# A Simultaneous Deterministic Perturbation Actor-Critic Algorithm with an Application to Optimal Mortgage Refinancing

V. L. Raju Chinthalapati and S. Bhatnagar

***Abstract*— We develop a simulation-based, two-timescale actor-critic algorithm for infinite horizon Markov decision processes with finite state and action spaces, with a discounted reward criterion. The algorithm is of the *gradient ascent* type and performs a search in the space of stationary randomized policies. The algorithm uses certain simultaneous deterministic perturbation stochastic approximation (SDPSA) gradient estimates for enhanced performance. We show an application of our algorithm on a problem of mortgage refinancing. Our algorithm obtains the optimal refinancing strategies in a computationally efficient manner.**

## I. Introduction

There are many sequential decision tasks in which the consequences of an action emerge at a multitude of times after the action is taken and the problem is to find good strategies for selecting actions based on both their short and long term consequences. Such tasks are encountered in many fields such as economics, manufacturing, and artificial intelligence. These are usually formulated in terms of a dynamical system whose behavior unfolds over time under the influence of a decision maker's actions. The randomness involved in the consequences of the decision maker's actions is taken care of by modeling the dynamical system as a controlled stochastic process. Markov Decision Processes (MDP) [1] are a natural choice to model such systems and Dynamic Programming (DP) [2] is a general methodology for solving these. DP, however, requires complete knowledge of transition probabilities. Moreover, the computational requirements using DP are high in the presence of large state space. Recently there has been a lot of interest in simulation-based *Reinforcement Learning (RL)* algorithms for solving MDPs. These algorithms neither use transition probabilities nor estimate them and are useful in general for finding optimal control strategies in *real life* systems for which model information is not known. There are a certain class of (RL-based) algorithms that go under the name of actor-critic algorithms. These can be viewed as stochastic approximation versions of the classical policy iteration technique for solving MDPs.

Policy iteration is performed using two loops - (1) evaluate the stationary value function for a given policy and (2) update the policy by using the stationary value function obtained in loop 1. Konda et al. [3] propose a variant of the actor-critic algorithm based on stochastic approximation with two time scales. The idea is to operate the two loops above with different step-size schedules, so that the first (inner) loop moves on a faster effective time scale than the second.

Bhatnagar et al. [4] present a two-timescale simulation based algorithm that adapts the gradient descent simultaneous perturbation stochastic approximation (SPSA) technique [5], to the setting of simulation optimization. Bhatnagar et al. [6] present a two-timescale actor-critic algorithm for solving MDPs with finite state and compact action sets. The algorithm works with stationary deterministic policies instead of randomized as in [3].

In this paper, we consider infinite horizon MDPs with finite state and finite action sets. We present a two timescale simulation based, actor-critic algorithm that uses a one-simulation deterministic perturbation SPSA estimate. As with [3], our algorithm updates in the space of stationary randomized policies with the inner and outer loops similar to those in [6], except that we use a one-simulation SPSA based gradient search using deterministic perturbations. Our algorithm being for reward maximization, uses a *gradient ascent* update unlike the one in [6] that is designed for cost minimization. Our algorithm converges to an optimal point by requiring less computational effort. We then consider the important problem of optimal mortgage refinancing [7] as an application for our algorithm. Our algorithm computes the optimal refinancing strategies and is seen to show good performance.

The rest of the paper is organized as follows. The next section provides the setting that we consider. We present the algorithm in Section III. Section IV presents the convergence analysis. Section V provides the simulation results for the mortgage refinancing application. Finally, Section VI provides the concluding remarks.

## II. A Prelude

### A. Markov Decision Process (MDP)

Consider a process, observed at time epochs $t$ = 0, 1, ..., to be in one of the states $i \in$ S. Let S = $\{1, 2, \ldots, s\}$ denote the state space. After observing the state of the process, an action $a \in A = \{a^0, a^1, \ldots, a^{|A|}\}$ is taken, where A is the set of all possible actions. If the process is in state $i$ at time $n$ and action $a$ is chosen, then two things happen: (1) we receive a finite reward $R(i, a)$ and (2) the next state of the system is chosen according to the transition probabilities $P_{ij}(a)$. We let $X_n$ denote the state of the process at time $n$ and $a_n$ the action chosen at that time.

V. L. Raju Chinthalapati is with the Department of Mathematics, London School of Economics, Houghton Street, WC2A 2AE, London, U.K. v.l.chinthalapati@lse.ac.uk

S. Bhatnagar is with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore-560012, India shalabh@csa.iisc.ernet.in. Author acknowledges support through grant number SR/S3/EE/43/2002-SERC-Engg. from Department of Science and Technology, Government of India.

We assume that $\mid R(i,a) \mid < M \ \forall$ *i,a*. An admissible policy or simply a policy is any rule for selecting feasible actions. An important subclass of policies is the class of stationary policies. A policy is said to be stationary if the action it chooses at any time *n* depends only on the state of the process at that time. Hence, a stationary policy is a function $\pi : S \rightarrow A$. We assume for ease of exposition that all actions are feasible in each state. A stationary randomized policy can be considered as a map $\varphi : S \rightarrow \mathcal{P}(A)$ ($\mathcal{P}$ (...) = the space of probability vectors on "..."), which gives the conditional probabilities of $a^j$ given $X_n$ for all $0 \leq j \leq |A|$. We consider the infinite horizon discounted reward as our optimality criterion.

*1) Infinite Horizon Discounted Reward:* This criterion assumes a discount factor $\alpha, 0 < \alpha < 1$, and among all policies $\pi$, attempts to maximize $V^\pi$ where

$$V^\pi(i) = E^\pi[\sum_{n=0}^{\infty} \alpha^n R(X_n, a_n) \mid X_0 = i], \quad i \in S, \tag{1}$$

and $a_n$ are the actions, which are being employed according to policy $\pi$. The function $V^\pi : S \rightarrow \mathcal{R}$ is called the value function for policy $\pi$. The use of a discount factor is economically motivated by the fact that the value of money earned tomorrow is worth only a discounted amount as is today. While using stationary deterministic policies, the optimal value (of the value function) is:

$$V^*(i) = \max_\pi V^\pi(i), \quad i \in S. \tag{2}$$

An important equation that $V^*$ satisfies is the Bellman optimality equation [2]:

$$V(i) = \max_a [R(i,a) + \alpha \sum_j P_{ij}(a) V(j)], \quad i \in S. \tag{3}$$

The RHS of (3) corresponds to the "*optimal* sum of the single-stage reward in state $i$ and the discounted conditional expected value of the reward-to-go from the next stage onwards". Intuitively, this should be the same as the reward-to-go from state $i$ itself. The conditional expectation above requires knowledge of transition probabilities. RL based approaches replace the same with *simulated transitions*. Now the optimal decision problem turns out to be one of finding $V^*$. Amongst the well-known classical approaches for solving MDP, we have

*Policy iteration:* This starts with an initial stationary policy $\pi_0 : S \rightarrow A$. For finding an optimal policy, it does iteratively for $n \geq 0$ as follows:

Step 1: Given $\pi_n(.)$, iterate over $l = 0, 1, 2, \ldots$

$$V_n^{l+1}(i) = R(i, \pi_n(i)) + \alpha \sum_j P_{ij}(\pi_n(i)) V_n^l(j), \quad i \in S. \tag{4}$$

Let $V_n(i) \stackrel{\triangle}{=} \lim_{l \rightarrow \infty} V_n^l(i)$.

Step 2 : Find

$$\pi_{n+1}(i) \in argmax(R(i,.) + \alpha \sum_j P_{ij}(.) V_n(j)), \quad i \in S. \tag{5}$$

If $\pi_{n+1} \neq \pi_n$, set $n := n + 1$ and go to Step 1.

*Convergence issues:* In policy iteration, we evaluate a fixed stationary policy $\pi$ (cf. Step 1), that requires solving a linear system of equations (4). Here, we can define an operator $B^\pi$ as

$$B^\pi(V) = R(\pi) + \alpha P(\pi) V \tag{6}$$

For $\alpha < 1$, the operator $B^\pi$ is a contraction operator because $\forall V \in \mathcal{R}^{|S|}, \|B^\pi(V) - V^\pi\|_\infty \leq \alpha \|V - V^\pi\|_\infty$ and $V^\pi$ is the unique solution to $V = B^\pi(V)$. Of course, the operator $B^\pi$ requires knowledge of the state transition probabilities. For getting more insight on the above issues, refer [8], [9].

Note that in principle, the inner loop of the policy iteration procedure can take a long time to converge for any given policy update. A two time-scale simulation based approach for policy iteration is presented in [3] for MDPs with finite state and finite action spaces. Bhatnagar et al. [6] present a two-timescale simulation based, actor-critic algorithm that adapts the gradient descent SPSA [5], to the case of solving MDPs with finite state and compact action sets under the discounted cost criterion. The algorithm works with stationary deterministic policies instead of randomized as in [3]. In [4], a two-timescale SPSA algorithm with certain deterministic perturbation sequences in place of randomized [5] is found to exhibit good performance in a simulation optimization setting. For our actor-critic algorithm, we use a similar perturbation sequence as in [4] for a one sided [5] simultaneous perturbation algorithm that performs gradient search in the space of stationary randomized policies, the latter being necessitated because of the finite action space setting considered here. We use a *gradient ascent* (and not *descent*) algorithm as our aim here is to maximize rewards (and not minimize costs).

## III. The Algorithm

Let $\pi_i$ be the vector of probabilities of selecting actions in state $i \in S$ that can be written as $\pi_i = (\pi_i^{a^0}, \ldots, \pi_i^{a^{|A|}})$. Any stationary randomized policy $\pi$ can be identified with the vector $(\pi_1^{a^1}, \ldots, \pi_1^{a^{|A|}}, \ldots, \pi_{|S|}^{a^1}, \ldots, \pi_{|S|}^{a^{|A|}})$ with probabilities $\pi_i^{a^0}$ of selecting actions $a^0$, for all $i \in S$ getting automatically specified as $\pi_i^{a^0} = 1 - \sum_{j=1}^{|A|} \pi_i^{a^j}$. For $i \in S$, let $\pi_i(n)$ denote the $n$th update of $\pi_i$. Then $\pi(n)$ corresponds to the $n$th update of policy $\pi$.

**Definition:** An $m \times m$ $(m \geq 2)$ matrix $H$ is called a Hadamard matrix of order $m$ if its entries belong to $\{+1, -1\}$ and $H^T H = mI_m$, where $I_m$ denotes the identity matrix of order $m$. A Hadamard matrix is said to be normalized if all the elements of its first column and row are 1's. Normalized Hadamard matrices of order $m = 2^k, k \in N$ can be constructed sequentially in $k$ as under:

- For $k = 1$, let $H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.
- For $k > 1, H_{2^k} = \begin{pmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{pmatrix}$.

Let $H$ be a normalized Hadamard matrix of order $C$ with $C \geq |A|$. Let $\bar{H}$ be the matrix obtained from $H$ by picking any $(|A| - 1)$ columns from it other than the first column.

Let $\bar{H}(i)$ be the $i$th row of $H$, $i = 1, \ldots, C$. Let $\triangle\pi_i(n) = (\triangle\pi_i^{a^1}(n), \ldots, \triangle\pi_i^{a^{|A|}}(n)) \in \{\pm 1\}^{|A|}$ for $n \geq 0, i \in S$ be the 'deterministic' perturbation. For $i \in S, n \geq 0$, we let $\triangle\pi_i(n) \in \{\bar{H}(1), \ldots, \bar{H}(C)\}$, where $C = 2^{\lceil \log_2(|A|+1) \rceil}$. The desired perturbation sequence $< \triangle\pi_i(1), \ldots, \triangle\pi_i(n), \ldots >$ can be obtained by cyclically selecting $\triangle\pi_i(.)$ from the set $\{\bar{H}(1), \ldots, \bar{H}(C)\}$ in the same arbitrary order.

Fix $a^0 \in A$. For a $|A|$-vector $x$, let $\Gamma(x)$ denote its projection onto the simplex $D = \{[x_1, \ldots, x_{|A|}] \mid x_i \geq 0, \forall i, \sum_i x_i \leq 1\}$. Let $\bar{\pi}_i(n) = \Gamma(\pi_i^{a^1}(n) + \delta\triangle\pi_i^{a^1}(n), \ldots, \pi_i^{a^{|A|}}(n) + \delta\triangle\pi_i^{a^{|A|}}(n)) \in D$, where $\delta > 0$ is a small constant. Let $a(n), b(n)$ be decreasing sequences in $(0,1)$ satisfying $\sum_n a(n) = \sum_n b(n) = \infty$, $\sum_n (a(n)^2 + b(n)^2) < \infty$ and $a(n) = o(b(n))$.

Suppose, for any $i \in S$ and action chosen according to the law $\pi_i$, $\{\xi_n(i, \pi_i)\}$ be a sequence of i.i.d random variables with distribution $\sum_a P_{i\cdot}(a)\pi_i^a$. These correspond to the *simulated next states* in the algorithm. Suppose $V_n(i), n \geq 0$ be the stationary value function estimates corresponding to policy updates $\pi_i(n)$, $i \in S$. Let $L \geq 1$ be an arbitrarily chosen integer. Let $(\triangle\pi_i(n))^{-1} = (1/\triangle\pi_i^{a^1}(n), \ldots, 1/\triangle\pi_i^{a^{|A|}}(n))$. We have

$$\pi_i(n+1) = \Gamma[\pi_i(n) + a(n)\frac{V_{nL}(i)}{\delta}(\triangle\pi_i(n))^{-1}] \tag{7}$$

where, for $m = 0, 1, \ldots, L-1$,

$$V_{nL+m+1}(i) = (1 - b(n))V_{nL+m}(i) + b(n)[$$

$$R(i, \bar{\pi}_i(n), \xi_{nL+m}(i, \bar{\pi}_i(n))) + \alpha V_{nL+m}(\xi_{nL+m}(i, \bar{\pi}_i(n)))]. \tag{8}$$

Note that the cost $R(\cdot, \cdot, \cdot)$ also depends on the next state and is more general than the one considered in (1)-(5). Also, the update (7) is in the *gradient ascent* direction as our aim here is to maximize rewards.

## IV. Convergence Analysis

Let $\mathcal{F}_l$ denote the $\sigma$-field obtained from the history of the algorithm up to instant $l$ defined as $\mathcal{F}_l = \sigma(\hat{\pi}_i(p), V_p(i), p \leq l, i \in S, \xi_p(i, \hat{\pi}_i(p)), p < l, i \in S)$, $l \geq 1$. In the above, $\hat{\pi}_i(p) = \bar{\pi}_i(n)$ for $nL \leq p \leq (n+1)L - 1$. For a given policy $\pi$ and $V \in \mathcal{R}^{|S|}$, define

$$F_\pi(i, \pi_i, V) \stackrel{\triangle}{=} \sum_a \sum_j P_{ij}(a)\pi_i^a[R(i, a, j) + \alpha V(j)]$$

From (1), for any policy $\pi$, $V_\pi$ is the solution to the system of equations

$$V_\pi(i) = F_\pi(i, \pi_i, V_\pi). \tag{9}$$

One now obtains in a similar manner as *Theorem 1* of [10]

*Lemma 1:* The iterates $V_k(i)$ in (8) remain bounded with probability one.

Define:

$$M_i(l) = \sum_{k=0}^{l-1} b(k)(R(i, \hat{\pi}_i(k), \xi_k(i, \hat{\pi}_i(k))) + \alpha V_k(\xi_k(i, \hat{\pi}_i(k))) - \sum_{a \in A}\sum_{j \in S}[R(i, a, j) + \alpha V_k(j)]P_{ij}(a)\hat{\pi}_i^a(k)) \tag{10}$$

It is easy to see that $\{M_i(n), \mathcal{F}_n\}, i \in S$ are martingale sequences. Let $w_i(l)$ be the associated martingale difference terms above such that $M_i(l) = \sum_{k=0}^{l-1} b(k)w_i(k)$. Let us rewrite (8) as

$$V_{k+1}(i) = V_k(i) + b(k)(R(i, \hat{\pi}_i(k), \xi_k(i, \hat{\pi}_i(k))) + \alpha V_k(\xi_k(i, \hat{\pi}_i(k))) - V_k(i)) \tag{11}$$

where $\hat{\pi}_i(k) = \bar{\pi}_i(n)$ for $nL \leq k \leq (n+1)L - 1$. Now

$$V_{k+1}(i) = V_k(i) + b(k)(\sum_{a \in A}\sum_{j \in S} P_{ij}(a)\hat{\pi}_i^a(k)$$

$$(R(i, \hat{\pi}_i(k), j) + \alpha V_k(j) - V_k(i))) + b(k)w_i(k). \tag{12}$$

We have

*Lemma 2:* The martingales $\{M_i(n)\}$ converge with probability one.

*Proof:* Note that $w_i(n)$ defined above remain bounded with probability one. A direct calculation shows that in addition $\psi_i(n) = E[w_i(n)^2 | \mathcal{F}_n]$ remain bounded with probability 1. Since $\sum_k b(k)^2 < \infty$, we have $\sum_k b(k)^2 w_i(k)^2 < \infty$ with probability one. The foregoing then ensures that the corresponding quadratic variation processes of $\{M_i(l)\}$ ([11]) are convergent. Proposition $7 - 3(c) p.149 - 150$ of [11] then ensures the claim. ■

We treat the above recursions (7) and (12) as noisy approximations to a system of ordinary differential equations (ODEs) and analyze them as such. Let

$$\bar{\Gamma}(v(y)) \stackrel{\triangle}{=} \lim_{0 < \eta \to 0} (\frac{(\Gamma(y + \eta v(y)) - y)}{\eta}).$$

Consider the following system of differential equations

$$\dot{\pi}_i(t) = \bar{\Gamma}[\frac{F_{\bar{\pi}(t)}(i, \bar{\pi}_i(t), V_{\bar{\pi}(t)})}{\delta}(\triangle\pi_i(t))^{-1}] \tag{13}$$

$$\dot{\pi}_i(t) = \bar{\Gamma}[\nabla_i V_{\pi(t)}(i)] \tag{14}$$

Define $\{\bar{b}(n)\}$ as, $\bar{b}(0) = 1$ and for $n > 1, \bar{b}(n) = b([\frac{n}{L}])$, where $[\frac{n}{L}]$ denotes the integer part of $\frac{n}{L}$. Now it is clear that $a(n) = o(\bar{b}(n))$ and $\{\bar{b}(n)\}$ is a faster step-size sequence than $\{b(n)\}$. For value function updation, $\{\bar{b}(n)\}$ works as the natural step-size sequence because of the 'extra' $L$-step averaging involved in (8). Now define $\{t(n)\}$ as follows: $t(0) = 0$; $t(n) = \sum_{i=0}^{n-1} \bar{b}(i), n \geq 1$. Consider the ODEs:

$$\dot{\pi}_i(t) = 0,$$

$$\dot{V}_i(t) = F_{\pi(t)}(i, \pi_i(t), V) - V_i(t). \tag{15}$$

One can view recursions (7) and (8) as noisy approximations of the system of differential equations (15) along the faster timescale. Note that the solution to the system of ODEs (15), is nothing but the solution of the Poisson equation or the Bellman optimality equation (3) for a given policy $\pi$. It now

follows as in [3] that $V^{\pi}(i)$ are the unique asymptotically stable equilibrium points for the second equation in (15).

The link between the recursions of the algorithm and (15) can be explained via the continuous, piecewise linear interpolation $\bar{x}(.) = (\bar{x}_1(.), \ldots, \bar{x}_{|S|}(.))^T$ of (8) defined as follows: Set $\bar{x}_i(t(n)) = V_{nL}(i)$ with linear interpolation on $[t(n), t(n+1)]$ for $n \geq 0$. Let $x^n(.)$ denote the trajectory of (15) on $[t(n), \infty)$ with $x_i^n(t(n)) = V_{nL}(i)$ for $n \geq 0$. Let $[t] \stackrel{\triangle}{=} \max\{t(n) : t(n) \leq t\}$ and for $T > 0$, let $n_T \stackrel{\triangle}{=} \min\{m > n : t(m) \geq t(n) + T\}$.

*Lemma 3:* $\lim_{n\to\infty} sup_{t\in[t(n),t(n)+T]} \|x_n(t) - \bar{x}(t)\| = 0$ a.s.

*Proof:* For $m > n$,

$$\bar{x}_i(t(m)) = \bar{x}_i(t(n)) + \int_{t(n)}^{t(m)} F_{\bar{\pi}_t}(i, \bar{\pi}_t(i), \bar{x}(t))dt$$

$$+ \int_{t(n)}^{t(m)} [F_{\bar{\pi}_{t(n)}}(i, \bar{\pi}_{t(n)}, \bar{x}([t]) - F_{\bar{\pi}_t}(i, \bar{\pi}_t, \bar{x}(t))]dt$$

$$+(M_i(m) - M_i(n)). \tag{16}$$

$$= \bar{x}_i(t(n)) + \int_{t(n)}^{t(m)} F_{\bar{\pi}_t}(i, \bar{\pi}_t(i), \bar{x}(t))dt$$

$$+O(\sum_{i=n}^{n_T} \bar{b}(i)^2) + \sup_{n\leq m\leq n_T} (M_i(n_T) - M_i(n)). \tag{17}$$

Since, $t \in [t(l), t(l+1)], n \leq l \leq m, \bar{\pi}_t(i) = \bar{\pi}_{t(l)}(i), i \in S$ and from the definition of $F_{\pi}(.)$ we can easily observe the *Lipschitz* property,

$$\|F_{\bar{\pi}_{t(l)}}(i, \bar{\pi}_{t(l)}(i), \bar{x}(t(l)) - F_{\bar{\pi}_{t(l)}}(i, \bar{\pi}_{t(l)}(i), \bar{x}(t))\|$$

$$\leq \alpha\|\bar{x}(t(l)) - \bar{x}(t)\| \tag{18}$$

$$x_i^n(t(m)) = x_i^n(t(n)) + \int_{t(n)}^{t(m)} F_{\bar{\pi}_t}(i, \bar{\pi}_t(i), x^n(t))dt. \tag{19}$$

$$\|x_i^n(t(n)) - \bar{x}_i(t(m))\| \leq$$

$$\| \int_{t(n)}^{t(m)} (F_{\bar{\pi}_t}(i, \bar{\pi}_t(i), x^n(t)) - F_{\bar{\pi}_t}(i, \bar{\pi}_t(i), \bar{x}(t)))dt\|$$

$$+O(\sum_{i=n}^{n_T} \bar{b}(i)^2) + \sup_{n\leq m\leq n_T} (M_i(n_T) - M_i(n)). \tag{20}$$

$$\|x_i^n(t(n)) - \bar{x}_i(t(m))\| \leq \alpha \int_{t(n)}^{t(m)} \|x_i^n(t) - \bar{x}_i(t)\|dt$$

$$+O(\sum_{i=n}^{n_T} \bar{b}(i)) + \sup_{n\leq m\leq n_T} (M_i(n_T) - M_i(n)). \tag{21}$$

Now from the *discrete Gronwall inequality*, we have

$$\sup_{t\in[t(n),t(n)+T]} \|x_i^n(t) - \bar{x}_i(t)\| \leq$$

$$\alpha T(O(\sum_{i=n}^{n_T} b(i)^2) + \sup_{n\leq m\leq n_T} (M_i(n_T) - M_i(n)). \tag{22}$$

The first term in parentheses on the RHS above is the contribution of the discretization error and goes to zero as $n \to \infty$ because of $\sum b(n)^2 < \infty$. The second term is the error due to noise and goes to zero a.s. by Lemma 2. The claim follows. ■

*Lemma 4:* We have $\|V_n(i) - F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})\| \to 0$ as $n \to \infty$.

*Proof:* Let us view the policy updation recursion (7) from the time scale of $\{\bar{b}(n)\}$. Since $a(n) = o(\bar{b}(n))$, the recursion takes the form:

$$\pi_i(n+1) = \Gamma[\pi_i(n) + \bar{b}(n)o(1)].$$

The rest follows from Lemma 3 and Theorem 1, pp.339 of [12]. ■

The rest of the analysis is geared towards showing convergence of the algorithm to an optimal policy. Let us rewrite (7) in the following form

$$\pi_i(n+1) = \Gamma[\pi_i(n) + a(n)\frac{F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})}{\delta}(\triangle\pi_i(n))^{-1}$$

$$+a(n)(\frac{V_{nL}(i) - F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})}{\delta}(\triangle\pi_i(n))^{-1})]$$

$$= \Gamma[\pi_i(n) + a(n)\frac{F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})}{\delta}(\triangle\pi_i(n))^{-1} + o(1)], \tag{23}$$

the last follows from Lemma 4. Now from the result on pp. 191-194 of [13] for projection based algorithms, it can be verified as before that (7) asymptotically tracks the trajectories of (13) on the slower time-scale. Recall that $C$ is the order of the normalized Hadamard matrix. We have

*Lemma 5:* (1) For every $l, m \in \{1, \ldots, |A|\}, l \neq m$ and for any $k \in \mathcal{N}, i \in S$ $\sum_{n=k}^{k+c} \frac{\triangle\pi_i^m(n)}{\triangle\pi_i^l(n)} = 0$,

(2) For every $q \in \mathcal{N}$, $\sum_{n=k}^{k+c} \frac{1}{\triangle\pi_i^q(n)} = 0$.

*Proof:* Both claims follow from the construction of Hadamard matrix of order $C$, see [4] for a detailed proof. ■

*Lemma 6:* Let $\pi(n) \in D^{|S|}$. For any $\bar{\pi}(n) = \Gamma(\pi(n) + \delta\triangle\pi(n))$, we have with probability one, $\lim_{\delta\to 0} \|\frac{F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})}{\delta}(\triangle\pi_i(n))^{-1} - \nabla_i V_{\bar{\pi}(n)}(i)\| = 0$.

*Proof:* For any $\bar{a} \in A$, one needs to show

$$\lim_{\delta\to 0} |\frac{F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})}{\delta\triangle\pi_i^{\bar{a}}(n)} - \frac{\partial V_{\pi(n)}(i)}{\partial\pi_i^{\bar{a}}(n)}| = 0.$$

Assume initially that $\bar{\pi}(n) \in (D^{|S|})^o$ (interior of the set $D^{|S|}$). Then

$$\frac{F_{\bar{\pi}(n)}(i, \bar{\pi}_i(n), V_{\bar{\pi}(n)})}{\delta\triangle\pi_i^{\bar{a}}(n)} = \sum_{a,j}(P(i,j,a)(\pi_i^a(n) + \delta\triangle\pi_i^a(n))$$

$$[R(i,a,j) + \alpha V_{\pi(n)}(j)])/(\delta\triangle\pi_i^{\bar{a}}(n)). \tag{24}$$

The claim now follows from Lemma 5. For $\pi(n)$ on the boundary of $D^{|S|}$, a similar argument as in [6] settles the claim. ■

Finally, one can show using standard arguments that

*Theorem 1:* Let $J$ be the set of local maxima for (14). Then for all $\epsilon > 0, \exists \delta_0 > 0$ such that $\forall \delta \in (0, \delta_0]$, the proposed algorithm converges to $J^\epsilon$ (the set of all points that are within an $\epsilon$-neighborhood of the set $J$) w.p.1.

## V. Application to Mortgage refinancing

Mortgage is a kind of long term loan to finance the purchase of real estate, usually with prescribed payment periods and interest rates. The mortgagor (borrower) assures the mortgagee (lender) by securing the loan against the value of the real estate. Since there are many different schemes for the payment of interest and repayment of the principle, mortgage could be much more complex than a loan. The mortgage market consists of two types of interest rates, fixed interest rate and variable interest rate. In the fixed interest case, interest rate is fixed for all payment periods, however, the same varies over different payment periods in the variable interest case. The interest rates usually depend on the base interest rate that is derived from the central bank's interest rate.

Refinancing a mortgage is nothing but paying off an existing loan and replacing it with new mortgage contract. Refinancing is commonly practiced in real life by mortgagors with the hope of obtaining a lower interest rate, to shorten the term of the mortgage, etc. We consider a refinancing model that is some what similar to Pliska's [7] optimal refinancing model.

In this model, we consider mortgage contracts having $N$ contracted coupon payments (for the time interval of $d$ days) each of amount $C$ dollars. We always assume that the contract will be initiated by a fixed mortgage interest rate $m_r$. Let $P_b$ be the principle balance and $P_b(n)$ be the same after $n$th coupon payment is made. Immediately after the $n$th coupon payment, the mortgagor either continues with the existing mortgage or goes for refinancing the principle balance $P_b(n)$ for another $N$-period mortgage by finding another mortgagee. While refinancing principle balance $P_b$, some fraction ($f_b$) of $P_b$ could be done with variable interest rate ($v_r$) and the remaining fraction $(1 - f_b)$ could be done with fixed interest rate. We assume the transaction cost $T_c(P_b)$ is incurred if and when the principle balance $P_b$ is refinanced. $T_c(.)$ is a specified, deterministic function. The mortgagor might choose to refinance several times before the loan is ultimately paid off. Below we provide the complete model description.

- We assume a riskless base interest rate process $r = \{r_t; t = 1, 2, \ldots\}$, the base interest rate of $r_t$ depends on $r_{t-1}$. Here $r_t$ is a random variable that is uniformly distributed on $\{r_{t-1}-2a, r_{t-1}-a, r_{t-1}, r_{t-1}+a, r_{t-1}+2a\}$, where $a$ is an arbitrary constant.
- Fixed interest rate $m_t = r_t + b$, where $b$ is a constant.
- Variable interest rate $v_t = r_t + c$, where $c$ is the random variable uniformly distributed on {2,2.5,3.0,3.5,4.0}. This is so chosen since the variable interest rate usually lies between $r_t + 2$ and $r_t + 4$.
- Opportunities for new mortgage (from various lenders, who offer both fixed and variable interests) appear in the market according to a Poisson process with rate $\lambda$.

The mortgagor's objective is to minimize the expected present value of the cash flow. The mortgagor is the learning agent in the system and uses reinforcement learning to learn the optimal refinancing strategy. The MDP has four state variables: $r_t$, $f_b$, $v_t$ and $n$, respectively. In the simulation experiments, we consider $P_b(0) = \$200,000, N = 60$, $r_0 = 0.06$, $a = b = 0.005$, $r_t$ is bounded between $0.04$ and $0.12$, and $f_b \in \{0.1, 0.2, \ldots, 0.9, 1.0\}$. We consider that transaction cost is equivalent to two percent of $P_b$ and $\lambda = 1$ lender appearances/day. The action set is $A = \{0.0, 0.1, \ldots, 0.9, 1.0\}$, where action $0.0$ means not to choose refinance and the other actions are meant to choose refinance with a fraction $f_b$ variable interest rate. The cost (or reward) function chosen is

$$\begin{aligned} R(i, \pi(i), j) &= 0 \text{ if } 0 \le t < d \\ &= C + Cf_b v_t + C(1 - f_b) r_t, \text{ if } t \ge d \\ &= 0.02(P_b - C) + C + Cf_b v_t(t/d) + \\ &\quad C(1 - f_b) r_t(t/d), \text{ if } 0 \le t < d. \end{aligned} \tag{25}$$

Note that the basic problem here is a finite horizon MDP. However, we use a similar technique as in [7] to convert the same to an infinite horizon MDP, thus enabling the use of the proposed actor-critic algorithm.

Table I shows the converged refinancing strategies (rounded off to second digit after decimal place) for the mortgagor for two randomly picked states (0.08, 0.2, 0.115, 34) and (0.07, 0.6, 0.09, 14). Fig. 1 shows the convergence and stability of learning refinancing policy at state $(0.07, 0.4, 0.095, 11)$ using our algorithm. In Figs. 2 and 3, we show for given components $r_t$, $v_t$ and $n$ of states, the variation of the optimal reward as obtained from our algorithm with the state component $f_b$, for two classes of states $(0.07, f_b, 0.09, 0)$ and $(0.08, f_b, 0.09, 0)$, respectively. Note that the optimal reward dips initially in both cases and subsequently shows an increase. The optimal reward is high for high values of $f_b$. This behavior is along expected lines.

## VI. Conclusions

We developed a simulation based two-timescale actor-critic algorithm for infinite horizon discounted reward Markov decision processes. The algorithm does gradient search in the space of stationary randomized policies and uses a one-simulation simultaneous deterministic perturbation gradient estimate. The algorithm has less computational requirements and exhibits fast convergence. We then studied an application using our algorithm for the problem of optimal mortgage refinancing to find the optimal refinancing strategies and reward. As future work, it would be worth exploring other reinforcement learning based algorithms. For instance, the algorithms in [**?**], for the simulation optimization setting, perform second order Newton based search. Reinforcement learning variants of similar algorithms could be developed and tried.

| Fraction of Principle Balance With $v_t$ | Policy of Mortgagor for state (0.08, 0.2, 0.115, 34) | Policy of Mortgagor for state (0.07, 0.6, 0.09, 14) |
|---|---|---|
| 0.0 | 0.42 | 0.18 |
| 0.1 | 0.04 | 0.15 |
| 0.2 | 0.12 | 0.21 |
| 0.3 | 0.30 | 0.01 |
| 0.4 | 0.08 | 0.10 |
| 0.5 | 0.00 | 0.09 |
| 0.6 | 0.01 | 0.07 |
| 0.7 | 0.00 | 0.06 |
| 0.8 | 0.03 | 0.00 |
| 0.9 | 0.00 | 0.03 |
| 1.0 | 0.00 | 0.10 |

TABLE I

POLICIES OF MORTGAGOR AT SOME INDIVIDUAL STATES

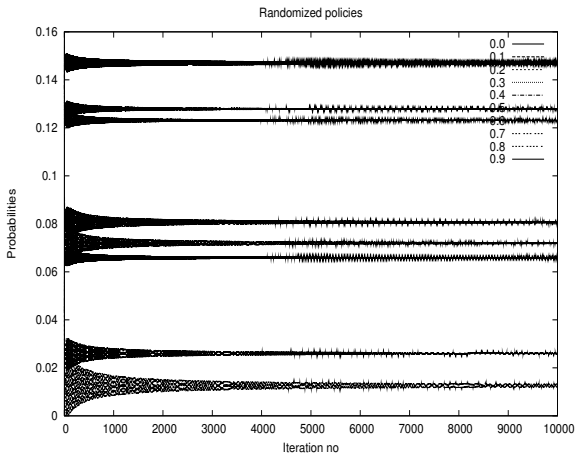


Fig. 1. Convergence of Mortgagor randomized policies at state (0.07, 0.4, 0.095, 11)

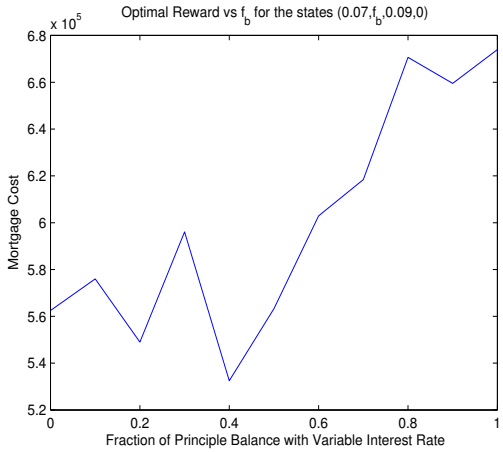


Fig. 2. Optimal reward as a function of $f_b$ for states $(0.07, f_b, 0.09, 0)$

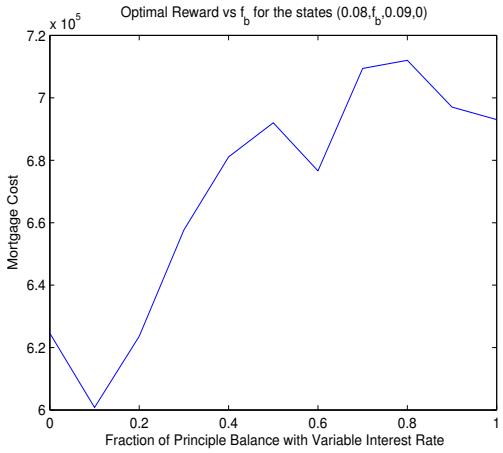


Fig. 3. Optimal reward as a function of $f_b$ for states $(0.08, f_b, 0.09, 0)$

## VII. ACKNOWLEDGEMENTS

We would like to thank Martin Anthony and Adam Ostaszewski for their useful comments on mortgage refinancing part of this work. The first author acknowledges research funding from London School of Economics.

## REFERENCES

[1] Puterman. *Markov Decision Processes*. Wiley Inter-science, 1994.
[2] S.M. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, 1983.
[3] V. R. Konda and V. S. Borkar. Actor-critic type learning algorithms for Markov decision processes. *SIAM Journal on Control and Optimization*, 38:94–123, 1999.
[4] Shalabh Bhatnagar, M. C. Fu, S. I. Marcus, and I. J. Wang. Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation*, 2002.
[5] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Contr.*, 37(3), 1992.
[6] Shalabh Bhatnagar and Shishir Kumar. A simultaneous perturbation stochastic approximation based actor-critic algorithm for Markov decision processes. *IEEE Transactions on Automatic Control*, 49(4):592–598, 2004.
[7] S. R. Pliska. Mortgage valuation and optimal refinancing. In *Proceedings of Stochastic Finance 2004, Lisbon, Portugal(Springer-Verlag*, 2004.
[8] S. Singh. Learning to solve Markovian decision processes. Ph.d dissertation, University of Michigan, Ann Arbor, 1994.
[9] D. P. Bertsekas and J. Tsitsiklis. *Neuro-dynamic Programming*. Athena Scientific, Boston, MA, USA, 1996.
[10] J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine Learning*, (8):279–292, 1994.
[11] J. Neveu. *Discrete Parameter Martingales*. North Holland, Amsterdam, 1975.
[12] M. W. Hirsch. Convergent activation dynamics in continuous time networks. *Neural Networks*, (2):331–349, 1989.
[13] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.