In Proc. of European Control Conference 2007, Kos, Greece

Q-Learning Algorithms for Optimal Stopping Based on Least Squares

Huizhen Yu and Dimitri P. Bertsekas

Abstract— **We consider the solution of discounted optimal stopping problems using linear function approximation methods. A Q-learning algorithm for such problems, proposed by Tsitsiklis and Van Roy, is based on the method of temporal differences and stochastic approximation. We propose alternative algorithms, which are based on projected value iteration ideas and least squares. We prove the convergence of some of these algorithms and discuss their properties.**

I. INTRODUCTION

Optimal stopping problems are a special case of Markovian decision problems where the system evolves according to a discrete-time stochastic system equation, until an explicit stopping action is taken. At each state, there are two choices: either to stop and incur a state-dependent stopping cost, or to continue and move to a successor state according to some transition probabilities and incur a state-dependent continuation cost. Once the stopping action is taken, no further costs are incurred. The objective is to minimize the expected value of the total discounted cost. Examples are classical problems, such as search, and sequential hypothesis testing, as well as recent applications in finance and the pricing of derivative financial instruments (see Tsitsiklis and Van Roy [1], Barraquand and Martineau [2], Longstaff and Schwartz [3]).

The problem can be solved in principle by dynamic programming (DP for short), but we are interested in problems with large state spaces where the DP solution is practically infeasible. It is then natural to consider approximate DP techniques where the optimal cost function or the Q-factors of the problem are approximated with a function from a chosen parametric class. Generally, cost function approximation methods are theoretically sound (i.e., are provably convergent) only for the single-policy case, where the cost function of a fixed stationary policy is evaluated. However, for the stopping problem of this paper, Tsitsiklis and Van Roy [1] introduced a linear function approximation to the optimal Q-factors, which they prove to be the unique solution of a projected form of Bellman's equation. While in general this equation may not have a solution, this difficulty does not occur in optimal stopping problems thanks to a critical fact: the mapping defining the Q-factors is a contraction mapping with respect to the weighted Euclidean norm corresponding to the steady-state distribution of the associated Markov chain. For textbook analyses, we refer to Bertsekas and Tsitsiklis [4], Section 6.8, and Bertsekas [5], Section 6.4.

The algorithm of Tsitsiklis and Van Roy is based on single trajectory simulation, and ideas related to the temporal differences method of Sutton [6], and relies on the contraction property just mentioned. We propose a new algorithm, which is also based on single trajectory simulation and relies on the same contraction property, but uses different algorithmic ideas. It may be viewed as a fixed point iteration for solving the projected Bellman equation, and it relates to the least squares policy evaluation (LSPE) method first proposed by Bertsekas and Ioffe [7] and subsequently developed by Nedić and Bertsekas [8], Bertsekas, Borkar, and Nedić [8], and Yu and Bertsekas [9] (see also the books [4] and [5]). We prove the convergence of our method for finite-state models. We also discuss variants of the method and prove convergence of some of them. We refer to an extended version of this paper [10] for the details of the corresponding convergence analysis.

The paper is organized as follows. In Section II, we introduce the optimal stopping problem, and we derive the associated contraction properties of the mapping that defines Q-learning. In Section III, we describe our LSPE-like algorithm, and we prove its convergence. We also discuss the convergence rate of the algorithm, and we provide a comparison with another algorithm that is related to the least squares temporal differences (LSTD) method, proposed by Bradtke and Barto [11], and further developed by Boyan [12]. In Section IV, we describe some variants of the algorithm, which involve a reduced computational overhead per iteration, and discuss the relation of our algorithms with the recent algorithm by Choi and Van Roy [13], which can be used to solve the same optimal stopping problem. In this section, we also give without proof a convergence result for some of the variants of Section IV. A computational comparison of our methods with other algorithms for the optimal stopping problem is beyond the scope of the present paper. However, our analysis and the available results using least squares methods (Bradtke and Barto [11], Bertsekas and Ioffe [7], Boyan [12], Bertsekas, Borkar, and Nedić [14], Choi and Van Roy [13]) clearly suggest a superior performance to the algorithm of Tsitsiklis and Van Roy [1], and likely an improved convergence rate over the method of Choi and Van Roy [13], at the expense of some additional overhead per iteration.

Huizhen Yu is with the Helsinki Institute for Information Technology, University of Helsinki, Finland janey.yu@cs.helsinki.fi

Dimitri Bertsekas is with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA 02139, USA dimitrib@mit.edu

II. *Q*-LEARNING FOR OPTIMAL STOPPING PROBLEMS

We are given a Markov chain with state space $\{1,\dots,n\}$, described by transition probabilities p_{ij}. We assume that the states form a single recurrent class, so the chain has a steady-state distribution vector $\pi = \big(\pi(1),\dots,\pi(n)\big)$ with $\pi(i) > 0$ for all states i. Given the current state i, we assume that we have two options: to stop and incur a cost $c(i)$, or to continue and incur a cost $g(i,j)$, where j is the next state (there is no control to affect the corresponding transition probabilities). The problem is to minimize the associated α-discounted infinite horizon cost, where $\alpha \in (0,1)$.

For a given state i, we associate a Q-factor with each of the two possible decisions. The Q-factor for the decision to stop is equal to $c(i)$. The Q-factor for the decision to continue is denoted by $Q(i)$. The optimal Q-factor for the decision to continue, denoted by Q^*, relates to the optimal cost function J^* of the stopping problem by

$$Q^*(i) = \sum_{j=1}^{n} p_{ij}\big(g(i,j) + \alpha J^*(j)\big), \qquad i = 1,\dots,n,$$

and

$$J^*(i) = \min\big\{c(i), Q^*(i)\big\}, \qquad i = 1,\dots,n.$$

The value $Q^*(i)$ is equal to the cost of choosing to continue at the initial state i and following an optimal policy afterwards. The function Q^* satisfies Bellman's equation

$$Q^*(i) = \sum_{j=1}^{n} p_{ij}\Big(g(i,j) + \alpha \min\big\{c(j), Q^*(j)\big\}\Big), \quad i = 1,\dots,n. \tag{1}$$

Once the Q-factors $Q^*(i)$ are calculated, an optimal policy can be implemented by stopping at state i if and only if $c(i) \le Q^*(i)$.

The Q-learning algorithm (Watkins [15]) is

$$Q(i) := Q(i) + \gamma\big(g(i,j) + \alpha \min\big\{c(j), Q(j)\big\} - Q(i)\big),$$

where i is the state at which we update the Q-factor, j is a successor state, generated randomly according to the transition probabilities p_{ij}, and γ is a small positive stepsize, which diminishes to 0 over time. The convergence of this algorithm is addressed by the general theory of Q-learning (see Watkins and Dayan [16], and Tsitsiklis [17]). However, for problems where the number of states n is large, this algorithm is impractical.

Let us now consider the approximate evaluation of $Q^*(i)$. We introduce the mapping $F : \Re^n \mapsto \Re^n$ given by

$$(FQ)(i) = \sum_{j=1}^{n} p_{ij}\big(g(i,j) + \alpha \min\big\{c(j), Q(j)\big\}\big), \ i = 1,\dots,n.$$

We denote by FQ or $F(Q)$ the vector whose components are $(FQ)(i)$, $i = 1,\dots,n$. By (1), the optimal Q-factor for the choice to continue, Q^*, is a fixed point of F, and it is the unique fixed point because F is a sup-norm contraction mapping.

For the approximation considered here, it turns out to be very important that F is also a Euclidean contraction. Let $\|\cdot\|_\pi$ be the weighted Euclidean norm associated with the steady-state probability vector π, i.e.,

$$\|v\|_\pi^2 = \sum_{i=1}^{n} \pi(i)\big(v(i)\big)^2.$$

It has been shown by Tsitsiklis and Van Roy [1] (see also Bertsekas and Tsitsiklis [4], Section 6.8.4) that F is a contraction with respect to this norm. For purposes of easy reference, we include the proof.

Lemma 1: The mapping F is a contraction with respect to $\|\cdot\|_\pi$, with modulus α.

Proof: For any two vectors Q and $\overline{Q}$, we have

$$\begin{aligned}\big|(FQ)(i) - (F\overline{Q})(i)\big| &\le \alpha \sum_{j=1}^{n} p_{ij}\big|\min\big\{c(j), Q(j)\big\} \\ &\qquad - \min\big\{c(j), \overline{Q}(j)\big\}\big| \\ &\le \alpha \sum_{j=1}^{n} p_{ij}\big|Q(j) - \overline{Q}(j)\big|,\end{aligned}$$

or, in vector notation,

$$|FQ - F\overline{Q}| \le \alpha P|Q - \overline{Q}|,$$

where $|x|$ denotes a vector whose components are the absolute values of the components of x. Hence,

$$\|FQ - F\overline{Q}\|_\pi \le \alpha \big\|P|Q - \overline{Q}|\,\big\|_\pi \le \alpha\|Q - \overline{Q}\|_\pi,$$

where the last inequality follows from the relation $\|PJ\|_\pi \le \|J\|_\pi$, which holds for every vector J (see Tsitsiklis and Van Roy [18] or Bertsekas and Tsitsiklis [4], Lemma 6.4). ■

We consider Q-factor approximations using a linear approximation architecture

$$\tilde{Q}(i,r) = \phi(i)'r,$$

where $\phi(i)$ is an s-dimensional feature vector associated with state i. (In our notation, all vectors are viewed as column vectors, and prime denotes transposition.) We also write the vector

$$\tilde{Q}_r = \big(\tilde{Q}(1,r),\dots,\tilde{Q}(n,r)\big)'$$

in the compact form

$$\tilde{Q}_r = \Phi r,$$

where Φ is the $n \times s$ matrix whose rows are $\phi(i)'$, $i = 1,\dots,n$. We assume that Φ has rank s, and we denote by Π the projection mapping with respect to $\|\cdot\|_\pi$ on the subspace

$$S = \{\Phi r \mid r \in \Re^s\},$$

i.e., for all $J \in \Re^n$,

$$\Pi J = \arg\min_{\hat{J} \in S} \|J - \hat{J}\|_\pi.$$

Because F is a contraction with respect to $\|\cdot\|_\pi$ with modulus α, and Π is nonexpansive, the mapping ΠF is a contraction with respect to $\|\cdot\|_\pi$ with modulus α. Therefore, the mapping ΠF has a unique fixed point within the subspace S, which (in view of the rank assumption on Φ) can be uniquely represented as Φr^*. Thus r^* is the unique solution of the equation

$$\Phi r^* = \Pi F(\Phi r^*).$$

Tsitsiklis and Van Roy [1] show that the error of this Q-factor approximation can be bounded by

$$\|\Phi r^* - Q^*\|_\pi \leq \frac{1}{\sqrt{1-\alpha^2}} \|\Pi Q^* - Q^*\|_\pi.$$

Furthermore, if we implement a policy μ that stops at state i if and only if $c(i) \leq \phi(i)'r^*$, then the cost of this policy, denoted by J_μ, satisfies

$$\sum_{i=1}^n \pi(i)\big(J_\mu(i) - J^*(i)\big) \leq \frac{2}{(1-\alpha)\sqrt{1-\alpha^2}} \|\Pi Q^* - Q^*\|_\pi.$$

These bounds indicate that if Q^* is close to the subspace S spanned by the basis functions, then the approximate Q-factor and its associated policy will also be close to the optimal.

The contraction property of ΠF suggests the fixed point iteration

$$\Phi r_{k+1} = \Pi F(\Phi r_k),$$

which in the related contexts of policy evaluation for discounted and average cost problems (see [14], [9], [5]) is known as *projected value iteration* [to distinguish it from the value iteration method, which is $Q_{k+1} = F(Q_k)$]; see Fig. 1. This iteration converges to the unique fixed point Φr^* of ΠF, but is not easily implemented because the dimension of the vector $F(\Phi r_k)$ is potentially very large. In the policy evaluation context, a simulation-based implementation of the iteration has been proposed, which does not suffer from this difficulty, because it uses simulation samples of the cost of various states in a least-squares type of parametric approximation of the value iteration method. This algorithm is known as least squares policy evaluation (LSPE), and can be conceptually viewed as taking the form

$$\Phi r_{k+1} = \Pi F(\Phi r_k) + \varepsilon_k,$$

where ε_k is simulation noise which diminishes to 0 (with probability 1) as $k \to \infty$ (see Fig. 1). The algorithm to be introduced in the next section admits a similar conceptual interpretation, and its analysis has much in common with the analysis given in [14], [5] for the case of single-policy evaluation. In fact, if the stopping option was not available [or equivalently if $c(i)$ is so high that it is never optimal to stop], our Q-learning algorithm would coincide with the LSPE algorithm for approximate evaluation of the discounted cost function of a fixed stationary policy. Let us also note that LSPE (like the temporal differences method) is actually a family of methods parameterized by a scalar $\lambda \in [0,1]$. Our Q-learning algorithm of the next section corresponds to LSPE(0), the case where $\lambda = 0$; we do not have a convenient Q-learning algorithm that parallels LSPE(λ) for $\lambda > 0$.

III. A LEAST SQUARES Q-LEARNING ALGORITHM

A. Algorithm

We generate a single[1] infinitely long simulation trajectory $(x_0, x_1, \dots)$ corresponding to an unstopped system, i.e., using the transition probabilities p_{ij}. Our algorithm starts with an initial guess r_0, and generates a parameter vector sequence $\{r_t\}$. Following the transition (x_t, x_{t+1}), we form the following least squares problem at each time t,

$$\min_{r \in \Re^s} \sum_{k=0}^t \Big(\phi(x_k)'r - g(x_k, x_{k+1}) - \alpha \min\big\{c(x_{k+1}), \phi(x_{k+1})'r_t\big\}\Big)^2, \tag{2}$$

whose solution is

$$\hat{r}_{t+1} = \left(\sum_{k=0}^t \phi(x_k)\phi(x_k)'\right)^{-1} \sum_{k=0}^t \phi(x_k)\Big(g(x_k, x_{k+1}) + \alpha \min\big\{c(x_{k+1}), \phi(x_{k+1})'r_t\big\}\Big). \tag{3}$$

Then we set

$$r_{t+1} = r_t + \gamma(\hat{r}_{t+1} - r_t), \tag{4}$$

where γ is some fixed constant stepsize, whose range will be given later.[2]

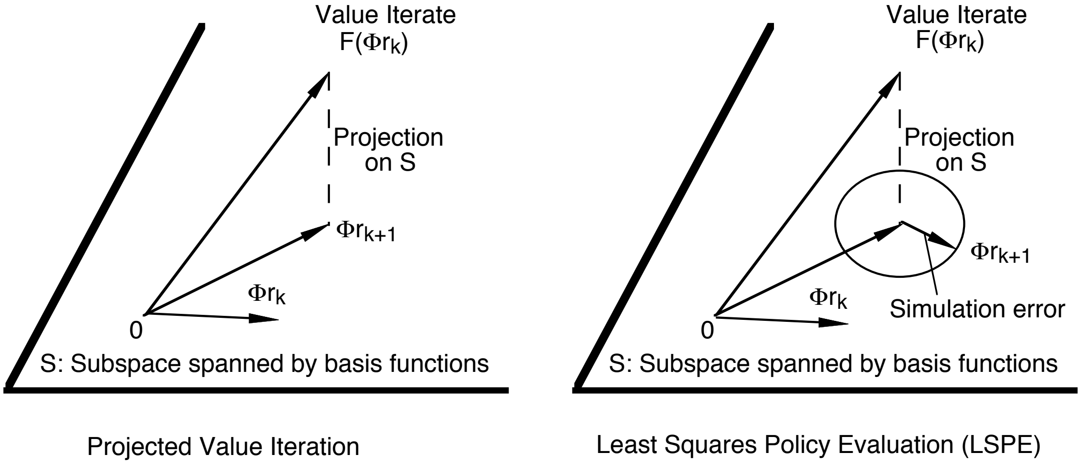
This algorithm is related to the LSPE(0) algorithm, which is used for the approximate evaluation of a single stationary policy of a discounted Markovian decision problem, and is analyzed by Bertsekas and Ioffe [7], Nedić and Bertsekas [8], Bertsekas, Borkar, and Nedić [14], and Yu and Bertsekas [9] (see also the recent book by Bertsekas [5], Chapter 6). In particular, if there were no stopping action (or equivalently if the stopping costs are so large that they are inconsequential), then, for $\gamma = 1$, the algorithm (3) becomes

$$r_{t+1} = \left(\sum_{k=0}^t \phi(x_k)\phi(x_k)'\right)^{-1} \sum_{k=0}^t \phi(x_k)\Big(g(x_k, x_{k+1}) + \alpha\phi(x_{k+1})'r_t\Big), \tag{5}$$

and is identical to the LSPE(0) algorithm for evaluating the policy that never stops. On the other hand, we note that the least squares Q-learning algorithm (3) has much higher computation overhead than the LSPE(0) algorithm (5) for evaluating this policy. In the process of updating r_t via (3), we can compute the matrix $\big(\frac{1}{t+1}\sum_{k=0}^t \phi(x_k)\phi(x_k)'\big)^{-1}$ and the vector $\frac{1}{t+1}\sum_{k=0}^t \phi(x_k)g(x_k, x_{k+1})$ iteratively and efficiently as in (5). The terms $\min\big\{c(x_{k+1}), \phi(x_{k+1})'r_t\big\}$, however, need to be recomputed for all the samples x_{k+1}, $k < t$. Intuitively, this computation corresponds to repartitioning the states into

[1] Multiple independent infinitely long trajectories can also be used similarly.

[2] We ignore the issues associated with the invertibility of the matrix in (3). They can be handled, for example, by adding a small positive multiple of the identity to the matrix if it is not invertible.



Projected Value Iteration

Least Squares Policy Evaluation (LSPE)

Fig. 1. A conceptual view of projected value iteration and its simulation-based implementation.

those at which to stop and those at which to continue, based on the current approximate Q-factors Φr_t. In Section IV, we will discuss how to reduce this extra overhead.

We will prove that the sequence $\{\Phi r_t\}$ generated by the least squares Q-learning algorithm (3) asymptotically converges to the unique fixed point of ΠF. The idea of the proof is to show that the algorithm can be written as a damped version of the iteration $\Phi r_{t+1} = \widehat{\Pi}_t \widehat{F}_t(\Phi r_t)$, where $\widehat{\Pi}_t$ and $\widehat{F}_t$ approximate Π and F, respectively, within simulation error that asymptotically diminishes to 0 with probability 1.

B. Convergence Proof

The iteration (3) can be written equivalently as

$$\hat{r}_{t+1} = \left(\sum_{i=1}^{n} \hat{\pi}_t(i) \phi(i) \phi(i)' \right)^{-1} \sum_{i=1}^{n} \hat{\pi}_t(i) \phi(i) \left(\hat{g}_t(i) + \alpha \sum_{j=1}^{n} \hat{\pi}_t(j|i) \min \left\{ c(j), \phi(j)' r_t \right\} \right),$$

where $\hat{\pi}_t(i)$ and $\hat{\pi}_t(j|i)$ are the empirical frequencies defined by

$$\hat{\pi}_t(i) = \frac{\sum_{k=0}^{t} \delta(x_k = i)}{t+1}, \quad \hat{\pi}_t(j|i) = \frac{\sum_{k=0}^{t} \delta(x_k = i, x_{k+1} = j)}{\sum_{k=0}^{t} \delta(x_k = i)},$$

with $\delta(\cdot)$ being the indicator function, and $\hat{g}_t$ is the empirical mean of the per-stage costs:

$$\hat{g}_t(i) = \frac{\sum_{k=0}^{t} g(x_k, x_{k+1}) \delta(x_k = i)}{\sum_{k=0}^{t} \delta(x_k = i)}.$$

[In the case where $\sum_{k=0}^{t} \delta(x_k = i) = 0$, we define $\hat{g}_t(i) = 0, \hat{\pi}_t(j|i) = 0$ by convention.] In a more compact notation,

$$\Phi \hat{r}_{t+1} = \widehat{\Pi}_t \widehat{F}_t(\Phi r_t), \tag{6}$$

where the mappings $\widehat{\Pi}_t$ and $\widehat{F}_t$ are simulation-based approximations to Π and F, respectively:

$$\widehat{\Pi}_t = \Phi(\Phi' \widehat{D}_t \Phi)^{-1} \Phi' \widehat{D}_t, \qquad \widehat{D}_t = \operatorname{diag}(\ldots, \hat{\pi}_t(i), \ldots),$$
$$\widehat{F}_t J = \hat{g}_t + \alpha \tilde{P}_t \min\{c, J\}, \ \forall J \in \Re^n, \quad \left(\tilde{P}_t\right)_{ij} = \hat{\pi}_t(j|i).$$

With a stepsize γ, the least squares Q-learning iteration (4) is written as

$$\Phi r_{t+1} = (1-\gamma)\Phi r_t + \gamma \widehat{\Pi}_t \widehat{F}_t(\Phi r_t). \tag{7}$$

By ergodicity of the Markov chain, we have

$$\hat{\pi}_t \to \pi, \quad \tilde{P}_t \to P, \quad \text{and} \quad \hat{g}_t \to g,$$

with probability 1, where g denotes the expected per-stage cost vector with $\sum_{j=1}^{n} p_{ij} g(i,j)$ as the i-th component.

For each t, denote the invariant distribution of $\tilde{P}_t$ by $\tilde{\pi}_t$. We now have three distributions, $\pi, \hat{\pi}_t, \tilde{\pi}_t$, which define, respectively, three weighted Euclidean norms, $\|\cdot\|_\pi, \|\cdot\|_{\hat{\pi}_t}, \|\cdot\|_{\tilde{\pi}_t}$. The mappings we consider are non-expansive or contraction mappings with respect to one of these norms. In particular:

- the mapping $\widehat{\Pi}_t$ is non-expansive with respect to $\|\cdot\|_{\hat{\pi}_t}$ (since $\widehat{\Pi}_t$ is projection with respect to $\|\cdot\|_{\hat{\pi}_t}$), and
- the mapping $\widehat{F}_t$ is a contraction, with modulus α, with respect to $\|\cdot\|_{\tilde{\pi}_t}$ (the proof of Lemma 1 can be used to show this).

We have the following facts, each being a consequence of the ones preceding it:

(i) $\hat{\pi}_t, \tilde{\pi}_t \to \pi$ with probability 1.

(ii) For any $\varepsilon > 0$ and a sample trajectory with converging sequences $\hat{\pi}_t, \tilde{\pi}_t$, there exists a time $\bar{t}$ such that for all $t \geq \bar{t}$ and all states i

$$\frac{1}{1+\varepsilon} \leq \frac{\hat{\pi}_t(i)}{\pi(i)} \leq 1+\varepsilon, \qquad \frac{1}{1+\varepsilon} \leq \frac{\tilde{\pi}_t(i)}{\pi(i)} \leq 1+\varepsilon,$$
$$\frac{1}{1+\varepsilon} \leq \frac{\tilde{\pi}_t(i)}{\hat{\pi}(i)} \leq 1+\varepsilon.$$

(iii) Under the condition of (ii), for any $J \in \Re^n$, we have

$$\|J\|_\pi \leq (1+\varepsilon)\|J\|_{\hat{\pi}_t}, \qquad \|J\|_{\hat{\pi}_t} \leq (1+\varepsilon)\|J\|_{\tilde{\pi}_t},$$
$$\|J\|_{\tilde{\pi}_t} \leq (1+\varepsilon)\|J\|_\pi,$$

for all t sufficiently large.

Fact (iii) implies the contraction of $\widehat{\Pi}_t \widehat{F}_t$ with respect to $\|\cdot\|_\pi$, as shown in the following lemma.

Lemma 2: Let $\hat{\alpha} \in (\alpha, 1)$. Then, with probability 1, $\widehat{\Pi}_t \widehat{F}_t$ is a $\|\cdot\|_\pi$-contraction mapping with modulus $\hat{\alpha}$ for all t sufficiently large.

Proof: Consider a simulation trajectory from the set of probability 1 for which $\tilde{P}_t \to P$ and $\hat{\pi}_t, \tilde{\pi}_t \to \pi$. Fix an $\varepsilon > 0$. For any functions J_1 and J_2, using fact (iii) above and the non-expansiveness and contraction properties of $\widehat{\Pi}_t$ and $\widehat{F}_t$, respectively, we have for t sufficiently large,

$$\begin{aligned}\|\widehat{\Pi}_t\widehat{F}_t J_1 - \widehat{\Pi}_t\widehat{F}_t J_2\|_\pi &\leq (1+\varepsilon)\|\widehat{\Pi}_t\widehat{F}_t J_1 - \widehat{\Pi}_t\widehat{F}_t J_2\|_{\hat{\pi}_t} \\ &\leq (1+\varepsilon)\|\widehat{F}_t J_1 - \widehat{F}_t J_2\|_{\hat{\pi}_t} \\ &\leq (1+\varepsilon)^2\|\widehat{F}_t J_1 - \widehat{F}_t J_2\|_{\tilde{\pi}_t} \\ &\leq (1+\varepsilon)^2\alpha\|J_1 - J_2\|_{\tilde{\pi}_t} \\ &\leq (1+\varepsilon)^3\alpha\|J_1 - J_2\|_\pi.\end{aligned}$$

Thus, by letting ε be such that $(1+\varepsilon)^3\alpha < \hat{\alpha} < 1$, we see that $\widehat{\Pi}_t\widehat{F}_t$ is a $\|\cdot\|_\pi$-contraction mapping with modulus $\hat{\alpha}$ for all t sufficiently large. ■

Proposition 1: For any constant stepsize $\gamma \in (0, \frac{2}{1+\alpha})$, r_t converges to r^* with probability 1, as $t \to \infty$.

Proof: We choose $\bar{t}$ such that for all $t \geq \bar{t}$, the contraction property of Lemma 2 applies. We have for such t,

$$\begin{aligned}\|\Phi r_{t+1} - \Phi r^*\|_\pi &= \Big\|(1-\gamma)\,(\Phi r_t - \Phi r^*) \\ &\qquad + \gamma\big(\widehat{\Pi}_t\widehat{F}_t(\Phi r_t) - \Pi F(\Phi r^*)\big)\Big\|_\pi \\ &\leq |1-\gamma|\,\|\Phi r_t - \Phi r^*\|_\pi \\ &\quad + \gamma\|\widehat{\Pi}_t\widehat{F}_t(\Phi r_t) - \widehat{\Pi}_t\widehat{F}_t(\Phi r^*)\|_\pi \\ &\quad + \gamma\|\widehat{\Pi}_t\widehat{F}_t(\Phi r^*) - \Pi F(\Phi r^*)\|_\pi \\ &\leq (|1-\gamma| + \gamma\hat{\alpha})\,\|\Phi r_t - \Phi r^*\|_\pi + \gamma\varepsilon_t, \quad (8)\end{aligned}$$

where

$$\varepsilon_t = \|\widehat{\Pi}_t\widehat{F}_t(\Phi r^*) - \Pi F(\Phi r^*)\|_\pi.$$

Because $\|\widehat{\Pi}_t\widehat{F}_t(\Phi r^*) - \Pi F(\Phi r^*)\|_\pi \to 0$, we have $\varepsilon_t \to 0$. Thus, for $\gamma \leq 1$, since

$$(1 - \gamma + \gamma\hat{\alpha}) < 1,$$

it follows that $\Phi r_t \to \Phi r^*$, or equivalently, $r_t \to r^*$, with probability 1. Similarly, based on (8), in order that $\|\Phi r_{t+1} - \Phi r^*\|_\pi$ converges to 0 under a stepsize $\gamma > 1$, it is sufficient that $\gamma - 1 + \gamma\hat{\alpha} < 1$, or equivalently,

$$\gamma < \frac{2}{1+\hat{\alpha}}.$$

Hence Φr_t converges to Φr^* for the stepsize $\gamma \in (0, \frac{2}{1+\alpha})$. ■

Note that the range of stepsizes for which convergence was shown includes $\gamma = 1$.

Remark 1: We can interpret the iteration of the least squares Q-learning algorithm, with the unit stepsize, for instance, as the deterministic fixed point iteration $\Pi F(\Phi r_t)$ plus an asymptotically diminishing stochastic disturbance (see Fig. 1). In particular,

$$\begin{aligned}&\widehat{\Pi}_t\widehat{F}_t(\Phi r_t) - \Pi F(\Phi r_t) \\ &\qquad = (\widehat{\Pi}_t\hat{g}_t - \Pi g) + \alpha(\widehat{\Pi}_t\tilde{P}_t - \Pi P)\min\{c, \Phi r_t\},\end{aligned}$$

so

$$\begin{aligned}\big\|\widehat{\Pi}_t\widehat{F}_t(\Phi r_t) - \Pi F(\Phi r_t)\big\| &\leq \|\widehat{\Pi}_t - \Pi\|\,\|\hat{g}_t\| + \|\Pi\|\,\|\hat{g}_t - g\| \\ &\quad + \alpha\|\widehat{\Pi}_t\tilde{P}_t - \Pi P\|\,\big\|\min\{c, \Phi r_t\}\big\|,\end{aligned}$$

where $\|\cdot\|$ is any norm. Since Φr_t is bounded with probability 1, the bound on the right-hand side can be seen to asymptotically diminish to 0.

C. Comparison to an LSTD Analogue

A natural alternative approach to finding r^* that satisfies $\Phi r^* = \Pi F(\Phi r^*)$ is to replace Π and F with asymptotically convergent approximations. In particular, let $\tilde{r}_{t+1}$ be the solution of $\Phi r = \widehat{\Pi}_t\widehat{F}_t(\Phi r)$, i.e.,

$$\Phi\tilde{r}_{t+1} = \widehat{\Pi}_t\widehat{F}_t(\Phi\tilde{r}_{t+1}), \qquad t = 0, 1, \ldots.$$

With probability 1 the solutions exist for t sufficiently large by Lemma 2. The conceptual algorithm that generates the sequence $\{\tilde{r}_t\}$ may be viewed as the analogue of the LSTD method, proposed by Bradtke and Barto [11], and further developed by Boyan [12] (see also the text by Bertsekas [5], Chapter 6). For the optimal stopping problem this is not a viable algorithm because it involves solution of a nonlinear equation. It is introduced here as a vehicle for interpretation of our least squares Q-learning algorithm (2)-(4).

In particular, we note that $\tilde{r}_{t+1}$ is the solution of the equation

$$\begin{aligned}\tilde{r}_{t+1} = \arg\min_{r \in \Re^s} \sum_{k=0}^{t} \Big(&\phi(x_k)'r - g(x_k, x_{k+1}) \\ &- \alpha\min\big\{c(x_{k+1}), \phi(x_{k+1})'\tilde{r}_{t+1}\big\}\Big)^2, \quad (9)\end{aligned}$$

so it is the fixed point of the "arg min" mapping in the right-hand side of the above equation. On the other hand, the least squares Q-learning algorithm (2)-(4), with stepsize $\gamma = 1$, that generates r_{t+1} can be viewed as a single iteration of a fixed point algorithm that aims to find $\tilde{r}_{t+1}$, starting from r_t. This relation can be quantified further. Using an argument similar to the one used in [9] for evaluating the optimal asymptotic convergence rate of LSPE, it can be shown that with any stepsize in the range $(0, \frac{2}{1+\alpha})$, the LSPE-like update Φr_t converges to the LSTD-like update $\Phi\tilde{r}_t$ asymptotically at the rate of $O(t)$ [while we expect both to converge to Φr^* at a slower rate $O(\sqrt{t})$]. We state this in the following proposition, the proof of which can be found in [10].

Proposition 2: For any constant stepsize $\gamma \in (0, \frac{2}{1+\alpha})$, $t(\Phi r_t - \Phi\tilde{r}_t)$ is bounded with probability 1.

IV. VARIANTS WITH REDUCED OVERHEAD PER ITERATION

At each iteration of the least squares Q-learning algorithm (3), (4), while updating r_t, it is necessary to recompute the terms $\min\{c(x_{k+1}), \phi(x_{k+1})'r_t\}$ for all the samples x_{k+1}, $k < t$. Intuitively, this corresponds to repartitioning the sampled states into those at which to stop and those at which to continue based on the most recent approximate Q-factors Φr_t. In this section we discuss some variants of the algorithm that aim to reduce this computation.

A. First Variant

A simple way to reduce the overhead in iteration (3) is to forgo the repartitioning just mentioned. Thus, in this variant we replace the terms $\min\{c(x_{k+1}), \phi(x_{k+1})' r_t\}$ by $\tilde{q}(x_{k+1}, r_t)$, given by

$$\tilde{q}(x_{k+1}, r_t) = \begin{cases} c(x_{k+1}) & \text{if } k \in K, \\ \phi(x_{k+1})' r_t & \text{if } k \notin K, \end{cases}$$

where $K = \{k \mid c(x_{k+1}) \le \phi(x_{k+1})' r_k\}$ is the set of states to stop based on the (earlier) approximate Q-factors Φr_k, rather than the (most recent) approximate Q-factors Φr_t. In particular, we replace the term

$$\sum_{k=0}^{t} \phi(x_k) \min\{c(x_{k+1}), \phi(x_{k+1})' r_t\}$$

in (3) with

$$\begin{aligned} &\sum_{k=0}^{t} \phi(x_k) \tilde{q}(x_{k+1}, r_t) \\ &\quad = \sum_{k \le t, k \in K} \phi(x_k) c(x_{k+1}) + \sum_{k \le t, k \notin K} \phi(x_k) \phi(x_{k+1})' r_t, \end{aligned}$$

which can be efficiently updated at each time t.

Some other similar variants are possible, which employ a limited form of repartitioning the states into those to stop and those to continue. For example, one may repartition only the sampled states within a time window of the m most recent time periods. In particular, in the preceding calculation, instead of the set K, we may use at time t the set

$$\begin{aligned} K_t = &\{k \mid k \in K_{t-1}, k < t - m\} \\ &\cup \{k \mid t - m \le k \le t, c(x_{k+1}) \le \phi(x_{k+1})' r_t\}, \end{aligned}$$

starting with $K_0 = \{0\}$. Here $m = \infty$ corresponds to the algorithm of the preceding section, while $m = 1$ corresponds to the algorithm of the preceding paragraph. Thus the overhead for repartitioning per iteration is proportional to m, and remains bounded.

An important observation is that in the preceding variations, if r_t converges, then asymptotically the terms $\min\{c(x_{k+1}), \phi(x_{k+1})' r_t\}$ and $\tilde{q}(x_{k+1}, r_t)$ coincide, and it can be seen that the limit of r_t must satisfy the equation $\Phi r = \Pi F(\Phi r)$, so it must be equal to the unique solution r^*. However, at present we have no proof of convergence of r_t.

B. Second Variant

Let us consider another variant, whereby we simply replace the terms $\min\{c(x_{k+1}), \phi(x_{k+1})' r_t\}$ in the least squares problem (2) with $\min\{c(x_{k+1}), \phi(x_{k+1})' r_k\}$. The idea is that for large k and t, these two terms may be close enough to each other, so that convergence may still be maintained. Thus we consider the iteration

$$\begin{aligned} r_{t+1} = \arg\min_{r \in \Re^s} \sum_{k=0}^{t} \Big(&\phi(x_k)' r - g(x_k, x_{k+1}) \\ &- \alpha \min\{c(x_{k+1}), \phi(x_{k+1})' r_k\} \Big)^2. \end{aligned} \tag{10}$$

This is a special case of an algorithm due to Choi and Van Roy [13], as we will discuss shortly. By carrying out the minimization over r, we can equivalently write (10) as

$$\begin{aligned} r_{t+1} = \frac{B_{t+1}^{-1}}{t+1} \sum_{k=0}^{t} \phi(x_k) \Big(&g(x_k, x_{k+1}) \\ &+ \alpha \min\{c(x_{k+1}), \phi(x_{k+1})' r_k\} \Big), \end{aligned}$$

where we denote

$$B_t = \frac{1}{t} \sum_{k=0}^{t-1} \phi(x_k) \phi(x_k)'.$$

To gain some insight into this iteration, let us rewrite it, using the definition of r_t and the relation

$$B_{t+1} = \frac{1}{t+1} \left(t B_t + \phi(x_t) \phi(x_t)' \right),$$

as follows:

$$\begin{aligned} r_{t+1} = r_t + \frac{B_{t+1}^{-1}}{t+1} \phi(x_t) \Big(&- \phi(x_t)' r_t + g(x_t, x_{t+1}) \\ &+ \alpha \min\{c(x_{t+1}), \phi(x_{t+1})' r_t\} \Big). \end{aligned} \tag{11}$$

The convergence of this iteration to r^* follows from a general convergence theorem of Choi and Van Roy [13]. However, we will show by example that its rate of convergence can be inferior to the least squares Q-learning algorithm [cf. (3)-(4)].

Accordingly, we consider another variant that aims to improve the practical (if not the theoretical) rate of convergence of iteration (10) [or equivalently (11)], and is new to our knowledge. In particular, we introduce a time window of size m, and we replace the terms $\min\{c(x_{k+1}), \phi(x_{k+1})' r_t\}$ in the least squares problem (2) with $\min\{c(x_{k+1}), \phi(x_{k+1})' r_{l_{k,t}}\}$, where

$$l_{k,t} = \min\{k + m - 1, t\}.$$

In other words, we consider the algorithm

$$\begin{aligned} r_{t+1} = \arg\min_{r \in \Re^s} \sum_{k=0}^{t} \Big(&\phi(x_k)' r - g(x_k, x_{k+1}) \\ &- \alpha \min\{c(x_{k+1}), \phi(x_{k+1})' r_{l_{k,t}}\} \Big)^2. \end{aligned} \tag{12}$$

Thus, at time t, the last m terms in the least squares sum are identical to the ones in the corresponding sum for the least squares Q-learning algorithm [cf. (2)]. The terms $\min\{c(x_{k+1}), \phi(x_{k+1})' r_{l_{k,t}}\}$ remain constant after m updates (when $l_{k,t}$ reaches the value $k + m - 1$), so they do not need to be updated further.

Note that in the first m iterations, this iteration is identical to the least squares Q-learning algorithm of Section III with unit stepsize. An important issue is the size of m. For large m, the algorithm approaches the least squares Q-learning algorithm, while for $m = 1$, it is identical to the earlier variant (10).

C. Comparison with Other Algorithms

Let us now consider an algorithm, due to Choi and Van Roy [13], and referred to as the *fixed point Kalman filter*. It applies to more general problems, but when specialized to the optimal stopping problem, it takes the form

$$r_{t+1} = r_t + \gamma_t B_{t+1}^{-1}\phi(x_t)\Big(-\phi(x_t)'r_t + g(x_t,x_{t+1}) + \alpha\min\big\{c(x_{t+1}),\phi(x_{t+1})'r_t\big\}\Big), \tag{13}$$

where γ_t is a diminishing stepsize. The algorithm is motivated by Kalman filtering ideas and the recursive least squares method in particular. It can also be viewed as a scaled version (with scaling matrix B_{t+1}^{-1}) of the method by Tsitsiklis and Van Roy [1], which has the form

$$r_{t+1} = r_t + \gamma_t\phi(x_t)\Big(-\phi(x_t)'r_t + g(x_t,x_{t+1}) + \alpha\min\{c(x_{t+1}),\phi(x_{t+1})'r_t\}\Big). \tag{14}$$

Scaling is believed to be instrumental for enhancing the rate of convergence.

It can be seen that when $\gamma_t = 1/(t+1)$, the iterations (11) and (13) coincide. However, the iterations (12) and (13) are different for a window size $m > 1$. As far as we know, the convergence proofs of [1] and [13] do not extend to iteration (12) or its modification that we will introduce in the next section (in part because of the dependence of r_{t+1} on as many as $t-m$ past iterates through the time window). The following example provides some insight into the behavior of the various algorithms discussed in this paper.

Example 1: This is a somewhat unusual example, which can be viewed as a simple DP model to estimate the mean of a random variable using a sequence of independent samples. It involves a Markov chain with a single state. At each time period, the cost produced at this state is a random variable taking one of n possible values with equal probability.[3] Let g_k be the cost generated at the kth transition. The "stopping cost" is taken to be very high so that the stopping option does not affect the algorithms. We assume that the costs g_k are independent and have zero mean and variance σ^2. The matrix Φ is taken to be the scalar 1, so r^* is equal to the true cost and $r^* = 0$.

Then, the least squares Q-learning algorithm of Section III with unit stepsize [cf. (3) and (4)] takes the form

$$r_{t+1} = \frac{g_0 + \cdots + g_t}{t+1} + \alpha r_t. \tag{15}$$

The first variant (Section IV-A) also takes this form, regardless of the method used for repartitioning, since the stopping cost is so high that it does not materially affect the calculations. Since iteration (15) coincides with the LSPE(0) method for this example, the corresponding rate of convergence results apply (see Yu and Bertsekas [9]). In particular, as $t \to \infty$, $\sqrt{t}\,r_t$ converges in distribution to a Gaussian distribution with mean zero and variance $\sigma^2/(1-\alpha)^2$, so that $E\{r_t^2\}$ converges to 0 at the rate $1/t$, i.e., there is a constant C such that

$$tE\{r_t^2\} \le C, \qquad \forall\ t = 0,1,\ldots.$$

The second variant [Section IV-B, with time window $m = 1$; cf. (11)], takes the form

$$r_{t+1} = \frac{g_t}{t+1} + \frac{t+\alpha}{t+1}r_t. \tag{16}$$

The fixed point Kalman filter algorithm [cf. (13)], and the Tsitsiklis and Van Roy algorithm [cf. (14)] are identical because the scaling matrix B_{t+1} is the scalar 1 in this example. They take the form

$$r_{t+1} = r_t + \gamma_t(g_t + \alpha r_t - r_t).$$

For a stepsize $\gamma_t = 1/(t+1)$, they become

$$r_{t+1} = \frac{g_t}{t+1} + \frac{t+\alpha}{t+1}r_t, \tag{17}$$

verifying that they are identical to the second variant (16).

We claim that iteration (17) converges more slowly than iteration (15), and that $tE\{r_t^2\} \to \infty$. To this end, we write

$$E\{r_{t+1}^2\} = \left(\frac{t+\alpha}{t+1}\right)^2 E\{r_t^2\} + \frac{\sigma^2}{(t+1)^2}.$$

Then

$$\zeta_{t+1} = \frac{(t+\alpha)^2}{t(t+1)}\zeta_t + \frac{\sigma^2}{t+1}.$$

From this equation (for $\alpha > 1/2$), we have

$$\zeta_{t+1} \ge \zeta_t + \frac{\sigma^2}{t+1},$$

so ζ_t tends to ∞.

Finally, the variant of Section IV-B with time window $m > 1$ [cf. (12)], for $t \ge m$ takes the form

$$r_{t+1} = \frac{g_0 + \cdots + g_t}{t+1} + \alpha\frac{r_{m-1} + r_m + \cdots + r_{t-1} + mr_t}{t+1}, \quad t \ge m. \tag{18}$$

For $t < m$, it takes the form

$$r_{t+1} = \frac{g_0 + \cdots + g_t}{t+1} + \alpha r_t, \quad t < m.$$

We may write iteration (18) as

$$r_{t+1} = \frac{g_t}{t+1} + \frac{t+\alpha}{t+1}r_t + \alpha\frac{(m-1)(r_t - r_{t-1})}{t+1}, \quad t \ge m,$$

and it can be shown again that $tE\{r_t^2\} \to \infty$, similar to iteration (17).

[3] A more conventional but equivalent example can be obtained by introducing states $1,\ldots,n$, one for each possible value of the cost per stage, and transition probabilities $p_{ij} = 1/n$ for all $i, j = 1,\ldots,n$.

D. Convergence Analysis

We can show the convergence of a slightly modified version of iteration (12), the variant of Section IV-B with time window $m > 1$. Our proof is different in style from the one of Choi and Van Roy [13], which applies to the case $m = 1$; it is based on a time scaling argument that is typical of the ODE approach. Generally, in the ODE approach, boundedness of the iterates must either be proved independently or assumed. In our case, we choose to modify the updates using a projection of $\phi(i)'r_t$ onto an appropriate lower bound, so that r_t is bounded with probability 1. It may be possible to remove this boundedness assumption through a more sophisticated analysis, but we have not been able to do so.

We first note that $\phi(i)'r_t$ is bounded from above for all states i. To ensure that $\phi(i)'r_t$ is also bounded from below, we introduce a scalar L satisfying

$$\phi(i)'r^* > L, \quad i = 1, \dots, n,$$

and we replace the terms $\min\big\{c(x_{k+1}), \phi(x_{k+1})'r_{l_{k,t}}\big\}$ in iteration (12) with

$$\min\big\{c(x_{k+1}), \max\{\phi(x_{k+1})'r_{l_{k,t}}, L\}\big\}.$$

In other words, defining functions f and h by

$$f(y) = \max\{y, L\}, \qquad h(x,r) = \min\big\{c(x), f(\phi(x)'r)\big\},$$

we consider the iteration

$$r_{t+1} = B_{t+1}^{-1} \frac{1}{t+1} \sum_{k=0}^{t} \phi(x_k)\Big(g(x_k, x_{k+1}) + \alpha h\big(x_{k+1}, r_{l_{k,t}}\big)\Big) \tag{19}$$

[cf. (12)]. Thus by construction r_t is bounded with probability 1. Since ΠF is a contraction mapping, the solution of the equation $f(\Phi r) = \Pi F f(\Phi r)$, where f is applied to each component, must still be Φr^*. We now state our convergence result.

Proposition 3: Let r_t be defined by (19). Then with probability 1, $r_t \to r^*$, as $t \to \infty$.

The proof of Prop. 3 can be found in [10]. Furthermore, the damped version of iteration (19) converges as well, i.e., the iteration

$$r_{t+1} = (1-\gamma)r_t + \gamma \hat{r}_{t+1}$$

converges to r^* with probability 1, where a constant stepsize $\gamma < 1$ is used to interpolate between r_t and the least squares solution (19), now denoted by $\hat{r}_{t+1}$. We refer readers to [10] for a detailed analysis.

V. CONCLUSIONS

In this paper, we have proposed new Q-learning algorithms for the approximate cost evaluation of optimal stopping problems, using least squares ideas that are central in the LSPE method for policy cost evaluation with linear function approximation. We have aimed to provide alternative, faster algorithms than those of Tsitsiklis and Van Roy [1], and Choi and Van Roy [13]. The distinctive feature of optimal stopping problems is the underlying mapping F, which is a contraction with respect to the projection norm $\|\cdot\|_\pi$ (cf. Lemma 1). Our convergence proofs made strong use of this property.

It is possible to consider the extension of our algorithms to general finite-spaces discounted problems. An essential requirement for the validity of such extended algorithms is that the associated mapping is a contraction with respect to some Euclidean norm. Under this quite restrictive assumption, it is possible to show certain convergence results. In particular, Choi and Van Roy [13] have shown the convergence of an algorithm that generalizes the second variant of Section IV for the case $m = 1$. It is also possible to extend this variant for the case where $m > 1$ and prove a corresponding convergence result by using our line of proof.

VI. ACKNOWLEDGMENTS

We thank Prof. Ben Van Roy for helpful comments.

REFERENCES

[1] J. N. Tsitsiklis and B. Van Roy, “Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing financial derivatives,” *IEEE Trans. Automat. Contr.*, vol. 44, pp. 1840–1851, 1999.

[2] J. Barraquand and D. Martineau, “Numerical valuation of high dimensional multivariate American securities,” *Journal of Financial and Quantitative Analysis*, vol. 30, pp. 383–405, 1995.

[3] F. A. Longstaff and E. S. Schwartz, “Valuing American options by simulation: A simple least-squares approach,” *Review of Financial Studies*, vol. 14, pp. 113–147, 2001.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Belmont, MA: Athena Scientific, 2007.

[6] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Machine Learning*, vol. 3, pp. 9–44, 1988.

[7] D. P. Bertsekas and S. Ioffe, “Temporal differences-based policy iteration and applications in neuro-dynamic programming,” MIT, LIDS Tech. Report LIDS-P-2349, 1996.

[8] A. Nedić and D. P. Bertsekas, “Least squares policy evaluation algorithms with linear function approximation,” *Discrete Event Dyn. Syst.*, vol. 13, pp. 79–110, 2003.

[9] H. Yu and D. P. Bertsekas, “Convergence results for some temporal difference methods based on least squares,” MIT, LIDS Tech. Report 2697, 2006.

[10] ——, “A least squares Q-learning algorithm for optimal stopping problems,” MIT, LIDS Tech. Report 2731, 2006.

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

[12] J. A. Boyan, “Least-squares temporal difference learning,” in *Proc. The 16th Int. Conf. Machine Learning*, 1999.

[13] D. S. Choi and B. Van Roy, “A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning,” *Discrete Event Dyn. Syst.*, vol. 16, no. 2, pp. 207–239, 2006.

[14] D. P. Bertsekas, V. S. Borkar, and A. Nedić, “Improved temporal difference methods with linear function approximation,” MIT, LIDS Tech. Report 2573, 2003, also appears in “Learning and Approximate Dynamic Programming,” by A. Barto, W. Powell, J. Si, (Eds.), IEEE Press, 2004.

[15] C. J. C. H. Watkins, “Learning from delayed rewards,” Doctoral dissertation, University of Cambridge, Cambridge, United Kingdom, 1989.

[16] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, pp. 279–292, 1992.

[17] J. N. Tsitsiklis, “Asynchronous stochastic approximation and Q-learning,” *Machine Learning*, vol. 16, pp. 185–202, 1994.

[18] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation,” *IEEE Trans. Automat. Contr.*, vol. 42, no. 5, pp. 674–690, 1997.