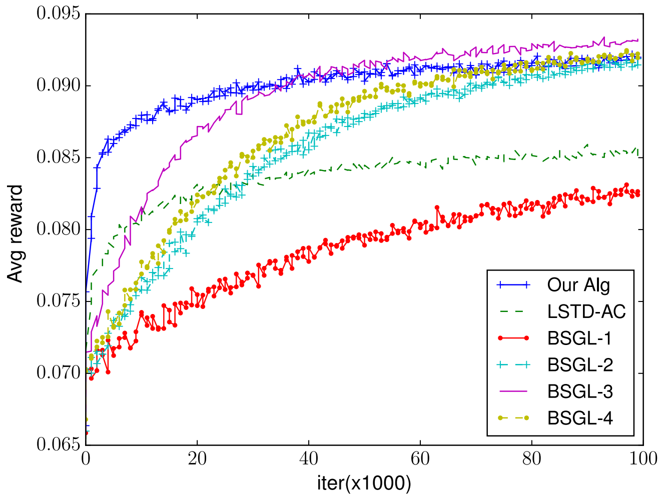

Fig. 4. Comparison of our algorithm with LSTD and natural actor-critic algorithms.

the robot to reach an unsafe region in the future. We define the affinity score of a state $\mathbf{x} \in \mathbb{X}$ as

$$\mathsf{affinity}(\mathbf{x}) = -\min_{\mathbf{y}\in\mathbb{X}_G} d(\mathbf{x}, \mathbf{y})$$

which is the negative of the minimum number of transitions from $\mathbf{x}$ to any goal state. The state feature is defined to be

$$\mathbf{f}_{\mathsf{S}}(\mathbf{x}) = [\mathsf{safety}(\mathbf{x}), \mathsf{affinity}(\mathbf{x})],$$

and the state-action feature $\mathbf{f}_{\mathsf{S}A}(\mathbf{x}, u)$ is calculated using (48). In this application, we use the following Boltzmann distribution.

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(u|\mathbf{x}) = \frac{e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}{\sum_{u\in\mathbb{U}_e(\mathbf{x})} e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}, \tag{51}$$

where T is the temperature. Note that the only difference of (51) with (49) is that (51) restricts to enabled actions.

Again, we compare our algorithm with the LSTD-AC algorithm in [14] and the four algorithms in [23]. We run each algorithm 100 times independently and Fig. 4 shows the comparison of the average reward for the first 100,000 iterations. For all algorithms, the initial $\boldsymbol{\theta}$ is $(0, 5)$ and the temperature $T = 5$. The step-sizes satisfy $\alpha_c = \frac{\alpha_0 \cdot \alpha_c}{\alpha_c + k^{2/3}}$ and $\beta_c = \frac{\beta_0 \cdot \beta_c}{\beta_c + k}$. For LSTD-AC and our algorithm, we set $\alpha_0 = 0.1$, $\alpha_c = 1000$, $\beta_0 = 0.01$ and $\beta_c = 1000$. For BSGL1-BSGL4, we set $\alpha_0 = 0.1$, $\alpha_c = 1000$, $\beta_0 = 0.001$ and $\beta_c = 10000$. We use $\chi_{min} = 30$ in (32).

Table I summarizes the convergence time and the converged reward for all algorithms. Among the three natural gradient-based algorithms, BSGL3 performs the best, but on average it is still slower than our method in this problem. The convergence rate of BSGL1 is much worse than the rest of the algorithms. For this problem, we did not observe numerical instability for BSGL2.

For the robot control problem, the average CPU time per iteration is 3281 μs for our algorithm vs. 2837 μs for LSTD-AC, that is, about 15.7% higher. The computational overhead of the second-order critic in this problem is much lower than in the GARNET problem, which is due to the fact that the robot control problem has less parameters.

The CPU time per iteration of both LSTD-AC and our algorithm is larger than that of BSGL1-BSGL4, but the difference is much smaller compared with the GARNET problem. Since significant less iterations are needed for our algorithm, it converges faster than all other algorithms. Specifically, the second-best algorithm, BSGL3, takes on average 20.3% more time to converge.

VIII. Conclusions and Future Work

In this paper we propose a general estimate for the Hessian matrix of the long-run reward in actor-critic algorithms. Based on this estimate, we present a novel second-order actor-critic algorithm which uses second-order critic and actor. The actor, in particular, uses a direct estimate of the Hessian matrix to improve the rate of convergence for ill-conditioned problems. Building on the LSTD-AC algorithm in [16], [14], our algorithm extends the *critic* to approximate the Hessian and revises the *actor* to update the policy parameters using Newton's method. We compare our algorithm with the LSTD-AC algorithm and the four algorithms in [23], three of which are based on natural gradients, in two applications. The results show that our method can achieve a better rate of convergence for many problems.

As a variant of Newton's method, our method has similar limitations. First, the cost of maintaining a Hessian estimate is quadratic to the number of parameters. As a result, our algorithm is only suitable for problems with relatively small number of parameters. Second, our algorithm requires the second derivative of the policy function, which implies that the method can not be applied if the policy function is not twice differentiable or its second-order derivatives are hard to obtain. Our algorithm is suitable for the cases where the reward is more sensitive to some parameters vs. others, leading to potentially ill-conditioned problems that are best handled by Newton's method.

One direction for future work is to use part of (9) rather than all four terms, so as to achieve a better tradeoff between convergence rate and computational cost per iteration. In addition, the algorithm described in this paper is suitable for the average reward problem. Since Theorem IV.2 holds for all three types of rewards, similar algorithms can be derived for the discounted and the total reward cases.

Acknowledgment

The authors would like to thank Weicong Ding for useful discussions relating to the proof of critic convergence.

Appendix A
Proof of Lemma VI.1

Lemma A.1: Suppose $\{\gamma_k\}$, $\{\zeta_k\}$, $\{\beta_k\}$ are three deterministic positive sequences that satisfy (36) for some $d_1, d_2 > 0$. Then,

$$\sum_k (\max(\gamma_k, \beta_k)/\zeta_k)^d < \infty \quad \text{for some d} > 0.$$

Proof: Note that $\lim_k(\gamma_k/\zeta_k) = 0$ and $\lim_k(\beta_k/\zeta_k) = 0$. Letting $d > \max(d_1, d_2)$, it follows $\sum_k(\gamma_k/\zeta_k)^d < \infty$ and $\sum_k(\beta_k/\zeta_k)^d < \infty$. Further,

$$\begin{aligned}\sum_k (\max(\gamma_k, \beta_k)/\zeta_k)^d &= \sum_k (\max(\gamma_k/\zeta_k, \beta_k/\zeta_k))^d \\ &= \sum_k \max((\gamma_k/\zeta_k)^d, (\beta_k/\zeta_k)^d) \\ &\le \sum_k (\gamma_k/\zeta_k)^d + \sum_k (\beta_k/\zeta_k)^d \\ &< \infty.\end{aligned}$$

The second equality is due to the function $f(x) = x^d$ being monotonically increasing in the range $[0, \infty)$ when $d > 0$. The first inequality follows because both $\{(\gamma_k/\zeta_k)^d\}$ and $\{(\beta_k/\zeta_k)^d\}$ are positive sequences. ■

A. Proof of Lemma VI.1:

Proof: Define $\hat{\boldsymbol{\theta}}_k = (\boldsymbol{\theta}_k, \mathbf{r}_k)$ to be the collection of all parameters in (39). We can write (39) as

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \zeta_k(\mathbf{h}_{\hat{\boldsymbol{\theta}}_k}(\mathbf{y}_k) - \mathbf{G}_{\hat{\boldsymbol{\theta}}_k}(\mathbf{y}_k)\mathbf{s}_k) + \zeta_k \boldsymbol{\Xi}_k \mathbf{s}_k. \quad (52)$$

We have

$$\begin{aligned}\|\hat{\boldsymbol{\theta}}_{k+1} - \hat{\boldsymbol{\theta}}_k\| &\le \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\| + \|\mathbf{r}_{k+1} - \mathbf{r}_k\| \\ &\le \beta_k F_k + \gamma_k F_k^r \\ &\le \max(\beta_k, \gamma_k)(F_k + F_k^r).\end{aligned}$$

The last inequality is implied since $\beta_k > 0$, $\gamma_k > 0$, F_k and F_k^r are nonnegative processes. Combined with Lemma A.1, we can see Assumptions 3.1.(1–3) in [12] are satisfied. In addition, Assumptions 3.1.(4–10) in [12] are satisfied due to Assumptions A.(3–11). As a result, Thm. 3.2 in [12] holds and implies

$$\lim_k \|\bar{\mathbf{G}}(\hat{\boldsymbol{\theta}}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\hat{\boldsymbol{\theta}}_k)\| = 0, \qquad \text{w.p.1.} \quad (53)$$

The left hand side of (53) is equivalent to the left hand side of the lemma. ■

Appendix B
Proof of Theorem VI.6

We first present the following lemmas. We define the norm $\|\cdot\|$ of a matrix to be the norm of the column vector containing all of its elements.

Lemma B.1: Under iteration (27), we have

$$\begin{aligned}\|\mathbf{A}_{k+1} - \mathbf{A}_k\| &\le \gamma_k F_k^A, \\ \|\mathbf{b}_{k+1} - \mathbf{b}_k\| &\le \gamma_k F_k^b,\end{aligned}$$

for some processes $\{F_k^A\}$ and $\{F_k^b\}$ with bounded moments, where γ_k is the stepsize in (27).

Proof: According to (27), we have

$$\begin{aligned}&\mathbf{A}_{k+1} - \mathbf{A}_k \\ &\quad = \gamma_k \left(\mathbf{z}_k(\boldsymbol{\phi}'_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k) - \boldsymbol{\phi}'_{\boldsymbol{\theta}_{k+1}}(\mathbf{x}_{k+1}, u_{k+1})) - \mathbf{A}_k\right).\end{aligned}$$

Similar to Lemma VI.5 and because $\mathbf{z}_k$ has bounded moments and $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, it can be verified that $\mathbf{A}_k$ has bounded moments. This establishes the first statement of the Lemma. We can prove the second statement of the Lemma for $\{\mathbf{b}_k\}$ in the same way given that the one-step reward function $g \in \mathcal{D}^{(2)}$, first by establishing that α_k has bounded moments. ■

Lemma B.2: Suppose $\mathbf{f}(\cdot)$ is a *locally Lipschitz continuous* function on a domain $\mathcal{D}$. Let $\{\boldsymbol{v}_k\}$ be a sequence of random variables with bounded moments defined on $\mathcal{D}$ such that $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| \le \gamma_k F_k$ for some $\{F_k\}$ with bounded moments w.p.1. Then $\|\mathbf{f}(\boldsymbol{v}_{k+1}) - \mathbf{f}(\boldsymbol{v}_k)\| \le \gamma_k F_k^f$ for some $\{F_k^f\}$ with bounded moments w.p.1.

Proof: Since $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| \le \gamma_k F_k$, it follows $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k| < \infty$ w.p.1. Since $\{\boldsymbol{v}_k\}$ has bounded moments, $\boldsymbol{v}_k$ must be in a compact set w.p.1 for $\forall k$. Then, by Lipschitz continuity, $\|\mathbf{f}(\boldsymbol{v}_{k+1}) - \mathbf{f}(\boldsymbol{v}_k)\| \le C\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| \le \gamma_k C F_k$ for some constant C. The lemma can be proved by letting $F_k^f = CF_k$. ■

Lemma B.3: Let $\boldsymbol{v} = \{\mathbf{A}, \mathbf{b}\}$ be a vector consisting of all elements in an $m \times m$ matrix $\mathbf{A}$ and a vector $\mathbf{b} \in \mathbb{R}^m$. The function $\mathbf{f}(\boldsymbol{v}) = \mathbf{A}^{-1}\mathbf{b}$ is *locally Lipschitz continuous* with respect to $\mathbf{A}$ and $\mathbf{b}$ on the domain $\mathcal{D} = \{\boldsymbol{v} : \det(\mathbf{A}) \ge \epsilon\}$, where ϵ is a positive constant.

Proof: Let $\mathbf{A}^a$ denote the adjoint matrix of $\mathbf{A}$. The function $\mathbf{f}^a(\boldsymbol{v}) = \mathbf{A}^a\mathbf{b}$ is locally Lipschitz continuous as it is a polynomial function, so $\|\mathbf{f}^a(\boldsymbol{v}_1) - \mathbf{f}^a(\boldsymbol{v}_2)\| \le C\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|$ for some constant C and $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ that belong to a compact set. Since $\mathbf{A}^{-1} = \mathbf{A}^a/\det(\mathbf{A})$ and for $\boldsymbol{v}_1 = \{\mathbf{A}_1, \mathbf{b}_1\}$, $\boldsymbol{v}_2 = \{\mathbf{A}_2, \mathbf{b}_2\}$, we have

$$\begin{aligned}\|\mathbf{f}(\boldsymbol{v}_1) - \mathbf{f}(\boldsymbol{v}_2)\| &= \|\mathbf{A}_1^{-1}\mathbf{b}_1 - \mathbf{A}_2^{-1}\mathbf{b}_2\| \\ &= \|\mathbf{A}_1^a\mathbf{b}_1/\det(\mathbf{A}_1) - \mathbf{A}_2^a\mathbf{b}_2/\det(\mathbf{A}_2)\| \\ &\le \frac{1}{\epsilon}\|\mathbf{A}_1^a\mathbf{b}_1 - \mathbf{A}_2^a\mathbf{b}_2\| \\ &= \frac{1}{\epsilon}\|\mathbf{f}^a(\boldsymbol{v}_1) - \mathbf{f}^a(\boldsymbol{v}_2)\| \\ &\le \frac{C}{\epsilon}\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|.\end{aligned}$$

So $\mathbf{f}(\boldsymbol{v}) = \mathbf{A}^{-1}\mathbf{b}$ must be locally Lipschitz continuous on the domain $\mathcal{D} = \{\boldsymbol{v} : \det(\mathbf{A}) > \epsilon\}$. ■

A. Proof of Theorem VI.6

Proof: Recall that $\mathbf{V}(\mathbf{A})$ is the column vector stacking all columns in a matrix $\mathbf{A}$. Let $\boldsymbol{v}_k = (\mathbf{V}(\mathbf{A}_k), \mathbf{b}_k)$ where $\mathbf{A}_k$ and $\mathbf{b}_k$ are the iterates in (27). It follows

$$\begin{aligned}\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| &= \|\mathbf{A}_{k+1} - \mathbf{A}_k\| + \|\mathbf{b}_{k+1} - \mathbf{b}_k\| \\ &\le \gamma_k(F_k^A + F_k^b).\end{aligned}$$

The last equality is due to Lemma B.1 and $F_k^A + F_k^b$ has bounded moments. Define the function $\mathbf{f}(\boldsymbol{v}_k) = \mathbf{A}_k^{-1}\mathbf{b}_k$, which implies $\mathbf{r}_k = \mathbf{f}(\mathbf{x}_k) = \mathbf{A}_k^{-1}\mathbf{b}_k$ when $\det(\mathbf{A}_k) \ge \epsilon$ by (28). The lemma can be easily proved by combining Lemma B.3 and Lemma B.2. ■

References

[1] J. Wang and I. C. Paschalidis, "A Hessian actor-critic algorithm," in *Proc. 53rd IEEE Conf. Decision Control*, Los Angeles, CA, Dec. 2014, pp. 1131–1136.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[3] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Nashua, NH: Athena Scientific, 1996.

[4] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[5] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006.

[6] M. Khamassi, L. Lachèze, B. Girard, A. Berthoz, and A. Guillot, "Actor-critic models of reinforcement learning in the basal ganglia: From natural to artificial rats," *Adaptive Behavior*, vol. 13, no. 2, pp. 131–148, 2005.

[7] K. Samejima and T. Omori, "Adaptive internal state space construction method for reinforcement learning of a real-world agent," *Neural Networks*, vol. 12, pp. 1143–1155, 1999.

[8] G. Gajjar, S. Khaparde, P. Nagaraju, and S. Soman, "Application of actor-critic learning algorithm for optimal bidding problem of a GenCo," *IEEE Trans. Power Eng. Rev.*, vol. 18, no. 1, pp. 11–18, 2003.

[9] D. Bertsekas, V. Borkar, and A. Nedic, "Improved temporal difference methods with linear function approximation," *LIDS REPORT*, Tech. Rep. 2573, 2003.

[10] J. Boyan, "Least-squares temporal difference learning," in *Proc. 16th Int. Conf. Machine Learning*, 1999.

[11] S. Bradtke and A. Barto, "Linear least-squares algorithms for temporal difference learning," *Machine Learning*, vol. 22, no. 2, pp. 33–57, 1996.

[12] V. Konda, *Actor-Critic Algorihms*, Ph.D. dissertation, MIT, Cambridge, MA, 2002.

[13] A. Nedic and D. Bertsekas, "Least squares policy evaluation algorithms with linear function approximation," *Discrete Event Dynamic Syst.: Theory Appl.*, vol. 13, pp. 79–110, 2003.

[14] R. Moazzez-Estanjini, K. Li, and I. C. Paschalidis, "A least squares temporal difference actor-critic algorithm with applications to warehouse management," *Naval Research Logistics*, vol. 59, no. 3, pp. 197–211, 2012.

[15] X.-C. Ding, J. Wang, M. Lahijanian, I. C. Paschalidis, and C. Belta, "Temporal logic motion control using actor-critic methods," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, St. Paul, MN, May 14–18 2012, pp. 4687–4692.

[16] R. Moazzez-Estanjini, X.-C. Ding, M. Lahijanian, J. Wang, C. A. Belta, and I. C. Paschalidis, "Least squares temporal difference actor-critic methods with applications to robot motion control," in *Proc. 50th IEEE Conf. Decision Control*, Orlando, FL, Dec. 12–15 2011.

[17] J. Wang, X. Ding, M. Lahijanian, I. C. Paschalidis, and C. Belta, "Temporal logic motion control using actor-critic methods," *Int. J. Robot. Research*, vol. 34, no. 10, pp. 1329–1344, 2015.

[18] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[19] S. Kakade, "A natural policy gradient," *Advances Neural Inform. Processing Syst.*, vol. 14, pp. 1531–1538, 2001.

[20] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, pp. 1180–1190, 2008.

[21] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee, "Incremental natural actor-critic algorithms," *Neural Information Processing Systems (NIPS)*, 2007 .

[22] S. Richter, D. Aberdeen, and J. Yu, "Natural actor-critic for road traffic optimisation," in *Advances in Neural Information Processing Systems*, 2007, p. 1169.

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

[24] S. Kakade, "Optimizing average reward using discounted rewards," in *Computational Learning Theory*. Springer, 2001, pp. 605–615.

[25] N. Le Roux and A. Fitzgibbon, "A fast natural Newton method," in *Proc. 27th Int. Conf. Machine Learning*. Citeseer, 2010.

[26] S. P. Meyn, R. L. Tweedie, and P. W. Glynn, *Markov Chains and Stochastic Stability*. Cambridge University Press Cambridge, MA, 2009, vol. 2.

[27] P. Marbach and J. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Trans. Autom. Control*, vol. 46, pp. 191–209, 2001.

[28] I. C. Paschalidis, K. Li, and R. M. Estanjini, "An actor-critic method using least squares temporal difference learning," in *Proc. Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 2564–2569.

[29] H. Yu, "*Approximate Solution Methods for Partially Observable Markov and Semi-Markov Decision Processes*," Ph.D. dissertation, 2006.

[30] H. Yu and D. P. Bertsekas, "Convergence results for some temporal difference methods based on least squares," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1515–1531, Jul. 2009.

[31] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. Springer, 2008.

[32] V. R. Konda and J. N. Tsitsiklis, "Appendix to "on actor-critic algorithms"." [Online]. Available: http://web.mit.edu/jnt/www/Papers/J094-03-kon-actors-app.pdf.

Jing (Conan) Wang received the B.E. degree in electrical and information engineering from Huazhong University of Science and Technology, Huazhong, China, in 2010 and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2015.

He is now a Software Engineer at Google, Inc. His research interests include interest-based advertising, cyber security, and approximate dynamic programming.

Ioannis Ch. Paschalidis (M'96–SM'06–F'14) received the M.S. and Ph.D. degrees both in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1993 and 1996, respectively.

In September 1996 he joined Boston University where he has been ever since. He is a Professor and Distinguished Faculty Fellow at Boston University with appointments in the Department of Electrical and Computer Engineering, the Division of Systems Engineering, and the Department of Biomedical Engineering. He is the Director of the Center for Information and Systems Engineering (CISE). He has held visiting appointments with MIT and Columbia University, New York, NY, USA. His current research interests lie in the fields of systems and control, networking, applied probability, optimization, operations research, computational biology, and medical informatics.

Dr. Paschalidis received the NSF CAREER award (2000), several best paper and best algorithmic performance awards, and a 2014 IBM/IEEE Smarter Planet Challenge Award. He was an invited participant at the 2002 Frontiers of Engineering Symposium, organized by the U.S. National Academy of Engineering and the 2014 U.S. National Academies Keck Futures Initiative (NAFKI) Conference. He is the inaugural Editor-in-Chief of the IEEE Transactions on Control of Network Systems.

An Actor-Critic Algorithm With Second-Order Actor and Critic

Jing Wang and Ioannis Ch. Paschalidis, *Fellow, IEEE*

***Abstract*—Actor-critic algorithms solve dynamic decision making problems by optimizing a performance metric of interest over a user-specified parametric class of policies. They employ a combination of an actor, making policy improvement steps, and a critic, computing policy improvement directions. Many existing algorithms use a steepest ascent method to improve the policy, which is known to suffer from slow convergence for ill-conditioned problems. In this paper, we first develop an estimate of the (Hessian) matrix containing the second derivatives of the performance metric with respect to policy parameters. Using this estimate, we introduce a new second-order policy improvement method and couple it with a critic using a second-order learning method. We establish almost sure convergence of the new method to a neighborhood of a policy parameter stationary point. We compare the new algorithm with some existing algorithms in two application and demonstrate that it leads to significantly faster convergence.**

***Index Terms*—Actor-critic algorithms, Markov decision processes, Newton's method, robotics.**

I. Introduction

MARKOV Decision Processes (MDPs) provide a general framework for sequential decision making problems. Although MDPs can be solved using *dynamic programming*, the well-known "curse of dimensionality" becomes an impediment for larger instances [1]. In addition, *dynamic programming* in a standard implementation requires explicit transition probabilities among states under each control, which are not available for many applications. To address these limitations, a number of *approximate dynamic programming* techniques have been developed, including *reinforcement learning* methods [2], a variety of techniques involving value function and policy approximations (*neuro-dynamic programming* [3]) and *actor-critic algorithms* [4].

Manuscript received April 16, 2016; accepted September 23, 2016. Date of publication; date of current version. This paper appeared in part at the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, 2014. This work was supported in part by the NSF under grants CNS-1239021, CCF-1527292, and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952. Recommended by Associate Editor Q.-S. Jia.

J. Wang is with the Center for Information & Systems Engineering, Boston University (e-mail: wangjing@bu.edu).

I. Ch. Paschalidis is with the Department of Electrical and Computer Engineering and Division of Systems Engineering, Boston University8 St. Mary's St., Boston, MA 02215, (e-mail: yannisp@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2016.2616384

This paper focuses on the latter *actor-critic algorithms*. They optimize a parametric user-designed *Randomized Stationary Policy* (RSP) using policy gradient estimation. RSPs are policies parameterized by a parsimonious set of parameters. To optimize the RSPs with respect to these parameters, *actor-critic algorithms* estimate policy gradients using learning methods that are much more efficient than computing a cost-to-go function over the entire state-action space. Many different variants of *actor-critic algorithms* have been proposed and shown to be effective for many applications such as robotics [5], biology [6], navigation [7], and optimal bidding for electricity generation [8].

In an attractive type of an *actor-critic algorithm* introduced in [4], a critic is used to estimate the policy gradient from observations on a single sample path and an actor is using this gradient to update the policy at a slower time-scale [4]. The estimate of the critic tracks the slowly-varying policy asymptotically, using first-order variants of the *Temporal Difference (TD)* learning algorithms (TD(1) and TD(λ)). However, it has been shown that second-order learning methods—Least Squares TD (LSTD)—are superior in terms of *rate of convergence* (see [9]–[14]). LSTD was first proposed for discounted cost problems in [11] and was shown to have the optimal *rate of convergence* in [12]. In [14], LSTD is used in the critic of an actor-critic algorithm, resulting in the LSTD Actor-Critic algorithm (LSTD-AC). Later, this algorithm was applied to applications of robot motion control with temporal specifications [15]–[17]. Despite faster convergence than TD-based methods, LSTD-AC exhibits slow convergence for ill-conditioned problems in which the performance metric is more sensitive to some parameters in the RSPs than others. The reason is that it uses a first order actor with an "unscaled" gradient, commonly known as steepest ascent, to update the policy. This often leads to a "zig-zagging" behavior in order to converge to a stationary point.

Several algorithms have been introduced which use a second-order method in the actor. The "natural" gradient method was originally proposed for stochastic learning [18], [19]. [20] proposed a different estimate of the natural gradient but its accuracy can be influenced by the choice of basis functions; an episodic algorithm was then proposed to guarantee the unbiasedness of the estimate. These methods use the inverse of the Fisher information matrix to scale the gradient. [21] suggested several incremental methods using the natural policy gradient. [22] presented an online natural actor-critic algorithm using a natural gradient and applied it to a road traffic optimization problem. Based on [20], [23] proposes three fully incremental natural actor-critic algorithms. It also describes a method that is based on a "vanilla" gradient and provides extensive empirical comparison of all algorithms in test problems (so called *Generic Average Reward Non-stationary Environment Testbed—GARNET* problems [23]).

Although natural gradients are very effective in stochastic learning, there are alternative ways to scale gradients. The Hessian matrix of the performance metric with respect to the parameters is commonly used to improve the *rate of convergence*. [24] proposes an estimate of the Hessian matrix for a discounted reward problem using a sample path of an MDP. Although the relationship between the Fisher information matrix and the Hessian matrix has been briefly discussed in [19] and [25], it is still not fully clear how they are related in the actor-critic framework and why natural actor-critic algorithms work well in practice.

In this work, we develop a more general estimate of the Hessian matrix for actor-critic algorithms. In Section V-C, we demonstrate that our Hessian estimate degenerates to the Fisher information matrix used in natural actor-critic algorithms if we assume no knowledge of the state-action value function and ignore second derivatives with respect to the parameter vector. In this light, natural actor-critic algorithms can be seen as equivalent to quasi-Newton methods that assume no knowledge of the state-action value function when approximating the Hessian matrix. In fact, [12] proposes a quasi-Newton actor-critic algorithm that is very similar to the methods in [20].

This paper proposes a method that uses LSTD-based critics to provide estimates of both the gradient and the Hessian and utilizes the Hessian estimate in the *actor* to update policy parameters.

We establish *almost sure* convergence in the neighborhood of a stationary point (with respect to policy parameters) of the performance metric. We remark that a subset of the results appeared in a preliminary conference paper in [1]. The present paper contains all proofs concerning the Hessian estimate, the convergence analysis which was absent from [1], and a much more extensive numerical evaluation of our method both in GARNET problems and in an application from robotics.

The remainder of the paper is organized as follows: Section II provides background on MDPs and establishes some of our notation. Section III presents the estimation of the policy gradient. Section IV develops the estimate of the policy Hessian, which is the foundation of the new algorithm. Section V describes our method and Section VI proves its convergence. Section VII presents two case studies.

Notation: Bold letters are used to denote vectors and matrices; typically vectors are lower case and matrices upper case. Vectors are column vectors, unless explicitly stated otherwise. Prime denotes transpose. For the column vector $\mathbf{x} \in \mathbb{R}^n$ we write $\mathbf{x} = (x_1, \ldots, x_n)$ for economy of space, while $\|\mathbf{x}\|$ denotes the Euclidean norm. The expressions $\succ 0$ and $\succeq 0$ denote positive-definiteness and positive-semi-definiteness, respectively. Vectors or matrices with all zeroes are written as $\mathbf{0}$ and the identity matrix as $\mathbf{I}$. For any set $\mathscr{S}$, $|\mathscr{S}|$ denotes its cardinality. $\boldsymbol{\theta}$ denotes the parameters in parameterized policies. If not explicitly specified, ∇ and ∇^2 denote the gradient and Hessian w.r.t. $\boldsymbol{\theta}$. To simplify the notation, a lot of equations in this paper are represented using functional notation and the domain of these functions is assumed to be $\mathbb{X} \times \mathbb{U}$, where $\mathbb{X}$ and $\mathbb{U}$ are the state and the action space, respectively, of the MDP. Vector-valued functions are denoted using bold letters while scalar-valued functions are denoted using normal letters. $\underline{0}$ and $\underline{1}$ are functions that assign the value 0 and 1 to all state-action pairs, respectively.

II. Markov Decision Processes

Consider a discrete-time *Markov Decision Process (MDP)* with a finite state space $\mathbb{X}$ and an action space $\mathbb{U}$. Let $\mathbf{x}_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$ be the state of the system and the action taken at time k, respectively. Let $g(\mathbf{x}_k, u_k)$ be the one-step reward of applying action u_k when the system is at state $\mathbf{x}_k$. We will use $\mathbf{x}_0$ to denote the initial state and $p(\mathbf{x}_{k+1}|\mathbf{x}_k, u_k)$ for the state transition probabilities, which are typically not explicitly known. We assume that $\{\mathbf{x}_k\}$ and $\{\mathbf{x}_k, u_k\}$ are ergodic Markov chains [12].

This paper considers policies that belong to a parameterized family of RSPs $\{\mu_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^n\}$. That is, given a state $\mathbf{x} \in \mathbb{X}$ and an n-dimensional parameter vector $\boldsymbol{\theta}$, the policy applies action $u \in \mathbb{U}$ with probability $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$. Given a fixed policy $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$, the history of $g(\mathbf{x}_k, u_k)$ can be represented by a random process. Let $E_{\boldsymbol{\theta}}\{\cdot\}$ be the expectation with respect to this random process; the long-term average reward for a policy $\mu_{\boldsymbol{\theta}}$ is $\bar{\alpha}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\{\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} [g(\mathbf{x}_k, u_k)]\}$.

In average reward MDP optimization problems, the performance metric is the long-term average reward $\bar{\alpha}(\boldsymbol{\theta})$ and the objective is to optimize $\bar{\alpha}(\boldsymbol{\theta})$. Similar problems can be defined by using discounted reward or total reward as performance metrics [12]. Note that the discounted reward and the total reward can be treated as the average reward of an artificial MDP (See Chapter 2 of [12]). Without loss of generality, this paper focuses on the average reward case. Corresponding results for the other cases can be obtained with modifications similar to Sec. 2.4 and 2.5 of [12].

III. Estimation of Policy Gradient

The *state-action value function* $Q_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ (sometimes referred to as the Q-value function) of a policy $\mu_{\boldsymbol{\theta}}$ is defined as the expected future reward given the current state $\mathbf{x}$ and the action u. $Q_{\boldsymbol{\theta}}$ is the unique solution of the Poisson equation with parameter $\boldsymbol{\theta}$ [26], [12] (written as a functional relationship)

$$Q_{\boldsymbol{\theta}} = g - \bar{\alpha}(\boldsymbol{\theta})\underline{1} + P_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}, \tag{1}$$

where $P_{\boldsymbol{\theta}}$ is the operator of taking expectation after one transition. More precisely, for any real-valued or vector-valued function f defined on $\mathbb{X} \times \mathbb{U}$,

$$(P_{\boldsymbol{\theta}} f)(\mathbf{x}, u) = \sum_{\mathbf{y},\nu} p(\mathbf{y}|\mathbf{x}, u)\mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) f(\mathbf{y}, \nu) \tag{2}$$

for all $(\mathbf{x}, u) \in \mathbb{X} \times \mathbb{U}$.

Let now

$$\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x}), \tag{3}$$

where $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \mathbf{0}$ when $\mathbf{x}, u$ are such that $\mu_{\boldsymbol{\theta}}(u|\mathbf{x}) \equiv 0$ for all $\boldsymbol{\theta}$'s. It is assumed that $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is bounded and continuously differentiable. Since $\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$ is the probability of action u at state $\mathbf{x}$ for $\boldsymbol{\theta}$, $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is the gradient of the *log-likelihood* $\ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})$. We write $\boldsymbol{\psi}_{\boldsymbol{\theta}} = (\psi^1_{\boldsymbol{\theta}}, \ldots, \psi^n_{\boldsymbol{\theta}})$ where n is the dimensionality of $\boldsymbol{\theta}$.

For each $\boldsymbol{\theta} \in \mathbb{R}^n$, let $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u)$ be the stationary probability of state-action pair $(\mathbf{x}, u)$ in the Markov chain $\{\mathbf{x}_k, u_k\}$. For any $\boldsymbol{\theta} \in \mathbb{R}^n$, we define the inner product operator $\langle \cdot, \cdot \rangle_{\boldsymbol{\theta}}$ of two real-valued or vector-valued functions Q_1, Q_2 on $\mathbb{X} \times \mathbb{U}$ by

$$\langle Q_1, Q_2 \rangle_{\boldsymbol{\theta}} = \sum_{\mathbf{x}, u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) Q_1(\mathbf{x}, u) Q_2(\mathbf{x}, u). \tag{4}$$

A key fact underlying actor-critic algorithms is that the policy gradient of $\bar{\alpha}(\boldsymbol{\theta})$ can be expressed as [27], [12]

$$\frac{\partial \bar{\alpha}(\boldsymbol{\theta})}{\partial \theta_i} = \left\langle Q_{\boldsymbol{\theta}}, \psi^i_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}, \; i = 1, \ldots, n. \tag{5}$$

IV. Estimation of the Policy Hessian

Earlier work in actor-critic methods has used critics based on TD(1), TD(λ), and LSTD methods to estimate the policy gradient $\nabla \bar{\alpha}(\boldsymbol{\theta})$ [4], [28]. Since we are interested in a Newton-like gradient ascent update in the actor, in this section we develop an estimate for the policy Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$.

Applying the operator ∇ on the real-valued function $g_{\boldsymbol{\theta}}(\mathbf{x}, u)$ parameterized by $\boldsymbol{\theta}$, we obtain a vector-valued function, abbreviated as $\nabla g_{\boldsymbol{\theta}}$, which maps $(\mathbf{x}, u)$ to $\nabla g_{\boldsymbol{\theta}}(\mathbf{x}, u)$. For a vector-valued function $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^m$ parameterized by $\boldsymbol{\theta}$, which can be denoted as $\mathbf{f}_{\boldsymbol{\theta}} = (f^1_{\boldsymbol{\theta}}, \ldots, f^m_{\boldsymbol{\theta}})$, we define $\nabla \mathbf{f}_{\boldsymbol{\theta}}$ to be an $n \times m$ matrix-valued function whose ith column is $\nabla f^i_{\boldsymbol{\theta}}$.

Lemma IV.1: For any vector-valued function $\mathbf{f}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^m$, we have

$$\nabla \left(P_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}} \right) = P_{\boldsymbol{\theta}} \left(\nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right).$$

Proof: For all state-action pairs $(\mathbf{x}, u) \in \mathbb{X} \times \mathbb{U}$, we have

$$\begin{aligned} \nabla (P_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}})(\mathbf{x}, u) = &\; \nabla \left(\sum_{\mathbf{y}, \nu} p(\mathbf{y}|\mathbf{x}, u) \mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \nu) \right) \\ = &\; \sum_{\mathbf{y}, \nu} p(\mathbf{y}|\mathbf{x}, u) \nabla \left(\mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \nu) \right). \end{aligned} \tag{6}$$

In the above, $\mu_{\boldsymbol{\theta}}(\nu|\mathbf{y}) \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}, \nu)$ is a function defined on $\mathbb{X} \times \mathbb{U}$, which is abbreviated as $\mu_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}}$. Using the chain rule and the definition of $\boldsymbol{\psi}_{\boldsymbol{\theta}}$, we obtain

$$\begin{aligned} \nabla \left(\mu_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}} \right) &= \mu_{\boldsymbol{\theta}} \nabla \mathbf{f}_{\boldsymbol{\theta}} + \nabla \mu_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \\ &= \mu_{\boldsymbol{\theta}} \left(\nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right). \end{aligned} \tag{7}$$

The lemma can be proved by substituting (7) to (6). ∎

Lemma IV.1 provides a way to interchange the $P_{\boldsymbol{\theta}}$ and ∇ operators. Similar to the definition of $\boldsymbol{\psi}_{\boldsymbol{\theta}}$, we define

$$\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \nabla^2 \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x}), \tag{8}$$

where $\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\mathbf{x}, u) = \mathbf{0}$ when $\mathbf{x}, u$ are such that $\mu_{\boldsymbol{\theta}}(u|\mathbf{x}) \equiv 0$ for all $\boldsymbol{\theta}$. $\boldsymbol{\varphi}_{\boldsymbol{\theta}}$ is the Hessian matrix of the *log-likelihood* $\ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})$. The following theorem establishes a similar result to (5) for the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$.

Theorem IV.2 (Hessian Matrix of Average Reward): Let $\varphi^{ij}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ be the scalar-valued (i, j)-th component of $\boldsymbol{\varphi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$. The second-order partial derivative of $\bar{\alpha}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ can be represented as:

$$\begin{aligned} \frac{\partial^2 \bar{\alpha}(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} = &\left\langle Q_{\boldsymbol{\theta}}, \psi^i_{\boldsymbol{\theta}} \psi^j_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle Q_{\boldsymbol{\theta}}, \varphi^{ij}_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \\ &+ \left\langle \frac{\partial Q_{\boldsymbol{\theta}}}{\partial \theta_i}, \psi^j_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle \frac{\partial Q_{\boldsymbol{\theta}}}{\partial \theta_j}, \psi^i_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \end{aligned} \tag{9}$$

for all $i, j = 1, \ldots, n$, where $\langle \cdot, \cdot \rangle_{\boldsymbol{\theta}}$ is the inner product operator defined in (4).

Proof: Applying the ∇ operator on both sides of (1) and using Lemma IV.1 with $\mathbf{f}_{\boldsymbol{\theta}}$ being the scalar function $Q_{\boldsymbol{\theta}}$, we obtain

$$\nabla \bar{\alpha}(\boldsymbol{\theta}) \underline{1} + \nabla Q_{\boldsymbol{\theta}} = P_{\boldsymbol{\theta}} \left(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + \nabla Q_{\boldsymbol{\theta}} \right). \tag{10}$$

Defining the vector-valued function $\mathbf{f}_{\boldsymbol{\theta}} = \boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + \nabla Q_{\boldsymbol{\theta}}$ and applying again the ∇ operator on both sides of (10), we have

$$\nabla (\nabla \bar{\alpha}(\boldsymbol{\theta}) \underline{1} + \nabla Q_{\boldsymbol{\theta}}) = \nabla (P_{\boldsymbol{\theta}} \mathbf{f}_{\boldsymbol{\theta}}),$$

which due to Lemma IV.1 implies

$$\nabla^2 \bar{\alpha}(\boldsymbol{\theta}) \underline{1} + \nabla^2 Q_{\boldsymbol{\theta}} = P_{\boldsymbol{\theta}} \left(\nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right). \tag{11}$$

Take now the inner product with $\underline{1}$ on both sides of (11) and notice that because $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is the stationary probability under θ, it holds $\langle \underline{1}, \mathbf{h} \rangle_{\boldsymbol{\theta}} = \langle \underline{1}, P_{\boldsymbol{\theta}} \mathbf{h} \rangle_{\boldsymbol{\theta}}$ for any function $\mathbf{h}$ defined on $\mathbb{X} \times \mathbb{U}$. We have

$$\nabla^2 \bar{\alpha}(\boldsymbol{\theta}) + \left\langle \underline{1}, \nabla^2 Q_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} = \left\langle \underline{1}, \nabla \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \mathbf{f}'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}.$$

Using the definition of $\mathbf{f}_{\boldsymbol{\theta}}$ and the fact $\nabla \mathbf{f}_{\boldsymbol{\theta}} = \nabla(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}) + \nabla^2 Q_{\boldsymbol{\theta}}$, we obtain

$$\begin{aligned} \nabla^2 \bar{\alpha}(\boldsymbol{\theta}) + \left\langle \underline{1}, \nabla^2 Q_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} &= \left\langle \underline{1}, \nabla(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}) + \nabla^2 Q_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \\ &+ \left\langle \underline{1}, Q_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}'_{\boldsymbol{\theta}} + \boldsymbol{\psi}_{\boldsymbol{\theta}} \nabla Q'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \end{aligned} \tag{12}$$

Applying the chain rule, noticing that $\nabla \boldsymbol{\psi}_{\boldsymbol{\theta}} = \boldsymbol{\varphi}_{\boldsymbol{\theta}}$, and reorganizing the terms in (12) it follows

$$\begin{aligned} \nabla^2 \bar{\alpha}(\boldsymbol{\theta}) = &\left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\varphi}_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} \\ &+ \left\langle \nabla Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}} + \left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \nabla Q'_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}}. \end{aligned} \tag{13}$$

∎

Corresponding results for the discounted reward and the total reward cases can be derived based on the relationship between these three problems we discussed earlier. Intuitively, the discounted and total rewards can be considered as average rewards in some artificial MDPs. More detailed information about constructing the artificial MDPs is available at Sec. 2.4 and Sec. 2.5 of [12].

Theorem IV.2 states that the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$ can be decomposed into four terms, all of which take the form of inner products. The first two terms are the inner products of the state-action value function $Q_{\boldsymbol{\theta}}$ with $\psi^i_{\boldsymbol{\theta}} \psi^j_{\boldsymbol{\theta}}$ and $\varphi^{ij}_{\boldsymbol{\theta}}$. Because of the similarity between the first two terms and (5), we can use similar techniques as in the LSTD-AC to estimate them.

For the last two terms in (13) we need an estimate of $\nabla Q_{\boldsymbol{\theta}}$. Note that (10) is the counterpart of the Poisson equation (1) for $\nabla Q_{\boldsymbol{\theta}}$, where $P_{\boldsymbol{\theta}}(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}})$ plays the role of the one-step reward. However, this equation can not be directly used to estimate $\nabla Q_{\boldsymbol{\theta}}$ because it is quite hard to obtain $P_{\boldsymbol{\theta}}(\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}})$. To address this problem, we present the following theorem.

Theorem IV.3: Let the function $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^n$ be the solution of the equation

$$\nabla \bar{\alpha}(\boldsymbol{\theta})\underline{1} + \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} = \boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \tag{14}$$

and $\nabla Q_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^n$ be the solution of (10). Then,

$$\left\langle \nabla Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} - \left\langle \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} = -\left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}}. \tag{15}$$

Proof: Applying the $P_{\boldsymbol{\theta}}$ operator on both sides of (14) and using the fact that $P_{\boldsymbol{\theta}}\underline{1} = \underline{1}$, we obtain

$$\nabla \bar{\alpha}(\boldsymbol{\theta})\underline{1} + P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} = P_{\boldsymbol{\theta}}(\psi_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}). \tag{16}$$

Comparing (10) and (16), it follows $P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} = \nabla Q_{\boldsymbol{\theta}}$. As a result,

$$\begin{aligned}
\left\langle \nabla Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} - \left\langle \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} &= \left\langle \nabla Q_{\boldsymbol{\theta}} - \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&= \left\langle P_{\boldsymbol{\theta}} \tilde{\mathbf{Q}}_{\boldsymbol{\theta}} - \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&= \left\langle -\boldsymbol{\psi}_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} + \nabla \bar{\alpha}(\boldsymbol{\theta})\underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&= -\left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} + \nabla \bar{\alpha}(\boldsymbol{\theta}) \left\langle \underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}},
\end{aligned} \tag{17}$$

where the third equality above used (14).

Let now $\pi_{\boldsymbol{\theta}}(\cdot)$ be the stationary probability of the Markov chain $\{\mathbf{x}_k\}$ under RSP $\boldsymbol{\theta}$. Then, $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u) = \pi_{\boldsymbol{\theta}}(\mathbf{x})\mu_{\boldsymbol{\theta}}(u|\mathbf{x})$, and

$$\begin{aligned}
\left\langle \underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} &= \sum_{\mathbf{x},u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)^{'} \\
&= \sum_{\mathbf{x},u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) \nabla \mu_{\boldsymbol{\theta}}(u|\mathbf{x})^{'} / (\mu_{\boldsymbol{\theta}}(u|\mathbf{x})) \\
&= \sum_{\mathbf{x}} \pi_{\boldsymbol{\theta}}(\mathbf{x}) \sum_{u} \nabla \mu_{\boldsymbol{\theta}}(u|\mathbf{x})^{'} \\
&= \mathbf{0},
\end{aligned} \tag{18}$$

where in the second equality we used (3) and the last equality follows from the fact that $\sum_u \mu_{\boldsymbol{\theta}}(u|\mathbf{x}) = 1$ for all $\boldsymbol{\theta}$. Eq. (15) follows by combining (17) and (18). ■

By symmetry to Eq. (15), it also holds that

$$\left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \nabla Q_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} - \left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} = -\left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}}. \tag{19}$$

Substituting (15) and (19) into (13), we obtain a new estimate of the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$ given in the following Corollary.

Corollary IV.4: With $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ being a solution of (14), the Hessian matrix $\nabla^2 \bar{\alpha}(\boldsymbol{\theta})$ can be expressed as:

$$\begin{aligned}
\nabla^2 \bar{\alpha}(\boldsymbol{\theta}) = &\left\langle Q_{\boldsymbol{\theta}}, \boldsymbol{\varphi}_{\boldsymbol{\theta}} - \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} + \left\langle \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}} \\
&+ \left\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{'} \right\rangle_{\boldsymbol{\theta}}.
\end{aligned} \tag{20}$$

A. Function Approximation

We can calculate $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ by solving (1) and (14). However, when $\mathbb{X} \times \mathbb{U}$ is very large, the computational cost becomes prohibitive. This problem can be addressed using *function approximation* techniques. One popular type of function approximation is to express $Q_{\boldsymbol{\theta}}$ and each component of $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ with a linear combination of feature functions. We choose a set of feature functions $\boldsymbol{\phi}_{\boldsymbol{\theta}} = (\psi_{\boldsymbol{\theta}}^i, \varphi_{\boldsymbol{\theta}}^{ij}, \psi_{\boldsymbol{\theta}}^i \psi_{\boldsymbol{\theta}}^j;\ i, j = 1, \ldots, n)$, where $\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is an N-dimensional vector for $\forall \mathbf{x}, u \in \mathbb{X} \times \mathbb{U}$ with $N = (2n^2 + n)$ and n being the dimensionality of $\boldsymbol{\theta}$. Similar to other actor-critic algorithms, the basis functions $\boldsymbol{\phi}_{\boldsymbol{\theta}}$ need to be uniformly linearly independent [4], [12], which can be enforced by choosing a suitable structure of policies. Some additional features can be added depending on the particular application. This added flexibility could be useful in a number of ways as it has been discussed in [4].

Similar to [12], we consider the following linear approximation for $Q_{\boldsymbol{\theta}}$

$$Q_{\boldsymbol{\theta}}^{\mathbf{r}}(\mathbf{x}, u) = \boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}(\mathbf{x}, u)\mathbf{r}, \qquad \mathbf{r} \in \mathbb{R}^N. \tag{21}$$

Let us now view the inner product operator in (4) for real-valued functions in $\mathbb{X} \times \mathbb{U}$ as an inner product between vectors in $\mathbb{R}^{|\mathbb{X}||\mathbb{U}|}$ and denote by $\|\cdot\|_{\boldsymbol{\theta}}$ the induced norm. Also denote by $\Phi_{\boldsymbol{\theta}}$ the low-dimensional subspace spanned by $\boldsymbol{\phi}_{\boldsymbol{\theta}}$. If we define

$$\mathbf{r}^* = \arg\min_{\mathbf{r} \in \mathbb{R}^N} ||Q_{\boldsymbol{\theta}}^{\mathbf{r}} - Q_{\boldsymbol{\theta}}||_{\boldsymbol{\theta}}, \tag{22}$$

then $Q_{\boldsymbol{\theta}}^{\mathbf{r}^*}$ is the projection of $Q_{\boldsymbol{\theta}}$ on $\Phi_{\boldsymbol{\theta}}$. Similar to (2.2) of [4],

$$\begin{aligned}
\left\langle Q_{\boldsymbol{\theta}}^{\mathbf{r}^*}, \psi_{\boldsymbol{\theta}}^i \right\rangle_{\boldsymbol{\theta}} &= \left\langle Q_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}}^i \right\rangle_{\boldsymbol{\theta}}, \\
\left\langle Q_{\boldsymbol{\theta}}^{\mathbf{r}^*}, \varphi_{\boldsymbol{\theta}}^{ij} - \psi_{\boldsymbol{\theta}}^i \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}} &= \left\langle Q_{\boldsymbol{\theta}}, \varphi_{\boldsymbol{\theta}}^{ij} - \psi_{\boldsymbol{\theta}}^i \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}},
\end{aligned} \tag{23}$$

for all $i, j = 1, \ldots, n$.

Define the linear approximation of $\tilde{Q}_{\boldsymbol{\theta}}^i$, the ith component of $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$, as

$$\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i}(\mathbf{x}, u) = \boldsymbol{\phi}_{\boldsymbol{\theta}}^{'}(\mathbf{x}, u)\mathbf{t}^i, \qquad \mathbf{t}^i \in \mathbb{R}^N. \tag{24}$$

Again, for all $i, j = 1, \ldots, n$ and

$$\mathbf{t}^{i*} = \arg\min_{\mathbf{t} \in \mathbb{R}^N} ||\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i} - \tilde{Q}_{\boldsymbol{\theta}}^i||_{\boldsymbol{\theta}}, \tag{25}$$

$\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^{i*}}$ is the projection of $\tilde{Q}_{\boldsymbol{\theta}}^i$ on $\Phi_{\boldsymbol{\theta}}$. Similar to (2.2) of [4], we have

$$\left\langle \tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^{i*}}, \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}} = \left\langle \tilde{Q}_{\boldsymbol{\theta}}^i, \psi_{\boldsymbol{\theta}}^j \right\rangle_{\boldsymbol{\theta}}. \tag{26}$$

Equations (23) and (26) state that the projections of $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ on the low-dimensional space $\Phi_{\boldsymbol{\theta}}$ are sufficient for estimating (20). This reduces the computational cost for obtaining $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ since we only have to compute the relative parsimonious vectors $\mathbf{r}^*$ and $\mathbf{t}^{i*}$, $i = 1, \ldots, n$, while it does not alter the inner products needed to compute the gradient $\nabla\bar{\alpha}(\boldsymbol{\theta})$ (cf. (5)) and the Hessian $\nabla^2\bar{\alpha}(\boldsymbol{\theta})$ (cf. (20)).

V. A Second-Order Actor-Critic Algorithm

A. Critic Step

We use the Least Squares Temporal Difference (LSTD) (see, e.g., [14]) with parameter λ to estimate $\mathbf{r}^*$ and $\mathbf{t}^{i*}$, $i = 1\ldots, n$, defined in (22) and (25), respectively. Recall that $\mathbf{x}_k$ and u_k denote the state and the action of the system at time k, respectively. Let α_k denote an estimate of the average reward at time k. $\mathbf{z}_k \in \mathbb{R}^N$ denotes Sutton's eligibility trace and $\mathbf{A}_k \in \mathbb{R}^{N\times N}$ a sample estimate of the matrix formed by $\mathbf{z}_k(\boldsymbol{\phi}^{'}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k) - \boldsymbol{\phi}^{'}_{\boldsymbol{\theta}_k}(\mathbf{x}_{k+1}, u_{k+1}))$, which can be viewed as a sample observation of the scaled difference of the features between time k and time $k+1$. $\mathbf{b}_k \in \mathbb{R}^N$ refers to a statistical estimate of the single period relative reward with eligibility trace $\mathbf{z}_k$. Let also use the initial values: $\mathbf{A}_0$ is an identity matrix, α_0 is zero, and $\mathbf{b}_0$ and $\mathbf{z}_0$ are column vectors with all zeros. To estimate $\mathbf{r}^*$, we use the following *Q-critic* update

$$\begin{aligned}\alpha_{k+1} &= \alpha_k + \gamma_k(g(\mathbf{x}_k, u_k) - \alpha_k), \\ \mathbf{z}_{k+1} &= \lambda\mathbf{z}_k + \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k), \\ \mathbf{A}_{k+1} &= \mathbf{A}_k + \gamma_k(\mathbf{z}_k\mathbf{w}_k^{'} - \mathbf{A}_k), \\ \mathbf{b}_{k+1} &= \mathbf{b}_k + \gamma_k[(g(\mathbf{x}_k, u_k) - \alpha_k)\mathbf{z}_k - \mathbf{b}_k],\end{aligned} \tag{27}$$

where $\mathbf{w}_k = \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k) - \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_{k+1}, u_{k+1})$ and γ_k is a stepsize. Let $\mathbf{r}_k$ be the estimate of $\mathbf{r}^*$ at time k; we set

$$\mathbf{r}_{k+1} = \begin{cases}\mathbf{A}_{k+1}^{-1}\mathbf{b}_{k+1}, & \text{if } \det(\mathbf{A}_{k+1}) \geq \epsilon, \\ \mathbf{r}_k, & \text{otherwise,}\end{cases} \tag{28}$$

where ϵ is a small positive constant used to judge whether $\mathbf{A}_{k+1}$ is "ill-conditioned" or not. $\mathbf{A}_k$ should be invertible when k is large enough [29], [30]. Our *Q-critic* (27) is the same with the critic update of the LSTD-AC algorithm in [14] and (28) estimates the same $\mathbf{r}^*$. In addition, we add another critic, named as $\tilde{\mathbf{Q}}$-critic, to estimate $\mathbf{t}^{i*}, \forall i$.

Let now $\mathbf{v}_0^i$, $i = 1, \ldots, n$, be a column vector with all zeros. Let also η_0^i, $i = 1, \ldots, n$, be a scalar set to zero. Notice the relationship between Eq. (1) for the Q-function and Eq. (14) for the $\tilde{\mathbf{Q}}$-function. To estimate $\mathbf{t}^{i*}$, $i = 1, \ldots, n$, defined in (25), we use the following LSTD $\tilde{\mathbf{Q}}$-*critic* update

$$\begin{aligned}\eta_{k+1}^i &= \eta_k^i + \zeta_k(q_k^i - \eta_k^i), \qquad i = 1, \ldots, n, \\ \mathbf{v}_{k+1}^i &= \mathbf{v}_k^i + \zeta_k[(q_k^i - \eta_k^i)\mathbf{z}_k - \mathbf{v}_k^i], \qquad i = 1, \ldots, n,\end{aligned} \tag{29}$$

where $q_k^i = \Gamma(\mathbf{r}_k)\mathbf{r}_k^{'}\boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k)\psi_{\boldsymbol{\theta}_k}^i(\mathbf{x}_k, u_k)$ is an estimate of the ith component of $\boldsymbol{\psi}_{\boldsymbol{\theta}}Q_{\boldsymbol{\theta}}$ which plays the role of the one-step reward in (14). ζ_k is the stepsize of the $\tilde{\mathbf{Q}}$-*critic* and $\Gamma(\mathbf{r}_k)$ is a function that restricts the influence of the error in the estimate $\mathbf{r}_k$. Let $\mathbf{t}_k^i$ be the estimate of $\mathbf{t}^{i*}$ at time k. Similar to the *Q-critic*, we set

$$\mathbf{t}_{k+1}^i = \begin{cases}\mathbf{A}_{k+1}^{-1}\mathbf{v}_{k+1}^i, & \text{if } \det(\mathbf{A}_{k+1}) \geq \epsilon, \\ \mathbf{t}_k^i, & \text{otherwise,}\end{cases} \tag{30}$$

for $i = 1, \ldots, n$. Note that the Sherman-Morrison update of a matrix inverse [22] and the matrix determinant lemma [31] can be applied to reduce the computational cost of calculating $\mathbf{A}_{k+1}^{-1}$ and $\det(\mathbf{A}_{k+1})$ in (28) and (30).

B. Actor Step

Let $Q_{\boldsymbol{\theta}}^{\mathbf{r}}(\mathbf{x}, u) = \Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ and $\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i} = \Gamma(\mathbf{t}^i)\mathbf{t}^{i'}\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ be our estimates for $Q_{\boldsymbol{\theta}}$ and $\tilde{Q}_{\boldsymbol{\theta}}^i$ given $\mathbf{r}$ and $\mathbf{t}^i$, $i = 1, \ldots, n$. As mentioned above, the function $\Gamma(\cdot)$ restricts the influence of the error in $\mathbf{r}$ and $\mathbf{t}^i$, respectively (cf. (21) and (24)). For convenience of notation, let $\mathbf{T} = (\mathbf{t}^1, \ldots, \mathbf{t}^n)$ and denote by $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{\mathbf{T}} = (\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^1}, \ldots, \tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^n})$ a vector-valued function mapping $\mathbb{X} \times \mathbb{U}$ onto $\mathbb{R}^n$ with ith element equal to $\tilde{Q}_{\boldsymbol{\theta}}^{\mathbf{t}^i}$. Motivated by (20) and using just a single sample to estimate the expectation (in a standard stochastic approximation fashion), we also define $\hat{\mathbf{U}}_{\boldsymbol{\theta},\mathbf{r},\mathbf{T}}$ to be an $n \times n$ matrix-valued function defined on $\mathbb{X} \times \mathbb{U}$ and parameterized by $(\boldsymbol{\theta}, \mathbf{r}, \mathbf{T})$ as follows

$$\hat{\mathbf{U}}_{\boldsymbol{\theta},\mathbf{r},\mathbf{T}} = Q_{\boldsymbol{\theta}}^{\mathbf{r}}(\boldsymbol{\varphi}_{\boldsymbol{\theta}} - \boldsymbol{\psi}_{\boldsymbol{\theta}}\boldsymbol{\psi}_{\boldsymbol{\theta}}^{'}) + \tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{\mathbf{T}}\boldsymbol{\psi}_{\boldsymbol{\theta}}^{'} + \boldsymbol{\psi}_{\boldsymbol{\theta}}(\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}^{\mathbf{T}})^{'}. \tag{31}$$

Let $\mathbf{H}_k$ be the estimate of $-\nabla^2\bar{\alpha}(\boldsymbol{\theta})$ at time k with initial condition $\mathbf{H}_0 = \mathbf{I}$. The update rule for $\mathbf{H}_k$ is:

$$\mathbf{H}_{k+1} = \begin{cases}\mathbf{H}_k + \mathbf{U}_k, & \text{if } \mathbf{U}_k \succ 0, \\ \mathbf{H}_k, & \text{otherwise,}\end{cases} \tag{32}$$

where $\mathbf{U}_k = -\hat{\mathbf{U}}_{\boldsymbol{\theta}_k,\mathbf{r}_k,\mathbf{T}_k}(\mathbf{x}_k, u_k)$. Note that $\mathbf{H}_k \succ 0$ because it is updated only when $\mathbf{U}_k \succ 0$. Let χ_k be the number of times the top branch in (32) is executed by iteration k and define

$$\hat{\mathbf{H}}_k = \begin{cases}\mathbf{I}, & \text{if } \chi_k < \chi_{\min}, \\ \mathbf{H}_k, & \text{otherwise,}\end{cases} \tag{33}$$

which will be used to avoid a noisy estimate in the initial updates. The actor update takes the form:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \beta_k\Gamma(\mathbf{r}_k)\mathbf{r}_k^{'}\boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k)\hat{\mathbf{H}}_k^{-1}\boldsymbol{\psi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k), \tag{34}$$

where β_k is a stepsize.

In the update (32), we make sure that our scaling matrix is always positive definite. Notice that $\mathbf{H}_k$ is the estimate of the negative Hessian matrix because we are dealing with a maximization problem. In particular, the Hessian matrix will generally be negative definite in the vicinity of a local maximum and we expect that the upper branch of the update (32) will be used as we approach such a point. The iteration (34) takes a scaled gradient ascent step, with the scaling matrix being positive definite.

The sequences $\{\gamma_k\}$ and $\{\zeta_k\}$ correspond to the stepsizes used by the critics, while β_k and $\Gamma(\mathbf{r}_k)$ control the stepsize for the actor. The function $\Gamma(\mathbf{r}_k)$ is selected such that for some positive constants $C_1 < C_2$:

$$\begin{aligned}&\|\mathbf{r}\|\Gamma(\mathbf{r}) \in [C_1, C_2], \qquad \forall\mathbf{r} \in \mathbb{R}^N, \\ &\|\Gamma(\mathbf{r}) - \Gamma(\hat{\mathbf{r}})\| \leq \frac{C_2\|\mathbf{r} - \hat{\mathbf{r}}\|}{1 + \|\mathbf{r}\| + \|\hat{\mathbf{r}}\|}, \qquad \forall\mathbf{r}, \hat{\mathbf{r}} \in \mathbb{R}^N.\end{aligned} \tag{35}$$

An example that satisfies these requirements is $\Gamma(\mathbf{r}) = \min(1, D/\|\mathbf{r}\|)$ for some positive constant D.

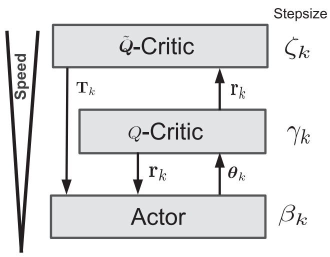

Fig. 1. Relationships between the critics and the actor.

We say a stepsize sequence $\{f_k\}$ is *Square Summable but Not Summable (SSNS)* if $f_k > 0$, $\sum_{k=0}^{\infty} f_k^2 < \infty$ and $\sum_{k=0}^{\infty} f_k = \infty$. For the algorithm to converge, $\{\zeta_k\}$, $\{\gamma_k\}$, and $\{\beta_k\}$ should be SSNS and satisfy

$$\sum_k (\beta_k/\gamma_k)^{d_1} < \infty, \quad \sum_k (\gamma_k/\zeta_k)^{d_2} < \infty, \qquad (36)$$

for some $d_1, d_2 > 0$.

The relationships between the two critics and the actor are shown in Fig. 1. The *Q-critic* and the $\tilde{\mathbf{Q}}$*-critic* generate estimates $\mathbf{r}_k$ and $\mathbf{T}_k = (\mathbf{t}_k^1, \ldots, \mathbf{t}_k^n)$ which yield linear approximations of $Q_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$, respectively. Both critics need to converge faster than the actor in order to track the changes in $\boldsymbol{\theta}$. Moreover, because the observed derivative q_k^i used in the $\tilde{\mathbf{Q}}$*-critic* depends on $\mathbf{r}_k$, the $\tilde{\mathbf{Q}}$*-critic* is updated faster than the *Q-critic* so that it can track changes in $\mathbf{r}_k$. We next present a result establishing a relationship between the stepsize sequences.

Proposition V.1: Suppose $\{\zeta_k\}$ and $\{\beta_k\}$ are two SSNS stepsize sequences that satisfy

$$\sum_k (\beta_k/\zeta_k)^d < \infty, \qquad \text{for some } d > 0. \qquad (37)$$

Let $\gamma_k = (\zeta_k \beta_k)^{1/2}$. Then, $\{\gamma_k\}$ is also SSNS and $\{\gamma_k\}$, $\{\beta_k\}$, $\{\zeta_k\}$ satisfy (36).

Proof: Due to the assumption in (37), $\lim_{k\to\infty}(\beta_k/\zeta_k) = 0$, which implies that there exists a positive constant K such that for $\forall k > K$, $\beta_k \le \zeta_k$. Since $\{\beta_k\}$ is SSNS, it follows

$$\sum_k \gamma_k = \sum_k (\zeta_k \beta_k)^{1/2} \ge C_1 + \sum_{k=K+1}^{\infty} \beta_k = \infty,$$

where $C_1 = \sum_{k=0}^{K} \gamma_k$. Furthermore, since $\{\zeta_k\}$ is SSNS

$$\sum_k \gamma_k^2 = \sum_k \zeta_k \beta_k \le C_2 + \sum_{k=K+1}^{\infty} \zeta_k^2 < \infty,$$

where $C_2 = \sum_{k=0}^{K} \gamma_k^2$. Finally, letting $d_1 = d_2 = 2d$ and due to (37) we have

$$\sum_k (\beta_k/\gamma_k)^{d_1} = \sum_k (\gamma_k/\zeta_k)^{d_2} = \sum_k (\beta_k/\zeta_k)^d < \infty.$$

■

Proposition V.1 simplifies the selection of stepsizes. We just need to select β_k and ζ_k first and let $\gamma_k = (\zeta_k \beta_k)^{1/2}$. An example of $\{\zeta_k\}$, $\{\gamma_k\}$, and $\{\beta_k\}$ that are SSNS and satisfy (36) is: $\zeta_k = 1/k$, $\beta_k = c/(k \ln k)$, where $k > 1$ and $c > 0$, and $\gamma_k = (\zeta_k \beta_k)^{1/2} = (1/k)\sqrt{c/\ln k}$.

C. Relationship With Natural Actor-Critic Algorithms

In our approach, we use the Hessian matrix to scale the gradient in order to improve the convergence rate. A similar idea is to use the Fisher information matrix to scale the gradient. It was first proposed by [19] and several related algorithms followed [20], [23], [21]. This section discusses the relationship of the Fisher information matrix with the Hessian matrix for actor-critic algorithms.

Suppose $\eta_{\boldsymbol{\theta}}(\mathbf{x}, u)$ is the stationary state-action distribution when the RSP parameter equals $\boldsymbol{\theta}$. [20] states that the Fisher information matrix is equal to

$$F_{\boldsymbol{\theta}} = \sum_{\mathbf{x},u} \eta_{\boldsymbol{\theta}}(\mathbf{x}, u) \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x}) \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})', \qquad (38)$$

which can also be written as $\left\langle \underline{1}, \boldsymbol{\psi}_{\boldsymbol{\theta}} \boldsymbol{\psi}_{\boldsymbol{\theta}}' \right\rangle_{\boldsymbol{\theta}}$, where $\boldsymbol{\psi}_{\boldsymbol{\theta}} = \nabla \ln \mu_{\boldsymbol{\theta}}(u|\mathbf{x})$ (cf. (3)).

Let us now compare this expression with the true Hessian matrix (cf. (9)). If we set $Q_\theta \equiv \underline{1}$, hence, $\nabla Q_{\boldsymbol{\theta}} \equiv \underline{0}$, and ignore second derivatives with respect to $\boldsymbol{\theta}$, then the Hessian matrix degenerates to the Fisher information matrix in (38). In this sense, natural actor-critic algorithms are quasi-Newton methods that approximate the Hessian without utilizing the state-action value function $Q_{\boldsymbol{\theta}}$. In contrast, our method takes advantage of the state-action value function.

VI. Convergence

A. Linear Stochastic Approximation Driven by a Slowly Varying Markov Chain

Our Q-critic in (27) has the same form as in [14] so its convergence can be proved in a similar way. In the $\tilde{Q}$-critic (29), the increment q_k^i depends on the parameter vector $\mathbf{r}_k$. To facilitate the convergence proof of the $\tilde{Q}$-critic, this section generalizes the theory of linear stochastic approximation driven by a slowly varying Markov chain developed in [12] to the case where the objective is affected by some additional parameters $\mathbf{r}$.

Let $\{\mathbf{y}_k\}$ be a finite Markov chain whose transition probabilities depend on a parameter $\boldsymbol{\theta} \in \mathbb{R}^n$. Let $\{\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot) : \boldsymbol{\theta} \in \mathbb{R}^n, \mathbf{r} \in \mathbb{R}^N\}$ be a family of m-vector-valued functions parameterized by $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\mathbf{r} \in \mathbb{R}^N$. Let $\boldsymbol{\Xi}_k$ be some $m \times m$ matrix. Consider the following iteration to update a vector $\mathbf{s} \in \mathbb{R}^m$:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \zeta_k(\mathbf{h}_{\boldsymbol{\theta}_k,\mathbf{r}_k}(\mathbf{y}_k) - \mathbf{G}_{\boldsymbol{\theta}_k}(\mathbf{y}_k)\mathbf{s}_k) + \zeta_k \boldsymbol{\Xi}_k \mathbf{s}_k. \quad (39)$$

In the above iteration, $\mathbf{s}_k \in \mathbb{R}^m$ is the approximation vector. $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ and $\mathbf{G}_{\boldsymbol{\theta}}(\cdot)$ are m-vector-valued and $m \times m$-matrix-valued functions parameterized by $\boldsymbol{\theta}, \mathbf{r}$ and $\boldsymbol{\theta}$, respectively. Let $\mathbf{E}[\cdot]$ denote expectation. In order to establish the convergence results, we make the following assumptions.

Assumption A:

1) The sequence $\{\zeta_k\}$ is deterministic, non-increasing and SSNS.

2) The random sequence $\{\boldsymbol{\theta}_k\}$ satisfies $\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\| \leq \beta_k F_k$ for some process $\{F_k\}$ with bounded moments, where $\{\beta_k\}$ is a positive deterministic sequence such that $\sum_k (\beta_k/\zeta_k)^d < \infty$ for some $d > 0$.
3) $\boldsymbol{\Xi}_k$ is an $m \times m$-matrix valued martingale difference with bounded moments.
4) The (random) sequence $\{\mathbf{r}_k\}$ satisfies $\|\mathbf{r}_{k+1} - \mathbf{r}_k\| \leq \gamma_k F_k^r$ for some nonnegative process $\{F_k^r\}$ with bounded moments, where $\{\gamma_k\}$ is a positive sequence such that $\sum_k (\gamma_k/\zeta_k)^d < \infty$ for some $d > 0$.
5) $\mathbf{r}_k$ converges to $\bar{\mathbf{r}}(\boldsymbol{\theta}_k)$ when $k \to \infty$, namely, $\lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\| = 0$, w.p.1.
6) *(Existence of solution to the Poisson Equation.)* For each $\boldsymbol{\theta}$ and $\mathbf{r}$, there exists $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) \in \mathbb{R}^m$, $\bar{\mathbf{G}}(\boldsymbol{\theta}) \in \mathbb{R}^{m\times m}$, and corresponding m-vector and $m \times m$-matrix function $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$, $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\cdot)$ that satisfy the Poisson equation. That is, for each $\mathbf{y}$,

$$\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) = \mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) + (P_{\boldsymbol{\theta}} \hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}})(\mathbf{y}),$$
$$\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{y}) = \mathbf{G}_{\boldsymbol{\theta}}(\mathbf{y}) - \bar{\mathbf{G}}(\boldsymbol{\theta}) + (P_{\boldsymbol{\theta}} \hat{\mathbf{G}}_{\boldsymbol{\theta}})(\mathbf{y}).$$

7) *(Boundedness.)* For all $\boldsymbol{\theta}$ and $\mathbf{r}$, we have $\max(\|\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})\|, \|\bar{\mathbf{G}}(\boldsymbol{\theta})\|) \leq C$ for some constant C.
8) *(Boundedness in expectation.)* For any $d > 0$, there exists $C_d > 0$ such that $\sup_k \mathbf{E}[\|\mathbf{f}_{\boldsymbol{\theta}_k}(\mathbf{y}_k)\|^d] \leq C_d$ and $\sup_k \mathbf{E}[\|\mathbf{g}_{\boldsymbol{\theta}_k,\mathbf{r}_k}(\mathbf{y}_k)\|^d] \leq C_d$, where $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$ represents $\mathbf{G}_{\boldsymbol{\theta}}(\cdot)$ and $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\cdot)$, and $\mathbf{g}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ represents $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ and $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$.
9) *(Lipschitz continuity.)* For some constant $C > 0$, and for all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$, $\|\bar{\mathbf{G}}(\boldsymbol{\theta}) - \bar{\mathbf{G}}(\bar{\boldsymbol{\theta}})\| \leq C\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|$. For all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ and $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, $\|\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) - \bar{\mathbf{h}}(\bar{\boldsymbol{\theta}}, \bar{\mathbf{r}})\| \leq C(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|)$.
10) *(Lipschitz continuity in expectation.)* There exists a positive measurable function $C(\cdot)$ such that for every $d > 0$, $\sup_k \mathbf{E}\left[C(\mathbf{y}_k)^d\right] < \infty$. In addition, for all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$, $\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{y}) - \mathbf{f}_{\bar{\boldsymbol{\theta}}}(\mathbf{y})\| \leq C(\mathbf{y})\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|$, where $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$ represents $\mathbf{G}_{\boldsymbol{\theta}}(\cdot)$ and $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\cdot)$. For all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ and $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, $\|\mathbf{g}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \mathbf{g}_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}(\mathbf{y})\| \leq C(\mathbf{y})(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|)$, where $\mathbf{g}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ represents $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$ and $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\cdot)$.
11) There exists $a > 0$ such that for all $\mathbf{s} \in \mathbb{R}^m$ and $\boldsymbol{\theta} \in \mathbb{R}^n$, $\mathbf{s}'\bar{\mathbf{G}}(\boldsymbol{\theta})\,\mathbf{s} \geq a\|\mathbf{s}\|^2$.

Lemma VI.1: If Assumptions A.(1–11) are satisfied, then $\lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k)\| = 0$ w.p.1.

Proof: See Appendix A. ■

Theorem VI.2: If Assumptions A.(1–11) are satisfied, then $\lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\,\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| = 0$ w.p.1.

Proof: We have

$$\begin{aligned} &\|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \\ &\leq \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k)\| + \|\bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\|. \end{aligned}$$

Due to Assumption A.(9), we have

$$\lim_{k\to\infty} \|\bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \leq C \lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\|,$$

where C is a constant. Combining the above, we have

$$\begin{aligned} 0 &\leq \lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \\ &\leq 0 + \lim_{k\to\infty} \|\bar{\mathbf{h}}(\boldsymbol{\theta}_k, \mathbf{r}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| \\ &\leq 0 + C \lim_{k\to\infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\| \\ &= 0, \qquad \text{w.p.1}, \end{aligned}$$

where the second inequality follows from Lemma VI.1 and the equality is due to Assumption A.(5). We conclude that $\lim_{k\to\infty} \|\bar{\mathbf{G}}(\boldsymbol{\theta}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\boldsymbol{\theta}_k, \bar{\mathbf{r}}(\boldsymbol{\theta}_k))\| = 0$, w.p.1.

B. Critic Convergence

In this section, we will use the results in Section VI-A to prove the convergence of the *Q-critic* and the $\tilde{\mathbf{Q}}$-*critic* presented in Section V-A. Before presenting the convergence results, we first state the following assumptions and definitions.

Assumption B: There exists a function $\tilde{L} : \mathbb{X} \to [1, \infty)$ and constants $0 \leq \rho < 1$, $b > 0$ such that for each $\boldsymbol{\theta} \in \mathbb{R}^n$,

$$\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\tilde{L}(\mathbf{x}_1)] \leq \rho\tilde{L}(\mathbf{x}) + bI_{\mathbf{x}^*}(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathbb{X}, \tag{40}$$

where $\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\cdot]$ denotes expectation under $\boldsymbol{\theta}$ with initial state $\mathbf{x}$, $I_{\mathbf{x}^*}(\cdot)$ is the indicator function for the initial state $\mathbf{x}^*$ being equal to the argument of the function, and $\mathbf{x}_1$ is the (random) state of the MDP after one transition from the initial state.

The assumption above is identical to [12, Assumption 2.5]. We call a function satisfying the inequality (40) a stochastic Lyapunov function. Let $L : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ be a function that satisfies the following assumption.

Assumption C: For each $d > 0$ there is $K_d > 0$ such that

$$\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[L(\mathbf{x}, U_0)^d] \leq K_d\tilde{L}(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathbb{X}, \boldsymbol{\theta} \in \mathbb{R}^n,$$

where U_0 is the random variable of the action at state $\mathbf{x}$.

Note that if any function is upper bounded by a function L as described in Assumption C, then all its steady-state moments are finite.

Lemma VI.3: If two functions $L_f : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ and $L_g : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ satisfy Assumption C, then so does $L_f L_g$.

Proof: For any two random variables A and B, $\mathbf{E}[AB] \leq (1/2)(\mathbf{E}[A^2] + \mathbf{E}[B^2])$. As a result, we have

$$\begin{aligned} &\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}\left[L_f(\mathbf{x}, U_0)^d L_g(\mathbf{x}, U_0)^d\right] \\ &\quad\leq \frac{1}{2}\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}\left[L_f(\mathbf{x}, U_0)^{2d}\right] + \frac{1}{2}\mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}\left[L_g(\mathbf{x}, U_0)^{2d}\right] \\ &\quad\leq \frac{1}{2}(K_{2d}^f + K_{2d}^g)\tilde{L}(\mathbf{x}), \end{aligned}$$

where K_{2d}^f and K_{2d}^g are the bounding constants of f and g appearing in Assumption C. ■

Definition 1: We define $\mathcal{D}^{(2)}$ to be the family of all functions $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, u)$ that satisfy: for all $\mathbf{x} \in \mathbb{X}$ and $u \in \mathbb{U}$, there exists a constant $K > 0$ such that

$$\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, u)\| \le K L(\mathbf{x}, u), \ \forall \boldsymbol{\theta} \in \mathbb{R}^n, \tag{41}$$

$$\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, u) - \mathbf{f}_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\| \le K \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| L(\mathbf{x}, u), \ \forall \boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n, \tag{42}$$

where the bounding function L satisfies Assumption C.

Lemma VI.4: If $\mathbf{f}_{\boldsymbol{\theta}}, \mathbf{g}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, then $\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{g}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ and $\mathbf{f}_{\boldsymbol{\theta}} \mathbf{g}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$.

Proof: The proof for $\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{g}_{\boldsymbol{\theta}}$ is immediate; we focus on $\mathbf{f}_{\boldsymbol{\theta}} \mathbf{g}_{\boldsymbol{\theta}}$. Inequality (41) can be proved using Lemma 4.3(f) of [4]. To prove inequality (42),

$$\begin{aligned}\|\mathbf{f}_{\boldsymbol{\theta}} \mathbf{g}_{\boldsymbol{\theta}} - \mathbf{f}_{\bar{\boldsymbol{\theta}}} \mathbf{g}_{\bar{\boldsymbol{\theta}}}\| &= \|\mathbf{f}_{\boldsymbol{\theta}} \mathbf{g}_{\boldsymbol{\theta}} + \mathbf{f}_{\boldsymbol{\theta}} \mathbf{g}_{\bar{\boldsymbol{\theta}}} - \mathbf{f}_{\boldsymbol{\theta}} \mathbf{g}_{\bar{\boldsymbol{\theta}}} - \mathbf{f}_{\bar{\boldsymbol{\theta}}} \mathbf{g}_{\bar{\boldsymbol{\theta}}}\| \\ &\le \|\mathbf{f}_{\boldsymbol{\theta}}\| \|\mathbf{g}_{\boldsymbol{\theta}} - \mathbf{g}_{\bar{\boldsymbol{\theta}}}\| + \|\mathbf{g}_{\bar{\boldsymbol{\theta}}}\| \|\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{f}_{\bar{\boldsymbol{\theta}}}\| \\ &\le 2 K_f K_g L_f L_g \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|,\end{aligned}$$

where K_f and L_f are the bounding constant and the bounding function for f in (41) and (42), while K_g and L_g are the corresponding quantities for $\mathbf{g}$. According to Lemma VI.3, $L_f L_g$ also satisfies Assumption C, which completes the proof. ∎

We assume $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, which is the same with Assumption 4.1 of [12]. This assumption ensures that the feature vector $\boldsymbol{\phi}_{\boldsymbol{\theta}} = (\phi^1_{\boldsymbol{\theta}}, \ldots, \phi^N_{\boldsymbol{\theta}})$, as a function of the policy parameter $\boldsymbol{\theta}$, is "well behaved." Given our feature vector definition, notice that this assumption requires that the RSP function family $\mu_{\boldsymbol{\theta}}$ is twice continuously differentiable for all $\boldsymbol{\theta}$ with bounded first and second derivatives that belong to $\mathcal{D}^{(2)}$. We also assume that the one-step reward function $g \in \mathcal{D}^{(2)}$.

The critic consists of two parts: a Q-critic that estimates $Q_{\boldsymbol{\theta}}$ (cf. (27), (28)) and a $\tilde{\mathbf{Q}}$-critic that estimates $\tilde{\mathbf{Q}}_{\boldsymbol{\theta}}$ (cf. (29), (30)). The Q-critic is exactly the same with the LSTD-AC algorithm [14], whose convergence has already been proved in [14] under the assumptions imposed. For the $\tilde{\mathbf{Q}}$-critic, denote by $\mathbf{V}(\mathbf{A})$ a column vector stacking all columns in a matrix $\mathbf{A}$. The $\tilde{\mathbf{Q}}$-critic can be written as in (39) if we let

$$\mathbf{s}_k = \left[M\eta^1_k \cdots M\eta^n_k \ (\mathbf{v}^1_k)' \cdots (\mathbf{v}^n_k)' \right]', \tag{43}$$

$$\mathbf{h}_{\boldsymbol{\theta}, \mathbf{r}}(\mathbf{y}) = \begin{bmatrix} M\Gamma(\mathbf{r})\mathbf{r}' \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u) \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u) \\ \Gamma(\mathbf{r})\mathbf{r}' \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u) \mathbf{V}(\mathbf{z} \boldsymbol{\psi}'_{\boldsymbol{\theta}}(\mathbf{x}, u)) \end{bmatrix},$$

$$\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{y}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathrm{diag}(\mathbf{z}, \ldots, \mathbf{z})/M & \mathbf{I} \end{bmatrix}$$

$$\boldsymbol{\Xi}_k = \mathbf{0},$$

where $\mathrm{diag}(\mathbf{z}, \ldots, \mathbf{z})$ denotes an $nN \times n$ block diagonal matrix with every diagonal element being equal to $\mathbf{z}$, $\mathbf{y} = (\mathbf{x}, u, \mathbf{z})$, M is an arbitrary (large) positive constant whose role is to facilitate the convergence proof, and at any iteration k of (39) $\mathbf{r}_k$ iterates as in (28). The stochastic process $\{\mathbf{z}_k\}$ is the eligibility trace iterating as in (27).

To prove the convergence of the $\tilde{\mathbf{Q}}$-*critic*, we just need to verify Assumptions A.(1–11). It is easy to verify that $\mathbf{z}_k = \sum_{l=0}^{k-1} \lambda^{k-l-1} \boldsymbol{\phi}_{\boldsymbol{\theta}_l}(\mathbf{x}_l, u_l)$. First, we establish the following lemma.

Lemma VI.5: For every $d > 0$, we have $\sup_k \mathbf{E}[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d] < \infty$, where $L : \mathbb{X} \times \mathbb{U} \to [1, \infty)$ is a bounded function that satisfies Assumption C.

Proof: According to the triangle inequality, we have

$$\begin{aligned}\|\mathbf{z}_k\|^d &= \|\sum_{l=0}^{k-1} \lambda^{k-l-1} \boldsymbol{\phi}_{\boldsymbol{\theta}_l}(\mathbf{x}_l, u_l)\|^d \\ &\le \sum_{l=0}^{k-1} \lambda^{d(k-l-1)} \|\boldsymbol{\phi}_{\boldsymbol{\theta}_l}(\mathbf{x}_l, u_l)\|^d \\ &\le K_1 \sum_{l=0}^{k-1} \lambda^{d(k-l-1)} L_1(\mathbf{x}_l, u_l)^d,\end{aligned}$$

for some bounded function L_1 that satisfies Assumption C and some positive constant K_1, where the last inequality is due to $\boldsymbol{\phi}_{\boldsymbol{\theta}_k} \in \mathcal{D}^{(2)}$. In addition, we can multiply with $L(\mathbf{x}_k, u_k)^d$ and take expectation on both sides of the above, which yields

$$\begin{aligned}&\mathbf{E}[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d] \\ &\quad \le K_1 \sum_{l=0}^{k-1} \lambda^{d(k-l-1)} \mathbf{E}[L(\mathbf{x}_k, u_k)^d L_1(\mathbf{x}_l, u_l)^d].\end{aligned} \tag{44}$$

Similar to the proof of Lemma VI.3,

$$\begin{aligned}&\mathbf{E}[L(\mathbf{x}_k, u_k)^d L_1(\mathbf{x}_l, u_l)^d] \\ &\quad \le \frac{1}{2}\mathbf{E}[L(\mathbf{x}_k, u_k)^{2d}] + \frac{1}{2}\mathbf{E}[L_1(\mathbf{x}_l, u_l)^{2d}] < \infty.\end{aligned} \tag{45}$$

Combining (44) and (45), we establish that $\mathbf{E}[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d]$ is bounded. ∎

Theorem VI.6: Under iterations (27) and (28),

$$\|\mathbf{r}_{k+1} - \mathbf{r}_k\| \le \gamma_k F^r_k, \qquad \text{w.p.1}, \tag{46}$$

for some random sequence $\{F^r_k\}$ that has bounded moments, where $\{\gamma_k\}$ is the stepsize in (27).

Proof: SeeAppendix B. ∎

Using SSNS stepsizes according to (36), Assumptions A.(1) and (4) will be satisfied because of Theorem VI.6. Now, $\|\mathbf{r}\|\Gamma(\mathbf{r})$ is bounded because of (35). According to (31), $\mathbf{U}_k$ has bounded moments because $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$, $\boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}, u)$, $Q_{\boldsymbol{\theta}}$, and $\tilde{Q}^i_{\boldsymbol{\theta}}$, $\forall i$, have bounded moments. $\mathbf{H}_k$ and $\hat{\mathbf{H}}_k$ should also have bounded moments because the update in (32) is applied only when $\mathbf{U}_k$ is positive definite. As a result, $\Gamma(\mathbf{r}_k)\mathbf{r}'_k \boldsymbol{\phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k) \hat{\mathbf{H}}_k \boldsymbol{\psi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k)$ should have bounded moments, thus, Assumption A.(2) holds. Assumption A.(3) is trivially satisfied. In addition, because the *Q-critic* converges, we have

$$\lim_{k \to \infty} \|\mathbf{r}_k - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\| = 0, \qquad \text{w.p.1},$$

which is Assumption A.(5).

For $i = 1, \ldots, n$, define the function $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} = \boldsymbol{\phi}_{\boldsymbol{\theta}} \psi^i_{\boldsymbol{\theta}}$. Because $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ and $\boldsymbol{\psi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, we obtain $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ according to Lemma VI.4. Notice that for any fixed $\mathbf{r}$ and $\boldsymbol{\theta}$, the $\tilde{\mathbf{Q}}$-critic (43) is equivalent to the Q-critic of an artificial Markov decision process with reward function $g^i_{\boldsymbol{\theta}, \mathbf{r}}(\mathbf{x}, u) = \Gamma(\mathbf{r})\mathbf{r}' \boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u)$, $i = 1, \ldots, n$. As a result, the Poisson equations of Assumption A.(6)

should be satisfied with appropriately defined average steady-state quantities $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})$ and $\bar{\mathbf{G}}(\boldsymbol{\theta})$. More specifically, similar to [4, Sec. 5.2], we have

$$\begin{aligned}
\bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r}) &= \left\langle \underline{1}, g^i_{\boldsymbol{\theta},\mathbf{r}} \right\rangle_{\boldsymbol{\theta}},\\
\bar{\mathbf{z}}(\boldsymbol{\theta}) &= (1-\lambda)^{-1} \left\langle \underline{1}, \phi_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}},\\
\mathbf{h}^i_1(\boldsymbol{\theta}, \mathbf{r}) &= \sum_{k=0}^{\infty} \lambda^k \left\langle P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} - \bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})\underline{1}, \phi_{\boldsymbol{\theta}} \right\rangle_{\boldsymbol{\theta}},\\
\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r}) &= (M\bar{\kappa}^1(\boldsymbol{\theta}, \mathbf{r}), \dots, M\bar{\kappa}^n(\boldsymbol{\theta}, \mathbf{r}),\\
&\quad (\mathbf{h}^1_1(\boldsymbol{\theta}, \mathbf{r}) + \bar{\kappa}^1(\boldsymbol{\theta}, \mathbf{r})\bar{\mathbf{z}}(\boldsymbol{\theta}), \dots,\\
&\quad (\mathbf{h}^n_1(\boldsymbol{\theta}, \mathbf{r}) + \bar{\kappa}^n(\boldsymbol{\theta}, \mathbf{r})\bar{\mathbf{z}}(\boldsymbol{\theta})),\\
\bar{\mathbf{G}}(\boldsymbol{\theta}) &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \operatorname{diag}(\bar{\mathbf{z}}(\boldsymbol{\theta}), \dots, \bar{\mathbf{z}}(\boldsymbol{\theta}))/M & \mathbf{I} \end{bmatrix},
\end{aligned}$$

where $P^k_{\boldsymbol{\theta}}$ denotes the application of the operator $P_{\boldsymbol{\theta}}$ k times. We can interpret $\bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})$ as the steady-state expectation of the "observed reward" function $g^i_{\boldsymbol{\theta},\mathbf{r}}$.

Let now $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) = \Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u)\mathbf{z}$, $i = 1, \dots, n$. It can be seen that if $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}$ are bounded and Lipschitz continuous in expectation for all $i = 1, \dots, n$, then $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}$ should also be bounded and Lipschitz continuous in expectation. Recall that $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$. For $i = 1, \dots, n$ and each $d > 0$,

$$\begin{aligned}
&\sup_k \mathbf{E}\left[\|\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}_k)\|^d\right]\\
&\le (\Gamma(\mathbf{r})\|\mathbf{r}\|)^d \sup_k \mathbf{E}\left[\|\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}_k, u_k)\|^d \|\mathbf{z}_k\|^d\right]\\
&\le (\Gamma(\mathbf{r})\|\mathbf{r}\|)^d K^d \sup_k \mathbf{E}\left[L(\mathbf{x}_k, u_k)^d \|\mathbf{z}_k\|^d\right],
\end{aligned}$$

for some function L that satisfies Assumption C and some positive constant K. According to (35), $\Gamma(\mathbf{r})\|\mathbf{r}\|$ is bounded. Using Assumption C and Lemma VI.5, it follows that $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}$ satisfies Assumption A.(8). Using Lemma VI.5 it also follows that $\mathbf{G}_{\boldsymbol{\theta}}$ satisfies the same assumption.

It is easy to verify that the function $f(\mathbf{r}) = \Gamma(\mathbf{r})\mathbf{r}$ is Lipschitz continuous and suppose its Lipschitz constant is C_Γ. We will next prove that $\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y})$ is Lipschitz continuous in expectation. For all $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$, $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, and $i = 1, \dots, n$, we have

$$\begin{aligned}
&\|\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \tilde{\mathbf{h}}^i_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}(\mathbf{y})\|\\
&\quad \le \|\Gamma(\mathbf{r})\mathbf{r}'\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u)\mathbf{z} - \Gamma(\bar{\mathbf{r}})\bar{\mathbf{r}}'\boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\mathbf{z}\|\\
&\quad \le \|\mathbf{z}\|\Gamma(\mathbf{r})\|\mathbf{r}'\left(\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u) - \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\right)\|\\
&\qquad + \|\mathbf{z}\|\|\left(\Gamma(\mathbf{r})\mathbf{r} - \Gamma(\bar{\mathbf{r}})\bar{\mathbf{r}}\right)'\boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\|\\
&\quad \le \|\mathbf{z}\|\Gamma(\mathbf{r})\|\mathbf{r}\|\|\boldsymbol{\xi}^i_{\boldsymbol{\theta}}(\mathbf{x}, u) - \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\|\\
&\qquad + \|\mathbf{z}\|\|\boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}(\mathbf{x}, u)\|C_\Gamma\|\mathbf{r} - \bar{\mathbf{r}}\|. \qquad (47)
\end{aligned}$$

Recall that $\boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$. Let K and L be the bounding constant and the bounding function for $\boldsymbol{\xi}^i_{\boldsymbol{\theta}}$; then

$$\|\tilde{\mathbf{h}}^i_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) - \tilde{\mathbf{h}}^i_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}(\mathbf{y})\| \le C(\mathbf{y})\left(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|\right),$$

where $C(\mathbf{y}) = (\Gamma(\mathbf{r})\|\mathbf{r}\| + C_\Gamma)KL(\mathbf{x}, u)\|\mathbf{z}\|$ and $\mathbf{y} = (\mathbf{x}, u, \mathbf{z})$. Using the fact that $\Gamma(\mathbf{r})\|\mathbf{r}\|$ is bounded and Lemma VI.5, it follows that $\mathbf{E}[C(\mathbf{y})^d] < \infty$ for each $d > 0$. As a result, $\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}$ satisfies Assumption A.(10). Moreover, replicating an argument from [4, Sec. 5.2] it can also be shown that $\mathbf{G}_{\boldsymbol{\theta}}$ satisfies the same assumption. Furthermore, defining

$$\begin{aligned}
\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}) &= \sum_{k=0}^{\infty} \mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\mathbf{h}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y}_k) - \bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})|\mathbf{y}_0 = \mathbf{y}],\\
\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{y}) &= \sum_{k=0}^{\infty} \mathbf{E}_{\boldsymbol{\theta},\mathbf{x}}[\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{y}_k) - \bar{\mathbf{G}}(\boldsymbol{\theta})|\mathbf{y}_0 = \mathbf{y}],
\end{aligned}$$

we can use similar arguments as above to establish that these functions satisfy Assumption A.(8) and (10).

Lemma VI.7: Let $\hat{\boldsymbol{\theta}} = (\boldsymbol{\theta}, \mathbf{r})$. Let also $\hat{\mathcal{D}}^{(2)}$ be the counterpart of $\mathcal{D}^{(2)}$ for functions parameterized by $\hat{\boldsymbol{\theta}}$. Then $P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}}$ belongs to $\hat{\mathcal{D}}^{(2)}$ for all nonnegative integers k.

Proof: A simple observation is that $\mathcal{D}^{(2)} \subseteq \hat{\mathcal{D}}^{(2)}$ and that Lemma VI.4 still holds for $\hat{\mathcal{D}}^{(2)}$. Namely, a product function $f_{\hat{\boldsymbol{\theta}}} g_{\hat{\boldsymbol{\theta}}} \in \hat{\mathcal{D}}^{(2)}$ if $f_{\hat{\boldsymbol{\theta}}} \in \hat{\mathcal{D}}^{(2)}$ and $g_{\hat{\boldsymbol{\theta}}} \in \hat{\mathcal{D}}^{(2)}$.

$P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}}$ can be written as $P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} = \Gamma(\mathbf{r})\mathbf{r}' P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}}$. We first observe that $P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$ according to [32, Corollary 2.4]. To verify (41), we have (in functional relationships)

$$\|P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}}\| \le \Gamma(\mathbf{r})\|\mathbf{r}\|\|P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}}\| \le \Gamma(\mathbf{r})\|\mathbf{r}\|KL.$$

To verify (42), for $\boldsymbol{\theta}, \bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ and $\mathbf{r}, \bar{\mathbf{r}} \in \mathbb{R}^N$, we have

$$\begin{aligned}
&\|P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} - P^k_{\bar{\boldsymbol{\theta}}} g^i_{\bar{\boldsymbol{\theta}},\bar{\mathbf{r}}}\|\\
&\quad \le \Gamma(\mathbf{r})\|\mathbf{r}\|\|P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}} - P^k_{\bar{\boldsymbol{\theta}}} \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}\| + \|P^k_{\bar{\boldsymbol{\theta}}} \boldsymbol{\xi}^i_{\bar{\boldsymbol{\theta}}}\|C_\Gamma\|\mathbf{r} - \bar{\mathbf{r}}\|\\
&\quad \le \Gamma(\mathbf{r})\|\mathbf{r}\|KL\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + KLC_\Gamma\|\mathbf{r} - \bar{\mathbf{r}}\|\\
&\quad \le \left(\Gamma(\mathbf{r})\|\mathbf{r}\| + C_\Gamma\right) KL \left(\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\| + \|\mathbf{r} - \bar{\mathbf{r}}\|\right),
\end{aligned}$$

where K and L are the bounding constant and function of $P^k_{\boldsymbol{\theta}} \boldsymbol{\xi}^i_{\boldsymbol{\theta}}$, respectively. ■

Using the fact that $g^i_{\boldsymbol{\theta},\mathbf{r}}, \phi_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, $\bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})$ and $\bar{\mathbf{z}}(\boldsymbol{\theta})$ are bounded and Lipschitz continuous with respect to $\hat{\boldsymbol{\theta}}$ due to [32, Corollary 5.3]. It can be easily verified that $(P^k_{\boldsymbol{\theta}} g^i_{\boldsymbol{\theta},\mathbf{r}} - \bar{\kappa}^i(\boldsymbol{\theta}, \mathbf{r})\underline{1})\phi_{\boldsymbol{\theta}} \in \hat{\mathcal{D}}^{(2)}$ using Lemma VI.7 and Lemma VI.4. Again, using [32, Corollary 5.3], we can obtain that $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})$ is bounded and Lipschitz continuous with respect to $\hat{\boldsymbol{\theta}}$. As a result, $\bar{\mathbf{h}}(\boldsymbol{\theta}, \mathbf{r})$ satisfies Assumption A.(7) and (9). Similarly, it can also be shown that $\bar{\mathbf{G}}(\boldsymbol{\theta})$ satisfies the same assumptions. Finally, it can also be verified that $\hat{\mathbf{h}}_{\boldsymbol{\theta},\mathbf{r}}(\mathbf{y})$ and $\hat{\mathbf{G}}_{\boldsymbol{\theta}}(\mathbf{y})$ satisfy the same assumptions using similar arguments.

The final step in verifying all parts of Assumption A is part (11). That follows from [4, Lemma 5.3]. Having established all parts of Assumption A, the convergence of the $\hat{\mathbf{Q}}$-critic follows.

C. Actor Convergence

The actor update defined in (34) is similar to the actor update using the unscaled gradient. The difference is that the gradient estimate is multiplied by a positive definite matrix. This section will present the convergence results for this type of actors.

Define

$$\mathbf{S}_{\boldsymbol{\theta}}(\mathbf{x},u) = \mathbf{H}_{\boldsymbol{\theta}}\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x},u)\boldsymbol{\phi}'_{\boldsymbol{\theta}}(\mathbf{x},u),$$

where $\mathbf{H}_{\boldsymbol{\theta}}$ is a positive definite matrix for all $\boldsymbol{\theta}$. Let $\bar{\mathbf{S}}(\boldsymbol{\theta}) = \langle \underline{1}, \mathbf{S}_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}}$ and let $\bar{\mathbf{r}}(\boldsymbol{\theta})$ be the limit of the critic parameter $\mathbf{r}$ if the policy parameter is held fixed to $\boldsymbol{\theta}$. Similar to [12], the *actor* update can be written as

$$\begin{aligned}\boldsymbol{\theta}_{k+1} =\ & \boldsymbol{\theta}_k + \beta_k \mathbf{S}_{\boldsymbol{\theta}}(\mathbf{x}_k,u_k)\mathbf{r}_k\Gamma(\mathbf{r}_k)\\ =\ & \boldsymbol{\theta}_k + \beta_k \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\\ & + \beta_k\left(\mathbf{S}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\right)\mathbf{r}_k\Gamma(\mathbf{r}_k)\\ & + \beta_k\bar{\mathbf{S}}(\boldsymbol{\theta}_k)\left(\mathbf{r}_k\Gamma(\mathbf{r}_k) - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\right).\end{aligned}$$

Define

$$\begin{aligned}\mathbf{f}(\boldsymbol{\theta}_k) &= \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\bar{\mathbf{r}}(\boldsymbol{\theta}_k),\\ \mathbf{e}_k^{(1)} &= \left(\mathbf{S}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\right)\mathbf{r}_k\Gamma(\mathbf{r}_k),\\ \mathbf{e}_k^{(2)} &= \bar{\mathbf{S}}(\boldsymbol{\theta}_k)\left(\mathbf{r}_k\Gamma(\mathbf{r}_k) - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\right).\end{aligned}$$

Then, the actor update becomes:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \beta_k\left(\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\mathbf{f}(\boldsymbol{\theta}_k) + e_k^{(1)} + e_k^{(2)}\right).$$

$\mathbf{f}(\boldsymbol{\theta}_k)$ is the expected actor update, while $e_k^{(1)}$ and $e_k^{(2)}$ are two error terms due to the fact that the update is performed on a sample path of the MDP. Using Taylor's series expansion,

$$\begin{aligned}\bar{\alpha}(\boldsymbol{\theta}_{k+1}) \geq\ & \bar{\alpha}(\boldsymbol{\theta}_k) + \beta_k\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\nabla\bar{\alpha}(\boldsymbol{\theta}_k)'\mathbf{f}(\boldsymbol{\theta}_k)\\ & + \beta_k\nabla\bar{\alpha}(\boldsymbol{\theta}_k)'\mathbf{e}_k^{(1)} + \beta_k\nabla\bar{\alpha}(\boldsymbol{\theta}_k)'\mathbf{e}_k^{(2)}.\end{aligned}$$

Lemma VI.8: (Convergence of the noise terms). It holds:

- $\sum_{k=0}^{\infty}\beta_k\nabla\bar{\alpha}(\boldsymbol{\theta}_k)'\mathbf{e}_k^{(1)}$ converges w.p.1.
- $\lim_k \mathbf{e}_k^{(2)} = 0$ w.p.1.

Proof: Let $\hat{\mathbf{e}}_k^{(1)} = \left(\boldsymbol{\xi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) - \bar{\boldsymbol{\xi}}(\boldsymbol{\theta}_k)\right)\mathbf{r}_k\Gamma(\mathbf{r}_k)$ and $\hat{\mathbf{e}}_k^{(2)} = \bar{\boldsymbol{\xi}}(\boldsymbol{\theta}_k)\left(\mathbf{r}_k\Gamma(\mathbf{r}_k) - \bar{\mathbf{r}}(\boldsymbol{\theta}_k)\Gamma(\bar{\mathbf{r}}(\boldsymbol{\theta}_k))\right)$, where $\boldsymbol{\xi}_{\boldsymbol{\theta}}(\mathbf{x},u) = \boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x},u)\boldsymbol{\phi}'_{\boldsymbol{\theta}}(\mathbf{x},u)$ and $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta}) = \langle \underline{1}, \boldsymbol{\xi}_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}} = \langle \boldsymbol{\psi}_{\boldsymbol{\theta}}, \boldsymbol{\phi}'_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}}$. Then, $\hat{\mathbf{e}}_k^{(1)}$ and $\hat{\mathbf{e}}_k^{(2)}$ are the two error terms for the actor update using the unscaled gradient [4]. It easily follows that $\mathbf{e}_k^{(1)} = \mathbf{H}_{\boldsymbol{\theta}_k}\hat{\mathbf{e}}_k^{(1)}$ and $\mathbf{e}_k^{(2)} = \mathbf{H}_{\boldsymbol{\theta}_k}\hat{\mathbf{e}}_k^{(2)}$. Furthermore, $\mathbf{S}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k) = \mathbf{H}_{\boldsymbol{\theta}_k}^{-1}\boldsymbol{\xi}_{\boldsymbol{\theta}_k}(\mathbf{x}_k,u_k)$. The lemma can be proved by combining these facts with [4, Lemma 6.2]. ■

Lemma VI.8 shows that $\mathbf{e}_k^{(1)}$ can be averaged out and $\mathbf{e}_k^{(2)}$ goes to zero. As a result, the two error terms are negligible and the update is determined by the expected direction $\mathbf{f}(\boldsymbol{\theta})$ in the long run.

Lemma VI.9: We have $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{\theta}) + \varepsilon(\lambda,\boldsymbol{\theta})$, where $\mathbf{g}(\boldsymbol{\theta})$ is a function such that $\nabla\bar{\alpha}(\boldsymbol{\theta})'\mathbf{g}(\boldsymbol{\theta}) \geq 0$, and $\sup_\theta|\varepsilon(\lambda,\boldsymbol{\theta})| < C(1-\lambda)$ for some constant $C > 0$ independent of λ.

Proof: According to (5), $\nabla\bar{\alpha}(\boldsymbol{\theta}) = \langle\boldsymbol{\psi}_{\boldsymbol{\theta}}, Q_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}} = \langle\boldsymbol{\psi}_{\boldsymbol{\theta}}, \boldsymbol{\phi}'_{\boldsymbol{\theta}}\bar{\mathbf{r}}(\boldsymbol{\theta})\rangle_{\boldsymbol{\theta}} = \bar{\boldsymbol{\xi}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta})$. For $\lambda = 1$, we have

$$\begin{aligned}\nabla\bar{\alpha}(\boldsymbol{\theta})'\mathbf{f}(\boldsymbol{\theta}) =\ & \nabla\bar{\alpha}(\boldsymbol{\theta})'\bar{\mathbf{S}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta})\\ =\ & \bar{\mathbf{r}}(\boldsymbol{\theta})'\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})'\bar{\mathbf{S}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta}).\end{aligned}$$

Notice that $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})'\bar{\mathbf{S}}(\boldsymbol{\theta}) \succeq 0$. Specifically,

$$\begin{aligned}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})'\bar{\mathbf{S}}(\boldsymbol{\theta}) =\ & \langle\boldsymbol{\psi}'_{\boldsymbol{\theta}}, \boldsymbol{\phi}_{\boldsymbol{\theta}}\rangle_{\boldsymbol{\theta}}\left\langle\mathbf{H}_{\boldsymbol{\theta}}, \boldsymbol{\psi}_{\boldsymbol{\theta}}\boldsymbol{\phi}'_{\boldsymbol{\theta}}\right\rangle_{\boldsymbol{\theta}}\\ =\ & \mathbf{H}_{\boldsymbol{\theta}}\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})'\bar{\boldsymbol{\xi}}(\boldsymbol{\theta}),\end{aligned}$$

where $\mathbf{H}_{\boldsymbol{\theta}} \succeq 0$ and $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})'\bar{\boldsymbol{\xi}}(\boldsymbol{\theta}) \succeq 0$ by construction. As a result, $\bar{\boldsymbol{\xi}}(\boldsymbol{\theta})'\bar{\mathbf{S}}(\boldsymbol{\theta}) \succeq 0$, which implies that $\nabla\bar{\alpha}(\boldsymbol{\theta})'\mathbf{f}(\boldsymbol{\theta}) \geq 0$.

The proof for $\lambda < 1$ follows the proof in [4]. Let us write $\bar{\mathbf{r}}^{\lambda}(\boldsymbol{\theta})$ for the steady-state expectation of $\mathbf{r}_k$. Following the proof of [4], we have $\|\bar{\mathbf{r}}^{\lambda}(\boldsymbol{\theta}) - \mathbf{r}(\boldsymbol{\theta})\| \leq C_0(1-\lambda)$ for some positive constant C_0. Let $\mathbf{g}(\boldsymbol{\theta}) = \bar{\mathbf{S}}(\boldsymbol{\theta})\bar{\mathbf{r}}(\boldsymbol{\theta})$, where $\bar{\mathbf{r}}(\boldsymbol{\theta})$ is the steady-state expectation of $\mathbf{r}_k$ when $\lambda = 1$. Then we can still obtain $\nabla\bar{\alpha}(\boldsymbol{\theta})'\mathbf{g}(\boldsymbol{\theta}) \geq 0$. In addition, $\|\mathbf{f}(\boldsymbol{\theta}) - \mathbf{g}(\boldsymbol{\theta})\| = \|\bar{\mathbf{S}}(\boldsymbol{\theta})(\bar{\mathbf{r}}^{\lambda}(\boldsymbol{\theta}) - \bar{\mathbf{r}}(\boldsymbol{\theta}))\| \leq C(1-\lambda)$ for some C. ■

Lemma VI.9 shows that the expected direction $\mathbf{f}(\boldsymbol{\theta})$ is always a gradient ascent direction for λ sufficiently close to 1. We arrive at the following convergence result whose proof is similar to [4, Thm. 6.3].

Theorem VI.10 Actor Convergence: For any $\epsilon > 0$, there exists some λ sufficiently close to 1 such that the second-order Actor-Critic algorithm satisfies $\lim_{k\to\infty}\inf_k|\nabla\bar{\alpha}(\theta_k)| < \epsilon$ w.p.1. That is, $\boldsymbol{\theta}_k$ visits an arbitrary neighborhood of a stationary point infinitely often.

VII. Case Study

A. Garnet Problem

This section reports empirical results from our method applied to GARNET problems introduced in [23]. GARNET problems do not correspond to any particular application; they are meant to be generic, yet, representative of MDPs one encounters in practical applications [23]. As we mentioned earlier, GARNET stands for "Generic Average Reward Non-stationary Environment Testbed."

A GARNET problem is characterized by 5 parameters and can be written as GARNET(n, m, b, σ, τ). The parameters n and m are the number of states and actions, respectively. For each state-action pair, there are b possible next states, and each next state is chosen randomly without replacement. The transition probabilities to these b states are generated as follows: we divide a unit-length interval into b segments by choosing $b-1$ breaking points according to a uniform random distribution. The lengths of these segments represent the transition probabilities and they are randomly assigned to the b states we have already selected.

The expected reward for each transition is a normally distributed random variable with zero mean and unit variance. The actual reward is a normally distributed random variable whose mean is the expected reward and whose variance is 1.

The parameter $\tau, 0 \leq \tau \leq 1/n$, determines the degree of non-stationarity in the problem. If $\tau = 0$, the GARNET problem is stationary. Otherwise, if $\tau > 0$, one of the states will be selected with probability $n\tau$ at each time step and randomly reconstructed as described above.

To apply the actor-critic algorithm, the key step is to design an RSP $\mu_{\theta}(u|\mathbf{x})$. In this case study, we define the RSP to be the Boltzmann distribution that is based on some state-action features. Good state-action features should be interpretable and could help reduce the number of parameters in the RSP.

We first define the state feature $\mathbf{f}_{\mathsf{S}}(\mathbf{x})$ to be a binary vector of length d, i.e., $\mathbf{f}_{\mathsf{S}}(\mathbf{x}) \in \{0,1\}^d$, for each state $\mathbf{x}$. There is a parameter l specifying the number of components in the state feature that are equal to 1. State features are randomly generated and we make sure no two states have the same state feature.

In [23], the state-action feature is constructed by padding zeros to state features so that the features for different actions are orthogonal. As a result, the dimensionality of the state-action feature constructed in this manner is equal to $d|\mathbb{U}|$. This approach significantly increases the feature dimensionality, especially when the action space is very large. In this paper, we use the state-action feature described below. For each state $\mathbf{x}_0$ and action u, the state-action feature is:

$$\mathbf{f}_{\mathsf{SA}}(\mathbf{x}_0, u) = \mathbf{E}[\mathbf{f}_{\mathsf{S}}(\mathbf{x}_1)|u] - \mathbf{f}_{\mathsf{S}}(\mathbf{x}_0), \tag{48}$$

where $\mathbf{E}[\mathbf{f}_{\mathsf{S}}(\mathbf{x}_1)|u] = \sum_{\mathbf{x}_1} p(\mathbf{x}_1|\mathbf{x}, u)\mathbf{f}_{\mathsf{S}}(\mathbf{x}_1)$ is the expected feature at the next state after applying action u.

With the state-action feature as in (48), the probability of taking action u in state $\mathbf{x}$ is set to

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(u|\mathbf{x}) = \frac{e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}{\sum_{u\in\mathbb{U}} e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}, \tag{49}$$

which is a typical Boltzmann distribution with T being the temperature of the distribution. With the state-action feature described above, we can interpret $-\mathbf{f}_{\mathsf{S}A}(\mathbf{x}, u)'\boldsymbol{\theta}$ as the "energy" and the distribution prefers actions that lead to lower energy.

A common consideration in RSP design is the so-called exploitation-exploration tradeoff [2]. An RSP exhibits higher exploitation if it is more greedy, i.e., it is more likely to only pick the most desirable action. However, sometimes the exploration of undesirable actions is necessary because they may be desirable in the long run. High exploitation and low exploration may result in a sub-optimal solution. On the contrary, low exploitation and high exploration may reduce the convergence rate of the actor-critic algorithm. Our RSP defined in (49) is flexible because tuning T in (49) can effectively adjust the degree of exploration. High temperature T implies more exploration while low temperature T implies more exploitation.

In this empirical study, we compare our algorithm with the LSTD-AC algorithm in [14], and the four algorithms in [23], which are henceforth referred to as BSGL1 to BSGL4, in a GARNET problem GARNET$(50, 4, 5, 0.1, 0)$. BSGL1 is based on a "vanilla" gradient ascent and BSGL2-BSGL4 are based on natural gradients. Henceforth, for state features we let $d = 8$ and $l = 3$. The state-features are randomly assigned and we make sure no two states have the same state-feature. For all algorithms, the critic step-size is $\alpha_k = \frac{\alpha_0 \cdot \alpha_c}{\alpha_c + k^{2/3}}$ and the actor stepsize $\beta_c = \frac{\beta_0 \cdot \beta_c}{\beta_c + k}$, where $\alpha_c = \beta_c = 1000$. For the LSTD actor-critic and our method $\alpha_0 = 0.1$ and $\beta_0 = 0.1$. For BSGL1 and BSGL2, $\alpha_0 = 0.1$ and $\beta_0 = 0.01$. For BSGL3 and BSGL4, we choose $\alpha_0 = 0.01$ and $\beta_0 = 0.001$. For all algorithms, the initial parameters

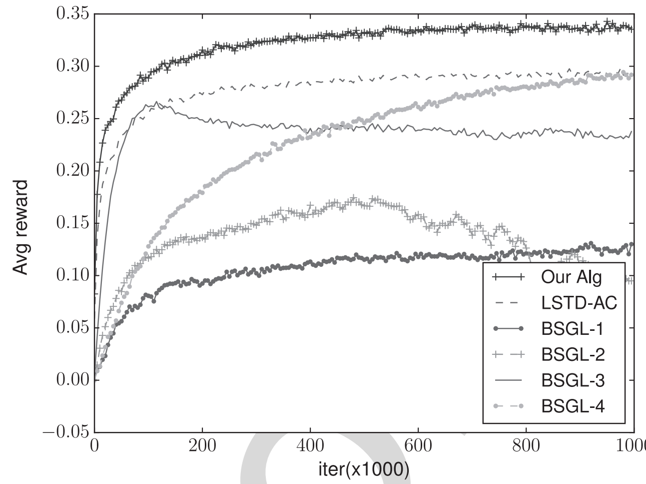

Fig. 2. Comparison of our algorithm with LSTD and natural actor-critic algorithms.

$\boldsymbol{\theta}_0$ are zero and the temperature in (49) is set to $T = 1$. For our algorithm, we choose $\chi_{\min} = 100$ (cf. (33)).

We run each algorithm 50 times independently and Fig. 2 displays the mean of the average reward for the first 1,000,000 iterations. Table I summarizes the convergence time and converged average reward for each algorithm. For each problem, the first two columns of Table I show the mean and standard deviation of the reward achieved. The third and fourth columns list the time (mean and standard deviation) it takes to convergence. The last column shows the average CPU time per iteration (TPI). The results are based on 50 independent runs for the GARNET problem and 100 independent runs for the robot control problem. Note that BSGL2 becomes numerically unstable after 500,000 iterations, so the reward of BSGL2 in Table I is the maximal reward before numerical instability occurs and the time is the time it takes to reach the maximal reward.

Compared to the LSTD-AC method, our method adds a second-order critic update and takes advantage of the Hessian estimate in the actor update. For this problem, the average CPU time of one LSTD-AC iteration is 1288 μs. In comparison, the average CPU time for one iteration of our algorithm is 1818 μs, which means that computing the second-order critic and the inverse of the Hessian adds about 41% to the computational cost. Despite the larger CPU time per iteration, our algorithm still converges faster than LSTD-AC because fewer iterations are needed. The CPU time per iteration of both our algorithm and LSTD-AC is larger than BSGL1-4. This is likely because both our algorithm and LSTD-AC use a state-action feature vector, whose dimensionality is larger than the one used in BSGL1-4 for value function approximations.

Among the four algorithms in [23], BSGL3 converges faster, which is consistent with the empirical study in [23]. Compared to BSGL3, although our algorithm uses longer time to converge, it converges to higher value (0.33) than BSGL3 (0.24). On average our algorithm takes only 43 seconds to reach an average reward of 0.24 vs. 122 seconds needed by BSGL3 to reach the same value.

TABLE I
COMPARISON OF ALL ALGORITHMS IN A GARNET AND A ROBOT CONTROL PROBLEM.

Alg. Name	GARNET					Robot Control				
	Reward		Conv. Time (s)		TPI(μ s)	Reward		Conv. Time (s)		TPI(μ s)
	Mean	Std	Mean	Std		Mean	Std	Mean	Std	
Our Alg.	0.33	0.070	727	10.9	1818	0.0916	0.00109	118	3.0	3281
LSTD-AC	0.29	0.091	773	9.9	1288	0.0851	0.0235	187	23	2837
BSGL-1	0.11	0.083	540	7.5	601	0.0819	0.000731	217	2.9	2173
BSGL-2	0.16	0.078	342	4.4	684	0.0909	0.00136	231	9.8	2313
BSGL-3	0.24	0.093	122	1.6	678	0.0927	0.000936	142	6.4	2372
BSGL-4	0.28	0.082	686	11.6	686	0.0916	0.000860	209	5.0	2319

For BSGL2, the Table Displays the Maximal Average Reward Before Numerical Instability Happens and the Time to Reach the Reward

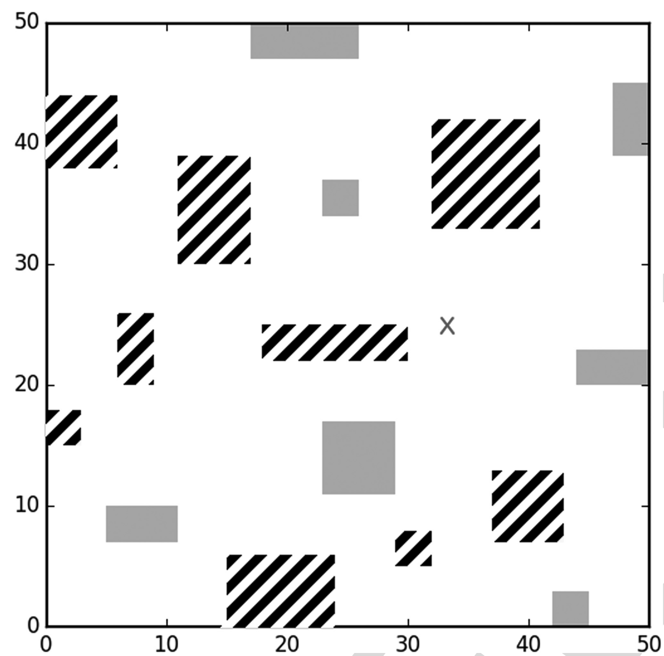

Fig. 3. View of the mission environment, where the initial region is marked by ``x'', the goal regions are marked by green colors, and the unsafe regions are displayed in black stripes.

B. Robot Control Problem

In this section we compare the performance of our algorithm with other algorithms in a robotics application. Fig. 3 shows the mission environment, which is a 50×50 grid. We model the motion of the robot in the environment as the following MDP $\mathbb{M}$:

- **State space**. Each state $\mathbf{x} \in \mathbb{X}$ corresponds to a region in the mission environment and can be represented by a coordinate (i, j), where i is the row number and j is the column number.
- **Action space**. The action space $\mathbb{U} = \{u_1, u_2, u_3, u_4\}$ corresponds to four control primitives (actions): "North," "East," "South," and "West," which represent the directions in which the robot intends to move. Depending on the location of a region, some of these actions may not be enabled, for example, in the lower-left corner, only actions "North" and "East" are enabled. For each state $\mathbf{x}$, let $\mathbb{U}_e(\mathbf{x})$ denote the enabled actions in this state.
- **Transitional model**. A control action does not necessarily lead the robot to the intended direction because the outcome is subject to noise in actuation and possible surface roughness in the environment. In this problem, a robot can only move to the adjacent state in one step. For each enabled control, the robot moves to the intended direction with probability 0.7 and moves to other allowed directions with equal probabilities.
- **Initial state**. The robot starts from state $\mathbf{x}_0$, which is labeled as 'x' in Fig. 3.
- **Reward function**. There are some *unsafe* regions $\mathbb{X}_U$, which should be avoided, in the mission environment. There are also some *goal* states $\mathbb{X}_G$ that should be visited as often as possible. The *unsafe* and *goal* states are displayed as black stripes and green solid colors in Fig. 3, respectively. The objective is to find an optimal policy that maximizes the *expected average reward* with an one-step reward function defined by

$$g(\mathbf{x}, u) = \begin{cases} 1, & \text{if} x \in \mathbb{X}_G, \\ -1, & \text{if} x \in \mathbb{X}_U, \\ 0, & \text{otherwise.} \end{cases}$$

This problem is the foundation of many complex robot motion control problems in which MDPs are defined in more complex ways, i.e., using temporal logic [15]–[17].

In this problem, we consider two state features that represent the *safety* and *affinity* of the state, respectively. For each pair of states $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}$, we define $d(\mathbf{x}_i, \mathbf{x}_j)$ to be the minimum number of transitions from $\mathbf{x}_i$ to $\mathbf{x}_j$. We say $\mathbf{x}_j \in \mathscr{N}(\mathbf{x}_i)$—a neighborhood of $\mathbf{x}_i$—if and only if $d(\mathbf{x}_i, \mathbf{x}_j) \leq r_n$, for some fixed integer r_n given *a priori*. For each state $\mathbf{x} \in \mathbb{X}$, the safety score is defined as the ratio of the safe neighboring states over all neighboring states of $\mathbf{x}$. Namely,

$$\mathsf{safety}(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \mathscr{N}(\mathbf{x})} I_s(\mathbf{y})}{|\mathscr{N}(\mathbf{x})|}, \tag{50}$$

where $I_s(\mathbf{y})$ is an indicator function such that $I_s(\mathbf{y}) = 1$ if and only if $\mathbf{y} \in \mathbb{X} \setminus \mathbb{X}_U$ and $I_s(\mathbf{y}) = 0$ otherwise. A higher safety score for the current state of the robot means it is less likely for

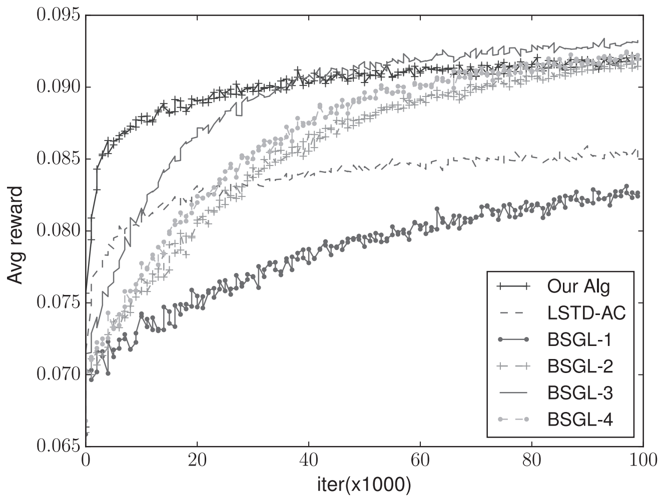

Fig. 4. Comparison of our algorithm with LSTD and natural actor-critic algorithms.

the robot to reach an unsafe region in the future. We define the affinity score of a state $\mathbf{x} \in \mathbb{X}$ as

$$\mathsf{affinity}(\mathbf{x}) = -\min_{\mathbf{y}\in\mathbb{X}_G} d(\mathbf{x}, \mathbf{y})$$

which is the negative of the minimum number of transitions from $\mathbf{x}$ to any goal state. The state feature is defined to be

$$\mathbf{f}_{\mathsf{S}}(\mathbf{x}) = [\mathsf{safety}(\mathbf{x}), \mathsf{affinity}(\mathbf{x})],$$

and the state-action feature $\mathbf{f}_{\mathsf{S}A}(\mathbf{x}, u)$ is calculated using (48). In this application, we use the following Boltzmann distribution.

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(u|\mathbf{x}) = \frac{e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}{\sum_{u\in\mathbb{U}_e(\mathbf{x})} e^{\mathbf{f}_{\mathsf{S}A}(\mathbf{x},u)'\boldsymbol{\theta}/T}}, \tag{51}$$

where T is the temperature. Note that the only difference of (51) with (49) is that (51) restricts to enabled actions.

Again, we compare our algorithm with the LSTD-AC algorithm in [14] and the four algorithms in [23]. We run each algorithm 100 times independently and Fig. 4 shows the comparison of the average reward for the first 100,000 iterations. For all algorithms, the initial $\boldsymbol{\theta}$ is $(0, 5)$ and the temperature $T = 5$. The step-sizes satisfy $\alpha_c = \frac{\alpha_0 \cdot \alpha_c}{\alpha_c + k^{2/3}}$ and $\beta_c = \frac{\beta_0 \cdot \beta_c}{\beta_c + k}$. For LSTD-AC and our algorithm, we set $\alpha_0 = 0.1$, $\alpha_c = 1000$, $\beta_0 = 0.01$ and $\beta_c = 1000$. For BSGL1-BSGL4, we set $\alpha_0 = 0.1$, $\alpha_c = 1000$, $\beta_0 = 0.001$ and $\beta_c = 10000$. We use $\chi_{min} = 30$ in (32).

Table I summarizes the convergence time and the converged reward for all algorithms. Among the three natural gradient-based algorithms, BSGL3 performs the best, but on average it is still slower than our method in this problem. The convergence rate of BSGL1 is much worse than the rest of the algorithms. For this problem, we did not observe numerical instability for BSGL2.

For the robot control problem, the average CPU time per iteration is 3281 μs for our algorithm vs. 2837 μs for LSTD-AC, that is, about 15.7% higher. The computational overhead of the second-order critic in this problem is much lower than in the GARNET problem, which is due to the fact that the robot control problem has less parameters.

The CPU time per iteration of both LSTD-AC and our algorithm is larger than that of BSGL1-BSGL4, but the difference is much smaller compared with the GARNET problem. Since significant less iterations are needed for our algorithm, it converges faster than all other algorithms. Specifically, the second-best algorithm, BSGL3, takes on average 20.3% more time to converge.

VIII. Conclusions and Future Work

In this paper we propose a general estimate for the Hessian matrix of the long-run reward in actor-critic algorithms. Based on this estimate, we present a novel second-order actor-critic algorithm which uses second-order critic and actor. The actor, in particular, uses a direct estimate of the Hessian matrix to improve the rate of convergence for ill-conditioned problems. Building on the LSTD-AC algorithm in [16], [14], our algorithm extends the *critic* to approximate the Hessian and revises the *actor* to update the policy parameters using Newton's method. We compare our algorithm with the LSTD-AC algorithm and the four algorithms in [23], three of which are based on natural gradients, in two applications. The results show that our method can achieve a better rate of convergence for many problems.

As a variant of Newton's method, our method has similar limitations. First, the cost of maintaining a Hessian estimate is quadratic to the number of parameters. As a result, our algorithm is only suitable for problems with relatively small number of parameters. Second, our algorithm requires the second derivative of the policy function, which implies that the method can not be applied if the policy function is not twice differentiable or its second-order derivatives are hard to obtain. Our algorithm is suitable for the cases where the reward is more sensitive to some parameters vs. others, leading to potentially ill-conditioned problems that are best handled by Newton's method.

One direction for future work is to use part of (9) rather than all four terms, so as to achieve a better tradeoff between convergence rate and computational cost per iteration. In addition, the algorithm described in this paper is suitable for the average reward problem. Since Theorem IV.2 holds for all three types of rewards, similar algorithms can be derived for the discounted and the total reward cases.

Acknowledgment

The authors would like to thank Weicong Ding for useful discussions relating to the proof of critic convergence.

Appendix A
Proof of Lemma VI.1

Lemma A.1: Suppose $\{\gamma_k\}$, $\{\zeta_k\}$, $\{\beta_k\}$ are three deterministic positive sequences that satisfy (36) for some $d_1, d_2 > 0$. Then,

$$\sum_k (\max(\gamma_k, \beta_k)/\zeta_k)^d < \infty \quad \text{for some d} > 0.$$

Proof: Note that $\lim_k(\gamma_k/\zeta_k) = 0$ and $\lim_k(\beta_k/\zeta_k) = 0$. Letting $d > \max(d_1, d_2)$, it follows $\sum_k(\gamma_k/\zeta_k)^d < \infty$ and $\sum_k(\beta_k/\zeta_k)^d < \infty$. Further,

$$\begin{aligned}\sum_k (\max(\gamma_k, \beta_k)/\zeta_k)^d &= \sum_k (\max(\gamma_k/\zeta_k, \beta_k/\zeta_k))^d \\ &= \sum_k \max((\gamma_k/\zeta_k)^d, (\beta_k/\zeta_k)^d) \\ &\leq \sum_k (\gamma_k/\zeta_k)^d + \sum_k (\beta_k/\zeta_k)^d \\ &< \infty.\end{aligned}$$

The second equality is due to the function $f(x) = x^d$ being monotonically increasing in the range $[0, \infty)$ when $d > 0$. The first inequality follows because both $\{(\gamma_k/\zeta_k)^d\}$ and $\{(\beta_k/\zeta_k)^d\}$ are positive sequences. ∎

A. Proof of Lemma VI.1:

Proof: Define $\hat{\boldsymbol{\theta}}_k = (\boldsymbol{\theta}_k, \mathbf{r}_k)$ to be the collection of all parameters in (39). We can write (39) as

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \zeta_k(\mathbf{h}_{\hat{\boldsymbol{\theta}}_k}(\mathbf{y}_k) - \mathbf{G}_{\hat{\boldsymbol{\theta}}_k}(\mathbf{y}_k)\mathbf{s}_k) + \zeta_k \boldsymbol{\Xi}_k \mathbf{s}_k. \quad (52)$$

We have

$$\begin{aligned}\|\hat{\boldsymbol{\theta}}_{k+1} - \hat{\boldsymbol{\theta}}_k\| &\leq \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\| + \|\mathbf{r}_{k+1} - \mathbf{r}_k\| \\ &\leq \beta_k F_k + \gamma_k F_k^r \\ &\leq \max(\beta_k, \gamma_k)(F_k + F_k^r).\end{aligned}$$

The last inequality is implied since $\beta_k > 0$, $\gamma_k > 0$, F_k and F_k^r are nonnegative processes. Combined with Lemma A.1, we can see Assumptions 3.1.(1–3) in [12] are satisfied. In addition, Assumptions 3.1.(4–10) in [12] are satisfied due to Assumptions A.(3–11). As a result, Thm. 3.2 in [12] holds and implies

$$\lim_k \|\bar{\mathbf{G}}(\hat{\boldsymbol{\theta}}_k)\mathbf{s}_k - \bar{\mathbf{h}}(\hat{\boldsymbol{\theta}}_k)\| = 0, \qquad \text{w.p.1.} \quad (53)$$

The left hand side of (53) is equivalent to the left hand side of the lemma. ∎

Appendix B
Proof of Theorem VI.6

We first present the following lemmas. We define the norm $\|\cdot\|$ of a matrix to be the norm of the column vector containing all of its elements.

Lemma B.1: Under iteration (27), we have

$$\begin{aligned}\|\mathbf{A}_{k+1} - \mathbf{A}_k\| &\leq \gamma_k F_k^A, \\ \|\mathbf{b}_{k+1} - \mathbf{b}_k\| &\leq \gamma_k F_k^b,\end{aligned}$$

for some processes $\{F_k^A\}$ and $\{F_k^b\}$ with bounded moments, where γ_k is the stepsize in (27).

Proof: According to (27), we have

$$\begin{aligned}&\mathbf{A}_{k+1} - \mathbf{A}_k \\ &= \gamma_k \left(\mathbf{z}_k(\boldsymbol{\phi}'_{\boldsymbol{\theta}_k}(\mathbf{x}_k, u_k) - \boldsymbol{\phi}'_{\boldsymbol{\theta}_{k+1}}(\mathbf{x}_{k+1}, u_{k+1})) - \mathbf{A}_k\right).\end{aligned}$$

Similar to Lemma VI.5 and because $\mathbf{z}_k$ has bounded moments and $\boldsymbol{\phi}_{\boldsymbol{\theta}} \in \mathcal{D}^{(2)}$, it can be verified that $\mathbf{A}_k$ has bounded moments. This establishes the first statement of the Lemma. We can prove the second statement of the Lemma for $\{\mathbf{b}_k\}$ in the same way given that the one-step reward function $g \in \mathcal{D}^{(2)}$, first by establishing that α_k has bounded moments. ∎

Lemma B.2: Suppose $\mathbf{f}(\cdot)$ is a *locally Lipschitz continuous* function on a domain $\mathcal{D}$. Let $\{\boldsymbol{v}_k\}$ be a sequence of random variables with bounded moments defined on $\mathcal{D}$ such that $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| \leq \gamma_k F_k$ for some $\{F_k\}$ with bounded moments w.p.1. Then $\|\mathbf{f}(\boldsymbol{v}_{k+1}) - \mathbf{f}(\boldsymbol{v}_k)\| \leq \gamma_k F_k^f$ for some $\{F_k^f\}$ with bounded moments w.p.1.

Proof: Since $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| \leq \gamma_k F_k$, it follows $\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k| < \infty$ w.p.1. Since $\{\boldsymbol{v}_k\}$ has bounded moments, $\boldsymbol{v}_k$ must be in a compact set w.p.1 for $\forall k$. Then, by Lipschitz continuity, $\|\mathbf{f}(\boldsymbol{v}_{k+1}) - \mathbf{f}(\boldsymbol{v}_k)\| \leq C\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| \leq \gamma_k C F_k$ for some constant C. The lemma can be proved by letting $F_k^f = C F_k$. ∎

Lemma B.3: Let $\boldsymbol{v} = \{\mathbf{A}, \mathbf{b}\}$ be a vector consisting of all elements in an $m \times m$ matrix $\mathbf{A}$ and a vector $\mathbf{b} \in \mathbb{R}^m$. The function $\mathbf{f}(\boldsymbol{v}) = \mathbf{A}^{-1}\mathbf{b}$ is *locally Lipschitz continuous* with respect to $\mathbf{A}$ and $\mathbf{b}$ on the domain $\mathcal{D} = \{\boldsymbol{v} : \det(\mathbf{A}) \geq \epsilon\}$, where ϵ is a positive constant.

Proof: Let $\mathbf{A}^a$ denote the adjoint matrix of $\mathbf{A}$. The function $\mathbf{f}^a(\boldsymbol{v}) = \mathbf{A}^a\mathbf{b}$ is locally Lipschitz continuous as it is a polynomial function, so $\|\mathbf{f}^a(\boldsymbol{v}_1) - \mathbf{f}^a(\boldsymbol{v}_2)\| \leq C\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|$ for some constant C and $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ that belong to a compact set. Since $\mathbf{A}^{-1} = \mathbf{A}^a/\det(\mathbf{A})$ and for $\boldsymbol{v}_1 = \{\mathbf{A}_1, \mathbf{b}_1\}$, $\boldsymbol{v}_2 = \{\mathbf{A}_2, \mathbf{b}_2\}$, we have

$$\begin{aligned}\|\mathbf{f}(\boldsymbol{v}_1) - \mathbf{f}(\boldsymbol{v}_2)\| &= \|\mathbf{A}_1^{-1}\mathbf{b}_1 - \mathbf{A}_2^{-1}\mathbf{b}_2\| \\ &= \|\mathbf{A}_1^a\mathbf{b}_1/\det(\mathbf{A}_1) - \mathbf{A}_2^a\mathbf{b}_2/\det(\mathbf{A}_2)\| \\ &\leq \frac{1}{\epsilon}\|\mathbf{A}_1^a\mathbf{b}_1 - \mathbf{A}_2^a\mathbf{b}_2\| \\ &= \frac{1}{\epsilon}\|\mathbf{f}^a(\boldsymbol{v}_1) - \mathbf{f}^a(\boldsymbol{v}_2)\| \\ &\leq \frac{C}{\epsilon}\|\boldsymbol{v}_1 - \boldsymbol{v}_2\|.\end{aligned}$$

So $\mathbf{f}(\boldsymbol{v}) = \mathbf{A}^{-1}\mathbf{b}$ must be locally Lipschitz continuous on the domain $\mathcal{D} = \{\boldsymbol{v} : \det(\mathbf{A}) > \epsilon\}$. ∎

A. Proof of Theorem VI.6

Proof: Recall that $\mathbf{V}(\mathbf{A})$ is the column vector stacking all columns in a matrix $\mathbf{A}$. Let $\boldsymbol{v}_k = (\mathbf{V}(\mathbf{A}_k), \mathbf{b}_k)$ where $\mathbf{A}_k$ and $\mathbf{b}_k$ are the iterates in (27). It follows

$$\begin{aligned}\|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\| &= \|\mathbf{A}_{k+1} - \mathbf{A}_k\| + \|\mathbf{b}_{k+1} - \mathbf{b}_k\| \\ &\leq \gamma_k(F_k^A + F_k^b).\end{aligned}$$

The last equality is due to Lemma B.1 and $F_k^A + F_k^b$ has bounded moments. Define the function $\mathbf{f}(\boldsymbol{v}_k) = \mathbf{A}_k^{-1}\mathbf{b}_k$, which implies $\mathbf{r}_k = \mathbf{f}(\mathbf{x}_k) = \mathbf{A}_k^{-1}\mathbf{b}_k$ when $\det(\mathbf{A}_k) \geq \epsilon$ by (28). The lemma can be easily proved by combining Lemma B.3 and Lemma B.2. ∎

References

[1] J. Wang and I. C. Paschalidis, "A Hessian actor-critic algorithm," in *Proc. 53rd IEEE Conf. Decision Control*, Los Angeles, CA, Dec. 2014, pp. 1131–1136.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[3] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Nashua, NH: Athena Scientific, 1996.

[4] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[5] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006.

[6] M. Khamassi, L. Lachèze, B. Girard, A. Berthoz, and A. Guillot, "Actor-critic models of reinforcement learning in the basal ganglia: From natural to artificial rats," *Adaptive Behavior*, vol. 13, no. 2, pp. 131–148, 2005.

[7] K. Samejima and T. Omori, "Adaptive internal state space construction method for reinforcement learning of a real-world agent," *Neural Networks*, vol. 12, pp. 1143–1155, 1999.

[8] G. Gajjar, S. Khaparde, P. Nagaraju, and S. Soman, "Application of actor-critic learning algorithm for optimal bidding problem of a GenCo," *IEEE Trans. Power Eng. Rev.*, vol. 18, no. 1, pp. 11–18, 2003.

[9] D. Bertsekas, V. Borkar, and A. Nedic, "Improved temporal difference methods with linear function approximation," *LIDS REPORT*, Tech. Rep. 2573, 2003.

[10] J. Boyan, "Least-squares temporal difference learning," in *Proc. 16th Int. Conf. Machine Learning*, 1999.

[11] S. Bradtke and A. Barto, "Linear least-squares algorithms for temporal difference learning," *Machine Learning*, vol. 22, no. 2, pp. 33–57, 1996.

[12] V. Konda, *Actor-Critic Algorihms*, Ph.D. dissertation, MIT, Cambridge, MA, 2002.

[13] A. Nedic and D. Bertsekas, "Least squares policy evaluation algorithms with linear function approximation," *Discrete Event Dynamic Syst.: Theory Appl.*, vol. 13, pp. 79–110, 2003.

[14] R. Moazzez-Estanjini, K. Li, and I. C. Paschalidis, "A least squares temporal difference actor-critic algorithm with applications to warehouse management," *Naval Research Logistics*, vol. 59, no. 3, pp. 197–211, 2012.

[15] X.-C. Ding, J. Wang, M. Lahijanian, I. C. Paschalidis, and C. Belta, "Temporal logic motion control using actor-critic methods," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, St. Paul, MN, May 14–18 2012, pp. 4687–4692.

[16] R. Moazzez-Estanjini, X.-C. Ding, M. Lahijanian, J. Wang, C. A. Belta, and I. C. Paschalidis, "Least squares temporal difference actor-critic methods with applications to robot motion control," in *Proc. 50th IEEE Conf. Decision Control*, Orlando, FL, Dec. 12–15 2011.

[17] J. Wang, X. Ding, M. Lahijanian, I. C. Paschalidis, and C. Belta, "Temporal logic motion control using actor-critic methods," *Int. J. Robot. Research*, vol. 34, no. 10, pp. 1329–1344, 2015.

[18] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[19] S. Kakade, "A natural policy gradient," *Advances Neural Inform. Processing Syst.*, vol. 14, pp. 1531–1538, 2001.

[20] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, pp. 1180–1190, 2008.

[21] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee, "Incremental natural actor-critic algorithms," *Neural Information Processing Systems (NIPS)*, 2007 .

[22] S. Richter, D. Aberdeen, and J. Yu, "Natural actor-critic for road traffic optimisation," in *Advances in Neural Information Processing Systems*, 2007, p. 1169.

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

[24] S. Kakade, "Optimizing average reward using discounted rewards," in *Computational Learning Theory*. Springer, 2001, pp. 605–615.

[25] N. Le Roux and A. Fitzgibbon, "A fast natural Newton method," in *Proc. 27th Int. Conf. Machine Learning*. Citeseer, 2010.

[26] S. P. Meyn, R. L. Tweedie, and P. W. Glynn, *Markov Chains and Stochastic Stability*. Cambridge University Press Cambridge, MA, 2009, vol. 2.

[27] P. Marbach and J. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Trans. Autom. Control*, vol. 46, pp. 191–209, 2001.

[28] I. C. Paschalidis, K. Li, and R. M. Estanjini, "An actor-critic method using least squares temporal difference learning," in *Proc. Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 2564–2569.

[29] H. Yu, "*Approximate Solution Methods for Partially Observable Markov and Semi-Markov Decision Processes*," Ph.D. dissertation, 2006.

[30] H. Yu and D. P. Bertsekas, "Convergence results for some temporal difference methods based on least squares," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1515–1531, Jul. 2009.

[31] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. Springer, 2008.

[32] V. R. Konda and J. N. Tsitsiklis, "Appendix to "on actor-critic algorithms"." [Online]. Available: http://web.mit.edu/jnt/www/Papers/J094-03-kon-actors-app.pdf.

Jing (Conan) Wang received the B.E. degree in electrical and information engineering from Huazhong University of Science and Technology, Huazhong, China, in 2010 and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2015.

He is now a Software Engineer at Google, Inc. His research interests include interest-based advertising, cyber security, and approximate dynamic programming.

Ioannis Ch. Paschalidis (M'96–SM'06–F'14) received the M.S. and Ph.D. degrees both in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1993 and 1996, respectively.

In September 1996 he joined Boston University where he has been ever since. He is a Professor and Distinguished Faculty Fellow at Boston University with appointments in the Department of Electrical and Computer Engineering, the Division of Systems Engineering, and the Department of Biomedical Engineering. He is the Director of the Center for Information and Systems Engineering (CISE). He has held visiting appointments with MIT and Columbia University, New York, NY, USA. His current research interests lie in the fields of systems and control, networking, applied probability, optimization, operations research, computational biology, and medical informatics.

Dr. Paschalidis received the NSF CAREER award (2000), several best paper and best algorithmic performance awards, and a 2014 IBM/IEEE Smarter Planet Challenge Award. He was an invited participant at the 2002 Frontiers of Engineering Symposium, organized by the U.S. National Academy of Engineering and the 2014 U.S. National Academies Keck Futures Initiative (NAFKI) Conference. He is the inaugural Editor-in-Chief of the IEEE Transactions on Control of Network Systems.