2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Zap Q-Learning for Optimal Stopping

Shuhang Chen* Adithya M. Devraj† Ana Bušić‡ Sean Meyn§

***Abstract*— This paper concerns approximate solutions to the optimal stopping problem for a geometrically ergodic Markov chain on a continuous state space. The starting point is the Galerkin relaxation of the dynamic programming equations that was introduced by Tsitsikilis and Van Roy in the 1990s, which motivated their Q-learning algorithm for optimal stopping. It is known that the convergence rate of Q-learning is in many cases very slow. The reason for slow convergence is explained here, along with a variant of "Zap-Q-learning" algorithm, designed to achieve the optimal rate of convergence. The main contribution is to establish consistency of Zap-Q-learning algorithm for a linear function approximation setting. The theoretical results are illustrated using an example from finance.**

I. Introduction

Consider a discrete-time Markov chain $\boldsymbol{X} = \{X_n : n \geq 0\}$ evolving on a general state-space X. The goal in optimal stopping time problems is to minimize over all stopping times τ, the associated expected cost:

$$\mathsf{E}\left[\sum_{n=0}^{\tau-1} \beta^n c(X_n) + \beta^\tau c_s(X_\tau)\right] \tag{1}$$

where $c : \mathsf{X} \to \mathbb{R}$ denotes the per-stage cost, $c_s : \mathsf{X} \to \mathbb{R}$ the terminal cost, and $\beta \in (0,1)$ is the discount factor. Applications arise in hypothesis testing (quickest change detection), and in finance (see Section V).

This paper concerns techniques from reinforcement learning to approximate the value function associated with the optimal stopping rule, and from this an approximation of the optimal policy.

Definitions & Problem Setup: The time-homogeneous Markov chain $\boldsymbol{X}$ is defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. It evolves on a compact state-space $\mathsf{X} \subset \mathbb{R}^m$, with Borel σ-algebra denoted $\mathcal{B}$, and its dynamics is determined by an initial distribution $\mu : \mathsf{X} \to [0,1]$, and a transition kernel P:

$$P(x, A) = \Pr(X_{n+1} \in A \mid X_n = x) \quad \text{for } x \in \mathsf{X},\ A \in \mathcal{B}$$

It is assumed that $\boldsymbol{X}$ is *uniformly ergodic*: There exists a unique invariant probability measure π, a constant $B_g < \infty$, and $0 < \rho < 1$, such that, for all $x \in \mathsf{X}$ and $A \in \mathcal{B}$,

$$\|P^n(x, A) - \pi(A)\| \leq B_g \rho^n\,, \qquad n \geq 0 \tag{2}$$

*S.C. is with the Department of Mathematics at the University of Florida
†A.D. is with the Department of ECE at the University of Florida
‡A.B. is with Inria and DI ENS, École Normale Supérieure, CNRS, PSL Research University, Paris, France
§S.M. is with Department of Electrical and Computer Engineering, University of Florida, and Inria International Chair, Paris

Acknowledgements: Financial support from ARO grant W911NF1810334 is gratefully acknowledged. Additional support from EPCN 1609131 & CPS 1646229, and French National Research Agency grant ANR-16-CE05-0008.

Denote by $\{\mathcal{F}_n : n \geq 0\}$ the filtration associated with $\boldsymbol{X}$. The Markov property asserts that for bounded measurable functions $h \colon \mathsf{X} \to \mathbb{R}$,

$$\mathsf{E}[h(X_{n+1}) \mid \mathcal{F}_n\,, X_n = x] = \int P(x, dy) h(y)$$

In this paper, a stopping time $\tau : \Omega \to [0, \infty)$ is a random variable taking on values in the non-negative integers, with the defining property $\{\omega : \tau(\omega) \leq n\,,\ \omega \in \Omega\} \in \mathcal{F}_n$ for each $n \geq 0$. A stationary policy is defined to be a measurable function $\phi \colon \mathsf{X} \to \{0,1\}$ that defines a stopping time:

$$\tau^\phi = \min\{n \geq 0 \colon \phi(X_n) = 1\} \tag{3}$$

The optimal value function is defined as the infimum of (1) over all stopping times: for any $x \in \mathsf{X}$,

$$h^*(x) := \inf_\tau \mathsf{E}\big[\sum_{n=0}^{\tau-1} \beta^n c(X_n) + \beta^\tau c_s(X_\tau) | X_0 = x\big] \tag{4}$$

The associated *Q-function* is defined as

$$Q^*(x) := c(x) + \beta \mathsf{E}[h^*(X_1) \mid X_0 = x]\,,$$

which solves the associated Bellman equation [13]:

$$Q^*(x) = c(x) + \beta \mathsf{E}[\min(c_s(X_1), Q^*(X_1)) | X_0 = x] \tag{5}$$

An optimal stopping rule is the stationary policy,

$$\phi^*(x) = \mathbb{I}\{c_s(x) \leq Q^*(x)\} \tag{6}$$

and thence $\tau^* = \tau^{\phi^*}$.

The Bellman equation (5) can be expressed as the functional fixed point equation: $Q^* = FQ^*$, where F denotes the *dynamic programming operator*:

$$FQ(x) := c(x) + \beta \mathsf{E}[\min(c_s(X_1), Q(X_1)) | X_0 = x] \tag{7}$$

Analysis is framed in the usual Hilbert space $L_2(\pi)$ of real-valued measurable functions on X with

$$\langle f, g \rangle_\pi = \mathsf{E}[f(X) g(X)]\,, \quad X \sim \pi \tag{8a}$$

$$\|f\|_\pi = \sqrt{\langle f, f \rangle_\pi}\,, \tag{8b}$$

It is assumed throughout that $c,\ c_s \in L_2(\pi)$.

Objective: The goal here as in [12], [13] is to approximate Q^* within a parameterized family $\{Q^\theta : \theta \in \mathbb{R}^d\}$. We restrict to linear parameterizations throughout:

$$Q^\theta(x) := \theta^{\scriptscriptstyle T} \psi(x), \qquad x \in \mathsf{X} \tag{9}$$

where $\psi_i \in L_2(\pi)$, $1 \leq i \leq d$, are the basis functions, and $\psi := [\psi_1, \ldots, \psi_d]^{\scriptscriptstyle T}$. It is assumed that the basis functions are

linearly independent in $L_2(\pi)$, which means that the $d \times d$ matrix Σ_ψ is full rank, with

$$\Sigma_\psi(i,j) = \langle \psi_i, \psi_j \rangle_\pi, \quad 1 \le i,j \le d \tag{10}$$

For each $\theta \in \mathbb{R}^d$, the *Bellman error* is the function

$$\mathcal{B}_\mathcal{E}^\theta = FQ^\theta - Q^\theta.$$

If $Q^{\theta^*} = Q^*$, then $\mathcal{B}_\mathcal{E}^{\theta^*} \equiv 0$ by (5). This isn't realistic, so we consider instead a *Galerkin relaxation* of (5): As in the prior work [13], [5], [16], the goal is to solve

$$\langle FQ^{\theta^*} - Q^{\theta^*}, \psi_i \rangle_\pi = 0\,, \quad 1 \le i \le d. \tag{11}$$

Literature Survey: The relaxation was first considered in [13]. The authors propose an extension of the TD(0) algorithm of [10], [12], and obtain convergence results under a finite state-action space setting.

The algorithm of [13] is in fact closely connected to Watkins' Q-learning algorithm [14], [15]. It differs from the standard formulation of TD(0) algorithm because the recursion is non-linear even when using a linear function approximation (this is due to the minimum in the definition of F in (7)). The nonlinear nature of Q-learning is in general a theoretical challenge, but not in the special case of optimal stopping. This is because the operator F is a contraction $L_2(\pi)$ [13]:

$$\|FQ - FQ'\|_\pi \le \beta \|Q - Q'\|_\pi, \quad \text{for all } Q, Q' \in L_2(\pi)$$

Matrix gain variants of the algorithm presented in [13] are proposed in [5], [16]. The algorithm of [16] is similar to the LSPE (0) algorithm of [9]. Variants are also proposed in order to reduce computational complexity. All of these papers are posed within a function-approximation setting, and the state-space is assumed finite.

The present paper is devoted to a version of the Zap Q-learning algorithm introduced in [7], [8] (see [6] for a survey). In this prior work, convergence is established only in the tabular setting (wherein the ψ_i's span all possible functions). The contributions are summarized as follows:

(i) Convergence of Zap-Q-learning is established for optimal-stopping in a linear function approximation setting, and without a finite state space.

(ii) The ODE analysis (cf. Theorem 3.6) provides significant insight into algorithm dynamics.

(iii) The algorithm has *optimal asymptotic variance*, implying better convergence rates (see Section III for discussion and Section V for examples).

The extension of [8] to the current setting is *not trivial*: The tabular case is far simpler to analyze, with lots of special structure. In contrast, theory for convergence of any Q-learning algorithm in a function approximation setting is scant.

The remainder of the paper is organized as follows: Section II contains notation and the Zap-Q-learning algorithm. Assumptions and main results are contained in Section III. Section IV provides a high-level proof of the results, numerical results are collected together in Section V, and conclusions in Section VI. Full proofs are available in the extended version of this paper, available on arXiv [3].

II. Q-learning for Optimal Stopping

The objective (11) can be expressed:

$$A(\theta^*)\theta^* + \beta \bar{c}_s(\theta^*) + b^* = 0\,, \tag{12}$$

where, for each $\theta \in \mathbb{R}^d$, $A(\theta)$ is a $d \times d$ matrix, and b^* and $\bar{c}_s(\theta)$ are d-dimensional vectors:

$$A(\theta) := \mathsf{E}[\psi(X_n)\beta\mathcal{S}_\theta \psi^{\scriptscriptstyle T}(X_{n+1}) - \psi(X_n)\psi^{\scriptscriptstyle T}(X_n)] \tag{13}$$

$$b^* := \mathsf{E}[\psi(X_n)c(X_n)] \tag{14}$$

$$\bar{c}_s(\theta) := \mathsf{E}[\psi(X_n)\mathcal{S}_\theta^c c_s(X_{n+1})] \tag{15}$$

where for any function f,

$$\mathcal{S}_\theta f(x) := \mathbb{I}\{Q^\theta(x) < c_s(x)\} f(x) \tag{16a}$$

$$\mathcal{S}_\theta^c f(x) := \mathbb{I}\{c_s(x) \le Q^\theta(x)\} f(x) \tag{16b}$$

For each $\theta \in \mathbb{R}^d$, we associate a policy:

$$\phi^\theta(x) := \mathbb{I}\{c_s(x) \le Q^\theta(x)\}\,, \quad x \in \mathsf{X} \tag{17}$$

This results in $\mathcal{S}_\theta f = (1 - \phi^\theta) f$.

Zap Q-Learning: Given a $d \times d$ matrix gain sequence $\{G_n : n \ge 0\}$, and a scalar step-size sequence $\{\alpha_n : n \ge 0\}$, the corresponding *matrix gain Q-learning algorithm* is defined by the recursion,

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_{n+1} \psi(X_n) d_{n+1} \tag{18}$$

with $\{d_n\}$ the "temporal difference" sequence:

$$d_{n+1} := c(X_n) + \beta \min(c_s(X_{n+1}), Q^{\theta_n}(X_{n+1})) - Q^{\theta_n}(X_n)$$

The algorithm proposed in [13] is (18), with $G_n \equiv I$. This is similar to the TD(0) algorithm [12], [10].

The *fixed point Kalman filter* algorithm of [5] can also be written as a special case of (18): We have $G_n \equiv [\widehat{\Sigma}_n^\psi]^\dagger$ (the pseudo-inverse), and

$$\widehat{\Sigma}_{n+1}^\psi = \widehat{\Sigma}_n^\psi + \alpha_{n+1}\big[\psi(X_n)\psi^{\scriptscriptstyle T}(X_n) - \widehat{\Sigma}_n^\psi\big] \tag{19}$$

(a Monte-Carlo estimate of Σ_ψ defined in (10).)

The Zap-Q algorithm uses $G_n = -\widehat{A}_{n+1}^\dagger$, with $\widehat{A}_{n+1}$ an estimate of $A(\theta_n)$ (see (13)). The term inside the expectation in (13), following the substitution $\theta = \theta_n$, is denoted

$$A_{n+1} := \psi(X_n)\big[\beta\mathcal{S}_{\theta_n}\psi(X_{n+1}) - \psi(X_n)\big]^{\scriptscriptstyle T} \tag{20}$$

Algorithm 1 *Zap-Q for Optimal Stopping*

Input: Initial $\theta_0 \in \mathbb{R}^d$, $\widehat{A}_0$: $d \times d$, negative definite; step-size sequences $\{\alpha_n\}$ and $\{\gamma_n\}$ and $n = 0$

1: **repeat**

2: Obtain the *Temporal Difference* term:

$$d_{n+1} = c(X_n) + \beta \min(c_s(X_{n+1}), Q^{\theta_n}(X_{n+1})) - Q^{\theta_n}(X_n)$$

3: Update the matrix gain estimate $\widehat{A}_n$ of $A(\theta_n)$, with A_{n+1} defined in (20):

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}\big[A_{n+1} - \widehat{A}_n\big] \tag{21}$$

4: Update the parameter vector:

$$\theta_{n+1} = \theta_n - \alpha_{n+1}\widehat{A}^{\dagger}_{n+1}\psi(X_n)d_{n+1} \tag{22}$$

5: $n = n + 1$

6: **until** $n \geq N$

Output: $\theta = \theta_N$

Algorithm 1 belongs to a general class of algorithms known as *two-time-scale stochastic approximation* [1]: the recursion (21) on the *faster* time-scale is designed to estimate $A(\theta_n)$. The step-size sequences $\{\alpha_n\}$ and $\{\gamma_n\}$ have to satisfy the standard requirements for separation of time-scales [1]: for any $\varrho \in (0.5, 1)$, we choose

$$\alpha_n = 1/n\,, \qquad \gamma_n = 1/n^{\varrho} \tag{23}$$

III. Assumptions and Main Results

We first summarize preliminary results here that will be used to establish the main results in the following sections. The proofs of all the technical results are contained in the Appendix of [3].

We start with the contraction property of the operator F defined in (7). The following is a result directly obtained from [13] (see [13, Lemma 4 on p. 1844]).

Lemma 3.1: The dynamic programming operator F defined in (7) satisfies:

$$\|FQ - FQ'\| \leq \beta\|Q - Q'\|, \qquad Q, Q' \in L_2(\pi).$$

Furthermore, Q^* is the unique fixed point of F in $L_2(\pi)$. □

Recall that $Q^\theta : \mathsf{X} \to \mathbb{R}$ is defined in (9). Similar to the operator F, for each $\theta \in \mathbb{R}^d$ we define operators H^θ and F^θ that operate on functions $Q : \mathsf{X} \to \mathbb{R}$ as follows:

$$H^\theta Q(x) = \begin{cases} Q(x), & \text{if } \ Q^\theta(x) < c_s(x) \\ c_s(x), & \text{otherwise} \end{cases} \tag{24}$$

$$F^\theta Q = c + \beta P H^\theta Q. \tag{25}$$

The following Lemma is a slight extension of Lemma 3.1.

Lemma 3.2: For each $\theta \in \mathbb{R}^d$, the operator F^θ satisfies:

$$\|F^\theta Q - F^\theta Q'\| \leq \beta\|Q - Q'\|, \qquad Q, Q' \in L_2(\pi)$$

□

The next result is a direct consequence of Lemma 3.2, and establishes the inveritbility of the matrix $A(\theta)$ for any $\theta \in \mathbb{R}^d$:

Lemma 3.3: For each $\theta \in \mathbb{R}^d$,

(i) The $d \times d$ matrix $A(\theta)$ defined in (13) satisfies:

$$-v^{\intercal}A(\theta)v \geq (1-\beta)v^{\intercal}\Sigma_\psi v, \tag{26}$$

for each $v \in \mathbb{R}^d$, with Σ_ψ defined in (10).

(ii) Eigenvalues of $A(\theta)$ are negative and strictly bounded away from 0, and $\{A^{-1}(\theta) : \theta \in \mathbb{R}^d\}$ is uniformly bounded.

□

Consider, for each $\theta \in \mathbb{R}^d$,

$$b(\theta) = -A(\theta)\theta - \beta\overline{c}_s(\theta) \tag{27a}$$

$$c^\theta(x) = Q^\theta(x) - \mathsf{E}\big[\beta \min(c_s(X_n), Q^\theta(X_n)) \,|\, X_{n-1} = x\big] \tag{27b}$$

The vector $b(\theta)$ is analogous to b^* in (12), and (27b) recalls the Bellman equation (5). Prop. 3.4 shows that $b(\theta)$ is the "projection" of the cost function c^θ, similar to how b^* is related to c through (14).

Proposition 3.4: For each $\theta \in \mathbb{R}^d$,

$$b(\theta) = \mathsf{E}\big[c^\theta(X)\psi(X)\big]\,, \qquad X \sim \pi \tag{28}$$

In particular, $b^* = b(\theta^*)$.

□

The proof of Prop. 3.4 follows from the definitions. It implies a Lipschitz bound on the function b:

Lemma 3.5: For some $B_L > 0$, and each $\theta^1, \theta^2 \in \mathbb{R}^d$,

$$\|b(\theta^1) - b(\theta^2)\| \leq B_L\|\theta^1 - \theta^2\|$$

□

The following assumptions are imposed throughout:

A1: $\boldsymbol{X}$ is a uniformly ergodic Markov chain on the compact state space X.

A2: The solution θ^* to (11) is unique.

A3.1: The conditional distribution of $\psi(X_{n+1})$ given $X_n = x$ has a density, $f_{\psi|x}(z)$; its likelihood ratio with respect to the Gaussian density $\mathcal{N}(\psi(x), I)$ is uniformly bounded.

A3.2: The function $c_s - 1$ is in the span of $\{\psi_i\}$.

A4: The sequence $\{\theta_n : n \geq 1\}$ is bounded *a.s.*.

The density assumption in **A3** is imposed to ensure that the conditional expectation given X_n of functions such as $\mathcal{S}_\theta\psi^{\intercal}(X_{n+1})$ are smooth as a function of θ. Assumption **A3** also implies that $\Sigma_\psi > 0$.

We conjecture that the boundedness assumption **A4** actually follows from the ODE analysis in this paper, since the "ODE at infinity" posed in [2], [1] is globally asymptotically stable under **A1**–**A3**. The remaining challenge is to extend the "Borkar-Meyn Theorem" of [2], [1] to two time-scale stochastic approximation with Markovian noise.

The main result of this paper establishes convergence:

Theorem 3.6: Subject to Assumptions **A1**–**A4**, the following hold for the Zap-Q algorithm:

(i) The parameter sequence $\{\theta_n\}$ converges to θ^* a.s., where θ^* satisfies (11).

(ii) The sequences $\{\theta_n, b(\theta_n)\}$ admit an ODE approximation by a pair of continuous functions $\{w_t, b_t\}$ solving

$$\begin{aligned} \tfrac{d}{dt} b_t &= -b_t + b \\ b_t &= -A(w_t) w_t - \beta \bar{c}_s(w_t) \end{aligned} \tag{29}$$

□

The term *ODE approximation* is standard in the SA literature: For $t \geq s$, let w_t^s denote the solution to:

$$\tfrac{d}{dt} w_t^s = \xi(w_t^s), \quad w_s^s = \overline{w}_s \tag{30}$$

for some $\xi \colon \mathbb{R}^d \to \mathbb{R}^d$, and $\overline{w}_t$ denote the continuous time process constructed from the sequence $\{\theta_n : n \geq 0\}$ via linear interpolation. We say that the ODE approximation $\frac{d}{dt} w_t = \xi(w_t)$ holds for the sequence $\{\theta_n : n \geq 1\}$, if the following is true for any $T > 0$:

$$\lim_{s \to \infty} \sup_{t \in [s, s+T]} \|\overline{w}_t - w_t^s\| = 0, \ a.s.$$

See [1] and [3] for more details.

We turn next to rates of convergence. The *asymptotic covariance* is defined to be the limit:

$$\Sigma_\Theta := \lim_{n \to \infty} n \mathsf{E}\big[(\theta_n - \theta^*)(\theta_n - \theta^*)^{\mathrm{T}}\big] \tag{31}$$

A characterization of Σ_Θ is obtained next based on a "noise covariance matrix" $\Sigma_{\mathcal{E}}$. A "noise sequence" $\{\mathcal{E}_n\}$ is introduced, and then

$$\Sigma_{\mathcal{E}} = \lim_{T \to \infty} \frac{1}{T} \mathsf{E}[S_T S_T^{\mathrm{T}}] \tag{32}$$

in which $S_T = \sum_{n=1}^T \mathcal{E}_n$. The sequence $\{\mathcal{E}_n\}$ is defined by the sum of errors:

$$\mathcal{E}_n := \tilde{A}_{n+1} \theta^* + \tilde{b}_{n+1} + \tilde{A}_{n+1} \tilde{\theta}_n \tag{33}$$

where $\tilde{\theta}_n := \theta_n - \theta^*$, $\tilde{A}_{n+1} := A_{n+1} - A(\theta^*)$ with A_{n+1} defined in (20), $\tilde{b}_{n+1} := b_{n+1} - b^*$ with

$$b_{n+1} := \psi(X_n)\big[c(X_n) + \mathcal{S}_{\theta_n}^c c_s(X_{n+1})\big] \tag{34}$$

and b^* defined in (14).

Consider the matrix gain Q-learning algorithm (18) in which $G_n \to G$ as $n \to \infty$ for some matrix G. Suppose that all eigenvalues of $A_G := GA(\theta^*)$ satisfy $\mathrm{Real}(\lambda) < -\frac{1}{2}$. Under further mild assumptions and standard analysis (see Section 2.2 of [7] and references therein), the asymptotic covariance solves the Lyapunov equation:

$$\big(A_G + \tfrac{1}{2} I\big)\Sigma_\Theta + \Sigma_\Theta \big(A_G + \tfrac{1}{2} I\big)^{\mathrm{T}} + G \Sigma_{\mathcal{E}} G^{\mathrm{T}} = 0 \tag{35}$$

If there exists an eigenvalue of A_G satisfying $\mathrm{Real}(\lambda) > -\frac{1}{2}$ then, subject to a non-degeneracy condition on $\Sigma_{\mathcal{E}}$, the mean-square error $\mathsf{E}[\|\tilde{\theta}_n\|^2]$ converges to zero *slower* than $O(1/n)$ [4].

Optimality of the asymptotic covariance: It is possible to optimize the covariance Σ_Θ over all matrix gains G using (35). The choice $G^* = -A(\theta^*)$ results in the *minimum* value:

$$\Sigma^* = A(\theta^*)^{-1} \Sigma_{\mathcal{E}} \big(A(\theta^*)^{-1}\big)^{\mathrm{T}} \tag{36}$$

That is, $\Sigma_\Theta^G - \Sigma^*$ is positive semi-definite, with Σ_Θ^G the solution to the Lyapunov equation (35).

The Zap Q algorithm is specifically designed to achieve the optimal asymptotic covariance: Thm. 3.6 tells us that we have the required convergence $G_n \to G^*$ for this algorithm. A few gaps in the theory remain: we haven't yet justified linearization of the dynamics to obtain the functional Central Limit Theorem for the scaled error $\sqrt{n}\tilde{\theta}_n$. For this we require additional tightness assumptions on this sequence [1]. Minor additional bounds are also needed to ensure convergence of (31).

IV. Overview of the Proof of Theorem 3.6

Unlike martingale difference assumptions in standard stochastic approximation, the noise in our algorithm is Markovian. The first part of this section establishes that our noise sequence satisfies the so called *ODE friendly property* [11]: A vector-valued sequence of random variables $\{\mathcal{E}_k\}$ will be called *ODE-friendly* if it admits the decomposition,

$$\mathcal{E}_k = \Delta_k + \mathcal{T}_k - \mathcal{T}_{k-1} + \varepsilon_k \,, \quad k \geq 1 \tag{37}$$

in which:

(i) $\{\Delta_k : k \geq 1\}$ is a martingale-difference sequence satisfying $\mathsf{E}[\|\Delta_k\|^2 \,|\, \mathcal{F}_k] \leq \bar{\sigma}_\Delta^2 < \infty$ *a.s.* for all k

(ii) $\{\mathcal{T}_k : k \geq 1\}$ is a bounded sequence

(iii) The final sequence $\{\varepsilon_k\}$ is bounded and satisfies:

$$\sum_{k=1}^{\infty} \gamma_k \|\varepsilon_k\| < \infty \quad a.s.. \tag{38}$$

Subject to these assumptions, the coupled recursion (21,22) that define $\{\theta_n, \widehat{A}_n\}$ can be regarded as a noisy Euler approximation of an ODE. This is made clear in the following:

Lemma 4.1: The coupled recursion (21,22) that define $\{\theta_n, \widehat{A}_n\}$ can be expressed,

$$\begin{aligned} \theta_{n+1} &= \theta_n - \alpha_{n+1} \widehat{A}_{n+1}^{\dagger} \big[A(\theta_n)\theta_n + \beta \bar{c}_s(\theta_n) + b^* \\ &\qquad + \mathcal{E}_{n+1}^A \theta_n + \mathcal{E}_{n+1}^\theta\big] \end{aligned} \tag{39a}$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}\big[A(\theta_n) - \widehat{A}_n + \mathcal{E}_{n+1}^A\big] \tag{39b}$$

in which $\{\mathcal{E}_n^\theta, \mathcal{E}_n^A : n \geq 1\}$ are ODE-friendly. □

The higher gain in (21) results in the approximation $\widehat{A}_{n+1} \approx A(\theta_n)$ because this is true for the associated ODE. This leads to a single ODE to approximate $\{\theta_n\}$:

$$\tfrac{d}{dt} w_t = -A^{-1}(w_t)\big[b^* - b(w_t)\big] \tag{40}$$

The ODE approximation is in the sense of (30).

Justification of the approximating ODE involves first constructing continuous time process derived from (21,22). Denote

$$t_n = \sum_{i=1}^n \alpha_i, \quad n \geq 1, \quad t_0 = 0 \,, \tag{41}$$

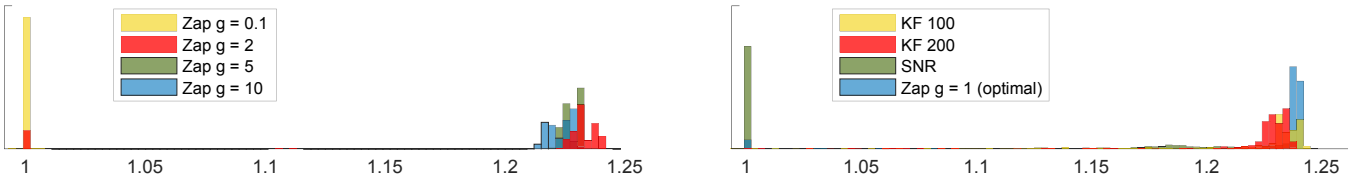

Fig. 1: Average rewards obtained using various Q-learning algorithms

and define $\overline{w}_{t_n} = \theta_n$, $\bar{\mathcal{A}}_{t_n} = \widehat{A}_n$ at these time points, with the definitions extended to $\mathbb{R}_+$ via linear interpolation. Furthermore, for each $t \geq 0$, denote

$$\bar{b}_t \equiv b(\overline{w}_t) := -A(\overline{w}_t)\overline{w}_t - \beta \bar{c}_s(\overline{w}_t)$$

The following result establishes that $\widehat{A}_n$ recursively obtained by (21) approximates the mean $A(\theta_n)$:

Lemma 4.2: Suppose the sequence $\{\mathcal{E}_n^A : n \geq 1\}$ is ODE-friendly. Then,

(i) $\lim_{n\to\infty} \|\widehat{A}_n - A(\theta_n)\| = 0$, *a.s.*

(ii) Consequently, $\widehat{A}_n^\dagger \neq \widehat{A}_n^{-1}$ only finitely often, and $\lim_{n\to\infty} \|\widehat{A}_n^\dagger - A^{-1}(\theta_n)\| = 0$, *a.s.*

□

Lemma 4.3: The ODE approximation for $\{\theta_n\}$ holds: with probability one, the piece-wise continuous function $\overline{w}_t$ asymptotically tracks the ODE (40). □

For a fixed (but arbitrary) time horizon $T > 0$, we define two families of uniformly bounded and uniformly Lipschitz continuous functions: $\{\overline{w}_{s+t}, t \in [0,T]\}_{s\geq 0}$ and $\{\bar{b}_{s+t}, t \in [0,T]\}_{s\geq 0}$. Sub-sequential limits of $\{\overline{w}_{s+t}, t \in [0,T]\}_{s\geq 0}$ and $\{\bar{b}_{s+t}, t \in [0,T]\}_{s\geq 0}$ are denoted w_t and b_t respectively. The ODE limit of the projected cost has a simple representation:

Lemma 4.4: For any sub-sequential limits $\{w_t, b_t\}$, the following hold for *a.e.* $t \in [0,T]$: $b_t = b(w_t)$ and

$$\frac{d}{dt} b_t = -A(w_t)\frac{d}{dt} w_t = -b_t + b^* \tag{42}$$

□

Proof of Thm. 3.6: Bounedness of sequences $\{\widehat{A}_n \ \widehat{A}_n^{-1} : n \geq 0\}$ is established in Lemma 4.2. Together with boundedness assumption of $\{\theta_n : n \geq 0\}$, the ODE approximation is established in Lemma 4.4. Result (i) then follows from those two results using standard arguments from [1]. ■

V. Numerical Results

The performance of the Zap Q-learning algorithm in comparison with existing techniques is illustrated using the finance example of [5], [13]. The Zap algorithm performs very well, despite the fact that density assumption imposed in **A3** does not hold, and X is not compact.

Finance model: The following example was used in [5], [13] to evaluate the performance of their algorithms for optimal stopping. The reader is referred to these references for complete details of the set-up.

The Markovian state process considered is the vector of ratios: $X_n = (\widetilde{p}_{n-99}, \widetilde{p}_{n-98}, \ldots, \widetilde{p}_n)^\intercal / \widetilde{p}_{n-100}$, $n \geq 0$, in which $\{\widetilde{p}_t : t \in \mathbb{R}\}$ is a geometric Brownian motion (models history of an exogenous price-process). This Markov chain is uniformly ergodic on $\mathsf{X} = \mathbb{R}^{100}$.

The "time to exercise" is modeled as a stopping time $\tau \in \boldsymbol{Z}^+$. The associated expected reward is defined as $\mathsf{E}[\beta^\tau r(X_\tau)]$, with $r(X_n) := X_n(100) = \widetilde{p}_n / \widetilde{p}_{n-100}$. The objective of finding a policy that maximizes the expected reward is as an optimal stopping time problem.

The value function is defined to be the infimum (4), with $c \equiv 0$ and $c_s \equiv -r$. The associated Q-function is defined using (5), and the associated optimal policy using (6): $\phi^*(x) = \mathbb{I}\{r(x) \geq Q^*(x)\}$.

When the Q-function is linearly approximated using (9), the associated value function can be expressed:

$$h_{\phi^\theta}(x) := \mathsf{E}[\beta^{\tau_\theta} r(X_{\tau_\theta}) \mid x_0 = x], \tag{43}$$

where,

$$\begin{aligned} \tau_\theta &:= \min\{n \colon \phi^\theta(X_n) = 1\} \\ \phi^\theta(x) &:= \mathbb{I}\{r(x) \geq Q^\theta(x)\} \end{aligned} \tag{44}$$

Given a parameter estimate θ and initial state $X(0) = x$, the corresponding average reward $h_{\phi^\theta}(x)$ was estimated using Monte-Carlo.

Implementation Details and Experimental Results: Along with Zap Q-learning algorithm we also implement the *fixed point Kalman filter* algorithm of [5]. This algorithm is given by the update equations (18) and (19). The objective here is to compare the performances of the fixed point Kalman filter algorithm with the Zap-Q learning algorithm in terms of the resulting average reward (43).

The computational/storage complexity of the least squares Q-learning algorithm (and its variants) made their implementation impossible in this study.

The experimental setting of [5], [13] was used to define the set of basis functions and other parameters, with the $d = 10$ basis functions defined in [5].

Recall that the step-size for the Zap Q-learning algorithm is given in (23). We set $\gamma_n = n^{-0.85}$ for all implementations of the Zap algorithm, but similar to what is done in [5], we experimented with different choices for α_n. Specifically, in addition to $\alpha_n = n^{-1}$, we considered

$$\alpha_n = \frac{g}{q+n} \tag{45}$$

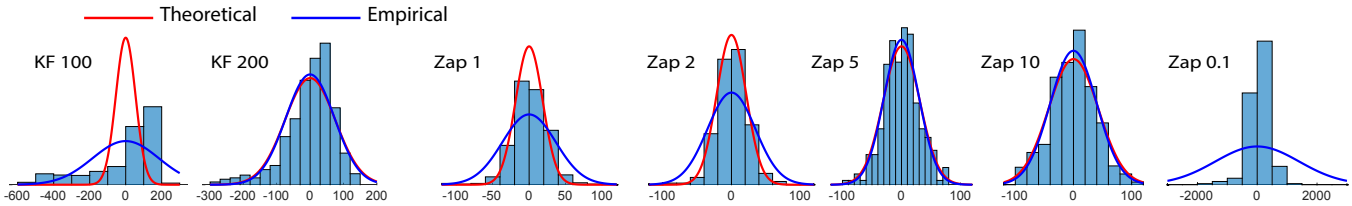

Fig. 2: Asymptotic variance for various Q-learning algorithms for optimal stopping in finance

with $a = 10^4$ and experiment with $g = 0.1, 1, 2, 5$, and 10 (the value $g = 0.1$ is the only case in which a Lyapunov equation analysis predicts infinite asymptotic covariance)

In the implementation of the fixed point Kalman filter algorithm, as suggested by the authors, we chose step-size $\alpha_n = 1/n$ for the matrix gain update rule in (19), and step-size of the form (45) for the parameter update in (18). Specifically, we let $b = 10^4$, and $g = 100$ and 200. The number of iterations for each of the algorithm is $N = 2 \cdot 10^6$.

Average reward histograms were obtained through the following steps: 500 parallel simulations were performed for each algorithm to obtain as many estimates of θ^*. Each of these estimates defines a policy ϕ^{θ_N} as defined in (44). We then estimated the corresponding average reward $h_{\phi^{\theta_N}}(x)$ defined in (43), with $X(0) = x = 1$.

Along with the average discounted rewards, histograms of parameter estimation error were obtained in order to test the predictive power of the asymptotic variance (31). The covariance matrices Σ^* and Σ^G_Θ were estimated through the following steps: The matrices $A(\theta^*)$ and Σ_ψ (the limit of the matrix gain used in [5]) were estimated via Monte-Carlo. Estimation of $A(\theta^*)$ requires an estimate of θ^*; this was taken to be θ_N obtained using the Zap-Q algorithm with $\alpha_n = n^{-1}$ and $\gamma_n = n^{-0.85}$. This estimate of θ^* was also used to estimate the covariance matrix $\Sigma_\mathcal{E}$ defined in (32) using the batch means method.

The matrices Σ^G_Θ and Σ^* were obtained using (35) and (36). The matrices $\Sigma_\mathcal{E}$ and Σ^* were found to be full rank, with $\lambda(\Sigma^*)$ ranging from 10^{-2} to 10^7, and the matrix $A(\theta^*)$ was found to have condition-number of order 10^4.

Fig. 1 contains the histograms of the average rewards. Despite a badly conditioned matrix gain, it is observed in Fig. 1 that the average rewards obtained using the Zap-Q algorithms are better than the competitors.

Fig. 2 shows the histograms of $\sqrt{N}\big(\theta_N(8) - \theta^*(8)\big)$ (8^{th} element of the parameter vector), along with a plot of the theoretical prediction (in all cases where the eigenvalue test $\lambda(A_G) < -\frac{1}{2}$ is satisfied). When finite, the asymptotic covariance is a good predictor of the finite-n performance of the algorithm.

We also see in Fig. 2 what goes wrong when the asymptotic covariance is not finite, which is the case for Zap-Q with step-size $\alpha_n = 0.1/n$: we then have $G_A = -0.1\, I$, so that the eigenvalue test $\lambda(G_A) < -\frac{1}{2}$ fails. The empirical variance is huge in this case because of outliers. Based on the main result of [4], we expect that the mean-square error $\mathsf{E}[\|\tilde{\theta}_n\|^2]$ converges to zero at rate $O(n^{-0.2})$.

VI. Conclusion

Zap Q-learning algorithm for optimal stopping can be analyzed using tools similar to prior analysis for Zap Q-learning for MDPs in the tabular setting. The possibility of an extension was at first surprising, since it is known that theory for convergence of Q-learning algorithms in a function approximation setting is not guaranteed. Further research on Zap algorithms is required on three fronts: stability theory outside of the narrow framework considered here (part of this requires a more complete stability theory generalizing [2], [1]), techniques to reduce the computational burden of matrix gain algorithms, and the development in parallel of efficient exploration schemes for reinforcement learning.

References

[1] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency and Cambridge University Press (jointly), Delhi, India and Cambridge, UK, 2008.

[2] V. S. Borkar and S. P. Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.*, 38(2):447–469, 2000.

[3] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn. Zap Q-Learning for optimal stopping. In *Proc. ACC 2020 and arXiv:1904.11538*, 2019.

[4] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn. Explicit Mean-Square Error Bounds for Monte-Carlo and Linear Stochastic Approximation. *arXiv:2002.02584, and to appear AISTATS*, 2020.

[5] D. Choi and B. Van Roy. A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning. *Discrete Event Dynamic Systems: Theory and Applications*, pp 207–239, 2006.

[6] A. M. Devraj, A. Bušić, and S. Meyn. Fundamental design principles for reinforcement learning algorithms. In *Handbook on Reinforcement Learning and Control*. Springer, 2020.

[7] A. M. Devraj and S. P. Meyn. Fastest convergence for Q-learning. *ArXiv e-prints*, July 2017.

[8] A. M. Devraj and S. P. Meyn. Zap Q-learning. In *Proceedings of the 31st Intl. Conf. on Neural Information Processing Systems*, 2017.

[9] A. Nedic and D. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems: Theory and Applications*, 13(1-2):79–110, 2003.

[10] R. S. Sutton. Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3(1):9–44, 1988.

[11] V. B. Tadic and S. P. Meyn. Asymptotic properties of two time-scale stochastic approximation algorithms with constant step sizes. In *Proceedings of the ACC, 2003.*, pp 4426–4431. IEEE, 2003.

[12] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automat. Control*, 42(5):674–690, 1997.

[13] J. N. Tsitsiklis and B. Van Roy. Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Trans. Automat. Control*, 44(10):1840–1851, 1999.

[14] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, Cambridge, UK, 1989.

[15] C. J. C. H. Watkins and P. Dayan. *Q*-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[16] H. Yu and D. P. Bertsekas. Q-learning algorithms for optimal stopping based on least squares. In *Proc. European Control Conf.*, July 2007.