Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 305–315
ISSN 1896-7094

Truncated Importance Sampling for Reinforcement Learning with Experience Replay

Paweł Wawrzyński, Andrzej Pacut

Warsaw University of Technology, Institute of Control and Computation Engineering
Nowowiejska 15/19, 00-665 Warsaw, Poland
p.wawrzynski@elka.pw.edu.pl, a.pacut@ia.pw.edu.pl
WWW home page: http://staff.elka.pw.edu.pl/~pwawrzyn/

Abstract. Reinforcement Learning (RL) is considered here as an adaptation technique of neural controllers of machines. The goal is to make Actor-Critic algorithms require less agent-environment interaction to obtain policies of the same quality, at the cost of additional background computations. We propose to achieve this goal in the spirit of *experience replay*. An estimation method of improvement direction of a changing policy, based on preceding experience, is essential here. We propose one that uses truncated importance sampling. We derive bounds of bias of that type of estimators and prove that this bias asymptotically vanishes. In the experimental study we apply our approach to the classic Actor-Critic and obtain 20-fold increase in speed of learning.

1 Introduction

We consider the problem of a controller (or a decision maker) that optimizes its reactive policy in a poorly structured and initially unknown environment. The general solution to this problem is Reinforcement Learning (RL). Algorithms in that area can be viewed as computational processes that transform observations of states, actions, and rewards, into policy parameters. Popular RL algorithms, like Q-Learning [17], and Actor-Critic methods [2, 6, 7], process the data sequentially. A single observation is utilized for an adjustment of the algorithms' parameters and becomes unavailable for further use. Let us call such methods *single-adjustment algorithms*. This sequential processing approach follows a common understanding that applications of RL to real life learning control problems require large amounts of data which cannot be kept in a limited memory assigned to the algorithm.

However, there are methods, not of single-adjustment type, that require much less environment steps to obtain a policy of the same quality. They achieve that at the cost of collecting data and their extensive processing. Let us call them *multi-adjustment algorithms* as they are not limited to a small number of operations after each environment step. Note that we do not identify single-adjustment- and on-line methods neither we identify multi-adjustment- and off-line algorithms. We are here interested *only* in on-line algorithms in the sense that they should optimize the policy as the agent-environment interaction is going

on. The difference lies rather in that how busy is the computer that optimizes the policy during that interaction.

Multi-adjustment algorithms can be roughly divided into three classes. The most obvious way to design the computations for a multi-adjustment algorithm is to estimate the model of the environment dynamics and use it in some variant of dynamic programming [14, 5, 1]. The second approach to multi-adjustment algorithms is as follows. Assume there exists a quality index $J : \Re^{n_\theta} \mapsto \Re$ that assigns to each policy parameter $\theta \in \Re^{n_\theta}$ a value $J(\theta)$ that evaluates the policy based on θ. Let us denote by $\widehat{J}_t(\theta)$ an estimator of $J(\theta)$ based on the course of the MDP before time t. The objective of data processing can be to optimize the policy by maximizing $\widehat{J}_t(\theta)$ with respect to θ [13, 10]. Within the third approach, a multi-adjustment algorithm is constructed in reference to a single-adjustment method: It takes historical events and applies to them the operations of the original single-adjustment method as if the events have just taken place. This idea of repeating many similar operations to the same event, called *experience replay* [3], was popular a few years ago but seems to receive less attention recently. In our paper we show how to apply this idea to a wide class of RL methods that includes Actor-Critics.

The approach proposed in the present paper is as follows. A single-adjustment RL algorithm defines a drift of the policy parameter θ. Within the process of computing previous experience, the original path of θ is *simulated* such that the parameter is adjusted along estimators of the direction of the drift. The estimators are based on the observations of preceding agent-environment interaction collected in a database. Our main concern here is how to estimate the direction of the drift on the basis of historical data.

2 Problem formulation

We will consider the standard RL setup [15]. A Markov Decision Process is a tuple $\langle \mathcal{S}, \mathcal{A}, P_s, r, P_0, \mathcal{S}^* \rangle$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively; $\{P_s(\cdot|s,a) : s \in \mathcal{S}, a \in \mathcal{A}\}$ is a set of state transition distributions; we write $s_{t+1} \sim P_s(\cdot|s_t, a_t)$ and assume that each P_s is a density. The immediate reward, r_t, depends on the action and the next state, $r_t = r(a_t, s_{t+1})$. P_0 is the distribution of initial states of each episode and $\mathcal{S}^*$ is the set of terminal states. Whenever $s_t \in \mathcal{S}^*$, the next state s_{t+1} is drawn from P_0.

Actions are generated according to a policy, π, which is a family of distributions parameterized by the state and a *policy vector* $\theta \in \Re^{n_\theta}$, namely

$$a_t \sim \pi(\cdot\,; s_t, \theta).$$

The objective of reinforcement learning is, in general, to make the policy maximize future rewards by optimizing θ. The strict meaning of this goal may be specified in various ways. We may require the policy to maximize the average reward or to maximize the expected sum of rewards within an episode. Alternatively, we may require the policy to maximize the sum of future discounted rewards expected in each state.

We are interested in applications of the MDP framework to adaptive control tasks. We thus require the learning algorithm to be fast, i.e. to obtain a satisfying policy after as few control steps as possible. The control cost should be minimized and the controlled machine should be kept from being damaged by inappropriate actions. We assume that the time span of learning is short enough to keep the entire agent-environment interaction history in a database. This history is available between each pair of consecutive steps for a limited number of operations aiming at policy optimization.

Below we analyze a large class of the existing incremental RL methods suitable for policy determination in simulations and propose a way of their acceleration based on a more extensive data processing. Our intention is to design methods that obtain a satisfying control policy after a much smaller amount of control time but not necessarily after a smaller amount of computation.

2.1 Single-adjustment Methods with Actor

Here we analyze a large class of RL algorithms defined by the following features:

1. Actions are generated by a stationary policy (Actor), i.e. a distribution π parametrized by state s_t and the policy parameter θ, namely

$$a_t \sim \pi(\cdot\,; s_t, \theta).$$

2. A visit in state s_t causes a modification of the policy parameter θ by a vector

$$\beta_t^\theta \widehat{\phi}_t.$$

   $\widehat{\phi}_t$ on the average indicates the direction in which θ assures larger future rewards expected in state s_t. $\{\beta_t^\theta, t = 1, 2, \dots\}$ is a vanishing sequence of step-sizes.
3. The algorithm may compute $\widehat{\phi}_t$ with the use of an auxiliary parameter $\upsilon \in \mathfrak{R}^{n_\upsilon}$ (like the weights of Critic, i.e. a value-function approximator). A visit in state s_t results in a modification of υ by a vector

$$\beta_t^\upsilon \widehat{\psi}_t.$$

   $\widehat{\psi}_t$ on the average points into the direction where υ assures better quality of $\widehat{\phi}_t$. $\{\beta_t^\upsilon, t = 1, 2, \dots\}$ is a vanishing sequence of step-sizes.
4. The vectors $\widehat{\phi}_t$ and $\widehat{\psi}_t$, while different from one another, are of the same form

$$G_t(\theta, \upsilon) \sum_{k \geq 0} (\alpha\rho)^k z_{t,k}(\theta, \upsilon) \tag{1}$$

   where G_t is a certain vector, $\alpha \in [0, 1)$, $\rho \in [0, 1)$, and $z_{t,k} \in \mathfrak{R}$ is unknown until instant $t + k + 1$.

Typically, the step-sizes β_t^θ and β_t^υ satisfy the standard stochastic approximation conditions [8], namely

$$\sum_{t\geq 1}\beta_t = +\infty, \qquad \sum_{t\geq 1}\beta_t^2 < +\infty. \tag{2}$$

Obviously a number of RL algorithms fit into the above schema. In the classic Actor-Critic algorithm with $TD(\lambda)$-errors [6] we have

$$\widehat{\phi}_t = \frac{\partial \ln \pi(a_t; s_t, \theta)}{\partial \theta^T} \sum_{k\geq 0} (\gamma\lambda)^k \frac{\beta_{t+k}^\theta}{\beta_t^\theta} \widehat{e}_{t+k}, \qquad \widehat{\psi}_t = \frac{\partial \bar{V}(s_t; \upsilon)}{\partial \upsilon^T} \sum_{k\geq 0} (\gamma\lambda)^k \frac{\beta_{t+k}^\upsilon}{\beta_t^\upsilon} \widehat{e}_{t+k},$$

where $\bar{V}$ is an approximation of the value function parametrized by υ, $\gamma \in [0, 1)$ is a discount factor, $\lambda \in (0, 1)$, and

$$\widehat{e}_t = r_t + \gamma \bar{V}(s_{t+1}; \upsilon) - \bar{V}(s_t; \upsilon) \tag{3}$$

is the *temporal difference*. More recent Actor-Critics [6, 7] generally fit into the discussed schema.

Analysis of RL methods usually assumes that for each policy parameter θ, the sequence of states $\{s_t, t = 1, 2, \dots\}$ forms an aperiodic and irreducible Markov chain with a stationary distribution. In our presentation of the mechanics of the incremental RL algorithms, we will also assume that the states visited when the policy parameter θ is applied are drawn independently from a stationary distribution. A reader interested in a deeper analysis and proofs of convergence is referred to [9, 7].

Let us analyze the average direction of $\widehat{\phi}_t$ and $\widehat{\psi}_t$. Namely, let ϕ be a function defined as

$$\phi(s, \theta, \upsilon) = \mathcal{E}_{\theta,\upsilon,\beta}\left(\widehat{\phi}_t \middle| s_t = s\right). \tag{4}$$

The definition of ϕ is based on the assumption that θ, υ, and the step-sizes remain constant when $\widehat{\phi}_t$ is calculated. In fact they slightly vary and each $\widehat{\phi}_t$ is in fact a biased estimator of $\phi(s_t, \theta, \upsilon)$ for θ and υ used at time t. However, this bias is small and since the dynamics of the parameters decreases in time, the bias asymptotically vanishes. ϕ averaged over the steady state distribution, approximates the drift of θ and the smaller the step-sizes, the better the approximation.

The drift of υ may be analyzed in a similar way. Namely, let ψ be a function defined as

$$\psi(s, \theta, \upsilon) = \mathcal{E}_{\theta,\upsilon,\beta}\left(\widehat{\psi}_t \middle| s_t = s\right). \tag{5}$$

As above, the definition of ϕ requires that θ, υ, and the step-sizes remain constant during the time when $\widehat{\psi}_t$ is computed. The approximation of the drift of υ is defined by ψ averaged over the steady state distribution.

The incremental RL algorithms make their parameters θ and υ follow the directions defined by estimators of $\phi(s, \theta, \upsilon)$ and $\psi(s, \theta, \upsilon)$, respectively, averaged over the steady-state distribution.

2.2 Experience Replay Based Acceleration

The main idea analyzed in this paper is to apply to the agent's experience the same treatment as a single-adjustment algorithm with Actor would do, but more intensively. Namely, let us consider a certain algorithm of this type. It defines vectors $\widehat{\phi}_t$ and $\widehat{\psi}_t$ and corresponding functions ϕ and ψ, respectively. A generic accelerated version of this algorithm has the form of the following loop:

1. Draw and execute an action, $a_t \sim \pi(\cdot\,; s_t, \theta)$.
2. Register s_t, a_t, θ, the reward received and the next state in a database.
3. Perform a certain number of times:
   3.1. Draw a state $s_i, 1 \leq i < t$ from the database.
   3.2. Adjust θ along an estimator of $\phi(s_i, \theta, \upsilon)$.
   3.3. Adjust υ along an estimator of $\psi(s_i, \theta, \upsilon)$.
4. Assign $t := t + 1$ and repeat from Point 1.

In words, the initial single-adjustment algorithm uses ϕ to define a drift of the policy parameter θ. The direction of this drift is an approximation of a policy gradient or a natural policy gradient. The drift itself can be *simulated* in a parallel computation process (Point 3. above) while the Markov Decision Process is going on. This way the original algorithms can be substantially accelerated.

In the following section, we discuss importance sampling as the main tool to construct estimators of ϕ and ψ. In Section 4, those estimators are constructed.

3 Importance Sampling Estimation

Let $g(\cdot\,;\theta)$ be a density of random elements a of values in $\mathcal{A}$ parametrized by a vector $\theta \in \Theta \subset \mathfrak{R}^{n_\theta}$. Suppose for a given parameter θ_0 an action a_0 is drawn from $g(\cdot\,;\theta_0)$. The action a_0 yields a certain vector $d(a_0)$ where $d : \mathcal{A} \mapsto \mathfrak{R}^n$. We are interested in estimation of the expected $d(a)$ for actions drawn with the use of an arbitrary parameter θ, namely

$$\mathcal{E}_\theta d(a) = \int\limits_{\mathcal{A}} d(\alpha) g(\alpha;\theta)\,\mathrm{d}\alpha = D(\theta). \tag{6}$$

In order to construct an estimator of $D(\theta)$, we may apply the estimation technique called *importance sampling* [12] frequently applied in RL [11, 13, 10, 16]. Namely, given θ_0 and θ, the statistic

$$d(a_0)\, g(a_0;\theta)/g(a_0;\theta_0) \tag{7}$$

is an unbiased estimator of $D(\theta)$ since

$$\mathcal{E}_{\theta_0}\left(d(a_0)\,\frac{g(a_0;\theta)}{g(a_0;\theta_0)}\right) = \int d(\alpha)\frac{g(\alpha;\theta)}{g(\alpha;\theta_0)}\, g(\alpha;\theta_0)\,\mathrm{d}\alpha = D(\theta). \tag{8}$$

A well-known drawback of estimator (7) is its excessive variance when distributions $g(\cdot\,;\theta_0)$ and $g(\cdot\,;\theta)$ are far from one another [19]. A simple yet satisfying in

practice way of fighting the variance problem is to truncate the density ratio in (7). Namely, the modified estimator is of the form

$$d(a_0)\min\left\{g(a_0;\theta)/g(a_0;\theta_0),\, b\right\} \tag{9}$$

for constant $b > 1$. The above *truncated IS estimator* has also been applied in [16, 18]. Let us analyze its properties. Obviously if $\|d\|$ is bounded, then also the variance of (9) is bounded. Importantly, its bias happens to be bounded as well if only g satisfies certain regularity conditions and θ is close enough to θ_0. Let us first specify those conditions. Namely, for the purpose of this discussion, a density g will be called (M_1, M_2)-*regular* if it has the following features:

(i) for every $a \in \mathcal{A}$ the mapping $\theta \mapsto g(a;\theta)$ is continuous and differentiable,
(ii) the trace of the Fisher information, namely $\operatorname{tr} I(\theta) = \mathcal{E}_\theta \frac{\partial \ln g(a;\theta)}{\partial \theta} \frac{\partial \ln g(a;\theta)}{\partial \theta^T}$, is bounded by M_1,
(iii) absolute eigenvalues of the Hesian $\frac{\partial^2 \ln g(a;\theta)}{\partial \theta^T \partial \theta}$ are bounded by M_2.

The below proposition establishes bounds of the bias of estimator (9).

Proposition 1. *Let the density g parameterized by θ be (M_1, M_2)-regular for certain M_1, M_2. Then, for all θ_0, θ,*

$$\left\| \mathcal{E}_{\theta_0}\left(d(a_0)\min\left\{\frac{g(a_0;\theta)}{g(a_0;\theta_0)}, b\right\}\right) - D(\theta)\right\| \le M P_\theta\left(\frac{g(a;\theta)}{g(a;\theta_0)} > b\right).$$

where $M = \sup_{a\in\mathcal{A}} \|d(a)\|$.

A slightly different form of the above fact is proved on p. 34 of reference [18]. ■
Note that P_θ is the probability defined by the condition that the random value a is drawn from the density parameterized by θ. Another proposition defines a bound of P_θ.

Proposition 2. *For each $M_1, M_2 > 0$, there exists $M > 0$ such that if the density g is (M_1, M_2)-regular than, for all θ_0, θ,*

$$P_\theta\big(g(a;\theta)/g(a;\theta_0) > b\big) \le M(\ln b)^{-2}\|\theta - \theta_0\|^2.$$

This proposition is proved on p. 35 of reference [18]. ■
In the present work we shall also discuss a special case of the analyzed problem, where d is a function of a finite sequence of independent random elements. Obviously, both the above propositions apply to this case because a sequence of random vectors is also a random vector. However, a joint density of a random sequence need not be (M_1, M_2)-regular even if densities of all its elements are. This rises difficulties in application of Proposition 2. The proposition below treats this issue.

Proposition 3. *For each $M_1, M_2 > 0$, there exists $M > 0$ such that if all densities in the sequence $g^1, \ldots, g^k$ are (M_1, M_2)-regular than for each set of pairs $(\theta^i, \theta_0^i), i = 1, \ldots, k$*

$$P_{\theta^1 \ldots \theta^k}\left(\prod_{i=1}^k \frac{g^i(a^i;\theta^i)}{g^i(a^i;\theta_0^i)} > b\right) \le \frac{k^2 M}{(\ln b)^2}\sum_{i=1}^k \|\theta^i - \theta_0^i\|^2.$$

provided random elements $a^i, i = 1, \ldots, k$ are drawn independently.

Proof (sketch): Let us denote

$$A = \left\{ \prod_{i=1}^{k} \frac{g^i(a^i;\theta^i)}{g^i(a^i;\theta_0^i)} > b \right\}, \quad A_i = \left\{ \frac{g^i(a^i;\theta^i)}{g^i(a^i;\theta_0^i)} > \sqrt[k]{b} \right\}, i = 1, \ldots, k.$$

We have

$$\Omega - A \supseteq \bigcap_{i=1}^{k} (\Omega - A_i) \quad \text{and hence} \quad 1 - P_{\theta^1 \ldots \theta^k}(A) \geq \prod_{i=1}^{k} (1 - P_{\theta^i}(A_i)).$$

After simple transformations we obtain

$$P_{\theta^1 \ldots \theta^k}(A) \leq \sum_{i=1}^{k} P_{\theta^i}(A_i).$$

Application of Proposition 2 completes the proof. ■

4 Experience Replay

In this section, we introduce estimators of $\phi(s_i, \theta, \upsilon)$ and $\psi(s_i, \theta, \upsilon)$ and discuss their properties. These estimators will play the role of stochastic gradients. In general, stochastic gradients are required to be of bounded variance and (asymptotically) unbiased. While unbiased estimators of ϕ and ψ may be of excessive variance, their modifications will be presented that will be of bounded variance and asymptotically unbiased.

In Section 2.1 we defined ϕ and ψ of the form (4) and (5), respectively, for $\widehat{\phi}_t$ and $\widehat{\psi}_t$ having, for constant step parameters, the same generic form

$$G_t(\theta, \upsilon) \sum_{k \geq 0} (\alpha\rho)^k z_{t,k}(\theta, \upsilon) \tag{10}$$

where G_t is a vector known at time t, $\alpha \in [0, 1), \rho \in [0, 1)$ is a given constant, and in a sequence $\{z_{t,k}(\theta, \upsilon), k \geq 0\}$ the value $z_{t,k}$ is known at time $t + k + 1$. Because ϕ and ψ have the same form, further in this section we will focus on ϕ, understanding that exactly the same can be said about ψ.

Let us first denote by $Geom(\rho)$ the geometric distribution with parameter $\rho \in [0, 1)$. That is, random variable K of values in $\{0, 1, \ldots\}$ has distribution $Geom(\rho)$, i.e. $K \sim Geom(\rho)$, iff

$$P(K = m) = (1 - \rho)\rho^m \quad \text{for} \quad m = 0, 1, \ldots$$

A simple property of that distribution is that $P(K \geq m) = \rho^m$. Let an estimator of $\phi(s_i, \theta, \upsilon)$ have the form

$$\widehat{\phi}_i^b(\theta, \upsilon) = G_i(\theta, \upsilon) \sum_{k=0}^{K} \alpha^k z_{i,k}(\theta, \upsilon) \min \left\{ \prod_{j=0}^{k} \frac{\pi(a_{i+j}; s_{i+j}, \theta)}{\pi(a_{i+j}; s_{i+j}, \theta_{i+j})}, b \right\}. \tag{11}$$

Because of truncation, it is easy to verify that its variance is bounded. However, for the same reason, it is also biased. Fortunately, the bias vanishes when the parameters θ_{i+j} of distributions that generated actions get closer to the analyzed value θ. The following proposition clarifies this issue.

Proposition 4. *Suppose G_i and $z_{i,k}$ are uniformly bounded and distributions $\pi(\cdot\,; s_{i+j}, \cdot)$ are (M_1, M_2)-regular for certain M_1, M_2 and all $j \geq 0$. Then, there exists a positive constants c such that*

$$\|\mathcal{E}\widehat{\phi}_i^b(\theta, \upsilon) - \phi(s_i, \theta, \upsilon)\| \leq c\delta^2$$

for all i, θ, υ and

$$\delta = \sup_{j \geq 0} \|\theta - \theta_{i+j}\|.$$

Proof (sketch): Each element of $\widehat{\phi}_i^b$ is a truncated importance-sampling estimator of $\mathcal{E}_{\theta,\upsilon}\big(G_i(\theta, \upsilon) z_{i,k}(\theta, \upsilon) | s_i\big)$. Propositions 1 and 3 define a bound of the bias of these estimators, which is of the form

$$\frac{M(k+1)^2}{(\ln b)^2} \sum_{j=0}^{k} \|\theta - \theta_{i+j}\|^2$$

Because the sum in the above formula is no greater than $(k+1)\delta^2$, we obtain

$$\begin{aligned}\|\mathcal{E}\widehat{\phi}_i^b(\theta, \upsilon) - \phi(s_i, \theta, \upsilon)\| &\leq \mathcal{E}\left(\sum_{k=0}^{K} \alpha^k \frac{M(k+1)^3\delta^2}{(\ln b)^2}\right) \\ &= \sum_{k \geq 0} (\rho\alpha)^k \frac{M(k+1)^3\delta^2}{(\ln b)^2} = \delta^2 \frac{M}{(\ln b)^2} \sum_{k \geq 0} (\rho\alpha)^k (k+1)^3.\end{aligned}$$

Because $\rho\alpha \in (0, 1)$, the value of the last sum is finite which completes the proof. ■

Implementation of the generic algorithm presented in Sec. 2.2 has to meet several postulates. The most important requirement for any algorithm based on stochastic approximation is boundedness of the variance of gradient estimators. That requirement is satisfied when the estimators of ϕ and ψ have the form (11). Second, we want the algorithm to inherit the limit properties from the original single-adjustment method.

The drawback of estimators (11) is that they are biased. However, they can be made asymptotically unbiased. Namely, suppose only a certain constant number, N, of recent events are kept in the database. Then, at the moment t the indexes i are drawn randomly from the set $\{t-N+1, t-N+2, \ldots, t\}$. Let us analyze δ_t as the upper bound of $\|\theta - \theta_{i+j}\|$ for i drawn at moment t. Because the step-sizes β_t^θ vanish in time, the changes of the policy parameter, θ, also vanish. Therefore, the value δ_t must vanish as well, the conditions of Proposition 4 are met and the estimators $\widehat{\phi}_i^b$ and $\widehat{\psi}_i^b$ are asymptotically unbiased.

5 Experimental Study

We apply the idea of simulation of parameters' path in a parallel computation process to the classic Actor-Critic (AC) [6]. For the purpose of this paper we shall call the resulting algorithm the replaying Actor-Critic. We apply this algo-

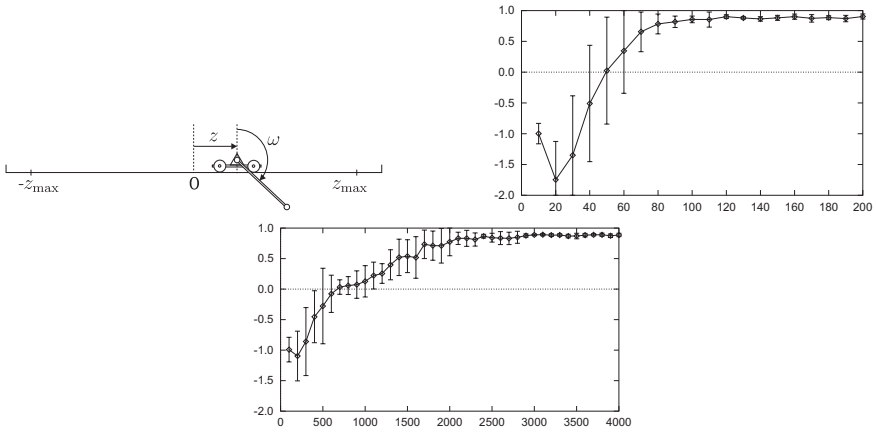

Fig. 1. Actor-Critics for Cart-Pole Swing-Up. *Left:* The plant. *Center:* The classic AC, the average reward vs. trial number. Each point averages 100 consecutive trials and the curve averages 10 runs. The one-sigma limits are calculated to assess run-to-run variability of trial averages. *Right:* The replaying AC, the average reward vs. trial number. Note that the number of trials in this figure is about 20 times smaller than that of the top figure for the classic method. Because of much shorter learning time, each point averages only 10 consecutive trials; the curve averages 10 runs. The one-sigma limits are wider than in the top figure because 10 times less trials are averaged.

rithm to a simple benchmark problem — Cart-Pole Swing-Up [4]. The plant here consists of a cart moving along a track, and a pole hanging freely from the cart (see Fig. 1). It is controlled by force applied to the cart. The control objective is to avoid hitting the track bounds, swing the pole, turn it up, and stabilize upwards. In our experiments the controlled plant is emulated, i.e. simulated in real time. A quantum of plant's time is equal to a quantum of the corresponding computer time. That means that the computer has a lot of spare time that can be devoted to parallel computations. The setup of our experiments is designed to closely resemble a situation in which control of a physical machine is optimized in real time by means of a certain form of adaptation.

The setting of our experiments is the same as of those presented in [18]. The result of application the replaying Actor-Critic is presented in the form of learning curve on the left-hand side of Fig. 1. For comparison we also apply the classic Actor-Critic to the same plant. The corresponding learning curve is presented in the center of Fig. 1.

The replaying AC (see Fig. 1) exhibits a satisfying behavior after just 90 trials (about 20 minutes of the plant's time) while the classic algorithm obtains the same behavior after about 2300 trials. Our experiments with the classic Actor-Critic may also be compared to those presented in [4]. The algorithm introduced there implements an indirect adaptive control method—it estimates a model of an unknown plant and optimizes its control on the basis of the model while the plant is controlled. This algorithm, applied to the Cart-Pole Swing-Up, obtained satisfactory results after about 750 trials (about 2 hours of the plant's real time).

6 Conclusions

In this paper we developed an idea of experience replay based speedup of incremental Actor-Critic-like reinforcement learning methods. A drift of parameters driven by a given incremental RL method may be simulated in a parallel computation process while the agent-environment interaction is going on. This simulation is based on estimators of the direction of parameter drift induced by the original algorithm. We introduced the appropriate estimators based on truncated importance sampling and analyzed their properties, namely bounded variance and asymptotically vanishing bias. In the experimental study we applied our approach to the classic Actor-Critic and obtained a satisfying controller of the Cart-Pole Swing-Up in 20 minutes of time of this plant which is about 20 shorter time than the original Actor-Critic requires.

References

1. P. Abbeel, M. Quigley, & A. Y. Ng: 2006, Using Inaccurate Models in Reinforcement Learning,. *Int. Conf. on Machine Learning.*
2. A. G. Barto, R. S. Sutton, & C. W. Anderson: 1983, Neuronlike Adaptive Elements That Can Learn Difficult Learning Control Problems. *IEEE Trans. on Systems, Man, and Cybernetics,* vol. SMC-13:834-846.
3. P. Cichosz: 1999, An analysis of experience replay in temporal difference learning. *Cybernetics and Systems,* 30:341-363.
4. K. Doya: 2000, Reinforcement learning in continuous time and space. *Neural Computation,* 12, pp. 243-269.
5. M. Kearns & S. Singh: 2002, Near-Optimal Reinforcement Learning in Polynomial Time. *Machine Learning,* Volume 49, Issue 2-3, pp. 209-232.
6. H. Kimura & S. Kobayashi: 1998, An Analysis of Actor/Critic Algorithm Using Eligibility Traces: Reinforcement Learning with Imperfect Value Functions. *Int. Conf. on Machine Learning.*
7. V. R. Konda & J. N. Tsitsiklis: 2003, Actor-Critic Algorithms. *SIAM Journal on Control and Optimization,* Vol. 42, No. 4, pp. 1143-1166.
8. H. J. Kushner & G. G. Yin: 1997, *Stochastic Approximation Algorithms and Applications,* Springer: New York.
9. P. Marbach & J. N. Tsitsiklis: 2001, Simulation-Based Optimization of Markov Reward Processes. *IEEE Trans. on Automatic Control,* Vol. 46, No. 2, pp. 191-209.
10. L. Peshkin & Ch. R. Shelton: 2002, Learning from Scarce Experience. *Int. Conf. on Machine Learning,* pp. 498-505.
11. D. Precup, R. S. Sutton, & S. Singh: 2000, Eligibility Traces for Off-Policy Policy Evaluation. *Int. Conf. on Machine Learning,* Morgan Kaufmann.
12. R. Rubinstein: 1981, *Simulation and The Monte Carlo Method.* New York, Wiley.
13. Ch. R. Shelton: 2001, Policy Improvement for POMDPs Using Normalized Importance Sampling. *Int. Conf. on Uncertainty in Artificial Intelligence,* pp. 496-503.
14. R. S. Sutton: 1990, Integrated Architectures For Learning, Planning, and Reacting Based on Approximating Dynamic Programming. *Int. Conf. on Machine Learning,* pp. 216-224, Morgan Kaufmann.

15. R. S. Sutton & A. G. Barto: 1998, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.
16. E. Uchibe & K. Doya: 2004, Competitive-Cooperative-Concurrent Reinforcement Learning with Importance Sampling. *Int. Conf. On Artificial Intelligence.*
17. C. Watkins & P. Dayan: 1992, Q-Learning. *Machine Learning*, vol. 8, pp. 279-292.
18. P. Wawrzyński, Intensive Reinforcement Learning, *PhD Dissertation,* Institute of Control and Computation Engineering, Warsaw University of Technology, 2005.
19. P. Wawrzyński & A. Pacut: 2006, Balanced Importance Sampling Estimation. *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU),* pp. 66-73.