Two-Time-Scale Online Actor-Critic Paradigm Driven by POMDP

Bo Liu[1], Haibo He[1,2], *Member, IEEE*, and Daniel W Repperger[3], *Fellow, IEEE*

***Abstract*—In this paper, we analyze a class of actor-critic algorithms under partially observable Markov decision process (POMDP) environment. Specifically, in this work we focus on the two-time-scale framework in which the critic uses a temporal difference with neural network (NN) as nonlinear function approximator, and the actor is updated using greedy algorithm with the stochastic gradient approach. Instead of the common construction of hidden state estimator, we develop the idea originated from Singh, Jaakkola and Jordan (1994) into an online action-dependent actor-critic paradigm. This framework explores the ability of the adaptive dynamic programming (ADP) approach in POMDP environment without implementing extra architectures such as state estimators. Both the theoretical analysis and simulation studies validate that the framework performs effectively under certain assumptions given in this paper.**

I. Introduction

In certain real-world applications, it may not be possible for the agent to know the environment thoroughly and completely. Let's consider, for example, a robot equipped with sensors that return an observation [1]. The robot may not know its exact location, but only the limited information from a camera with which sensory observations are recorded. Instead of telling the robot its exact location, the sensors observe whether it is in a corridor, an open room, a T-junction, or, the fact that there is a wall to its left [2] . This category of problems is referred in the community as various terms such as "hidden state", or less often used "incomplete perception", or "perceptual aliasing"[1]. Mathematically speaking, this category of problems, known as partially observable Markov decision process (POMDP), introduces the Hidden Markov Model (HMM) into reinforcement learning [3]. The POMDP differs from HMM in that there is not as yet known computationally tractable algorithm for POMDPs [4].

So far, there are two approaches to tackle this kind of problems [1]: state-free stochastic policies and policies with internal states. For instance, the policies with internal states approaches are rooted in the system theory: the observability and state observer theory of linear systems. Such approaches integrate the state estimation with learning control and construct an internal representation of the system states by feature extraction from past internal representations and the observation states from sensors [5]. This involves a discussion of the length of the past time-window frame: how long should the critic keep in mind? Recurrent neural network has been suggested to be a powerful tool in memory [2], which is successfully demonstrated in [6].

The prevailing research on POMDP up to now is based on building up a state estimator to output a "belief state" to estimate the underlying state [7],[8]. Briefly speaking, A belief state is a probability distribution over states of the environment, indicating the likelihood, given the agent's past experience that the environment is actually in each of those states. By introducing this term to summarize the past experience, the hidden states are kept Markov [2]. The state estimator can be constructed using either the estimated model or Bayes rule. However, an inevitable drawback of this kind of approaches lies in that state estimation is notorious for its heavy computational cost and infeasibility for online learning [4]. Another concern is that the state estimate component puts strong restrictions on the environment [4], which usually becomes an unrealistic assumptions in many real applications. There are other issues worth attention intrinsically aroused by state estimation, such as convergence. Implementing the state estimator into the system introduces the convergence issue to not only the state estimator itself but also other components in the system, like the critic and actor.

In this paper, we analyze a class of actor-critic algorithms under a POMDP environment. Specifically, in this work we focus on the two-time-scale framework in which the critic uses temporal difference with neural network (NN) as nonlinear function approximator, and the actor is updated using greedy algorithm with the stochastic gradient approach. Instead of the common construction of hidden state estimator, we develop the idea originated in [4] from semi-batched Q-learning to online action-dependent actor-critic structure. This is an intriguing topic since it explores the capability of the adaptive dynamic programming (ADP) approach in POMDP environment without implementing extra architectures such as state estimators. To our best knowledge, this is the first time to extend the theoretical framework in [4] to an online action-dependent actor-critic paradigm with neural network as nonlinear function approximator.

The rest of this paper is organized as follows. Section 2 introduces the framework and detailed analysis of each network and the whole system. In Section 3, simulation results with detailed analysis of the proposed method are presented. Finally, a conclusion is given in Section 4.

[1] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA.
[2] Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA.
[3] Air Force Research Laboratory, Wright-Patterson Air Force Base, Dayton, Ohio 45433.
Emails: bliu@stevens.edu, he@ele.uri.edu, hhe@stevens.edu, Daniel.Repperger@wpafb.af.mil

978-1-4244-6453-1/10/$26.00 ©2010 IEEE

II. SYSTEM ARCHITECTURE AND CONVERGENCE ANALYSIS

Fig. 1 shows the main architecture of the actor-critic paradigm adopted in this paper. In this architecture, chain backpropagation rule is the key to the training and adaptation of the parameters of all these networks (action network, and critic network) [9]. Since the learning in the action network is similar to the classic ADP model, we will only focus on the system architecture and the convergence issue. In order to focus on the learning principle, we assume neural networks with a simple 3-layer nonlinear network (with 1 hidden layer) are used in both action network and critic network. We would like to note that the learning principles discussed here can also be generalized to any arbitrary function approximator by properly applying backpropagation rule.

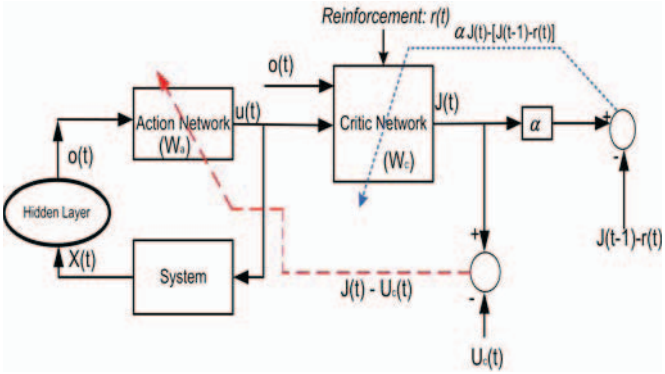

Fig. 1. The framework from the actor-critic perspective

A. Convergence of Each Network

In the following, we first give the assumptions for the convergence of each network at a time, and then prove the convergence of the algorithm in an analytical way. When one component (e.g., the critic network) is updating, other components are considered as stationary, namely, their weight matrices do not change.

For clear presentation, we adopt the following notations in this paper.

- $l_a(t)$ the learning rate of action network at time t
- $l_c(t)$ the learning rate of critic network at time t
- $x(t)$ the actual state of the system at time t
- $o(t)$ the observation of the system at time t
- $r_x^u(t)$ the sample average of actual payoffs received on executing action u in state x_t at time t
- $u(t)$ the action the actor takes in observation o_t at time t

Assumption 1: (Stepsize) The step sizes $l_c(t)$, $l_a(t)$ are positive, non-increasing, and predetermined chosen prior to execution of the algorithm. Further, they satisfy

$$\lim_{t\to\infty} l_c(t) = 0, \sum_{t=0}^{\infty} l_c(t) = \infty, \sum_{t=0}^{\infty} l_c^2(t) < \infty$$
$$\lim_{t\to\infty} l_a(t) = 0, \sum_{t=0}^{\infty} l_a(t) = \infty, \sum_{t=0}^{\infty} l_a^2(t) < \infty \quad (1)$$

Remark: This assumption concerns the step size parameters and is standard for any stochastic approximation algorithm.

Assumption 2: (MDP property) [10] The underlying MDPs are ergodic for every stationary policy, namely, the MDP is irreducible and aperiodic. Furthermore, there is a unique stationary distribution that satisfies $\pi e = 1, \pi P = \pi$ with $\pi(i) > 0$ for all i; here, π is a finite or infinite vector, depending on the cardinality of state space S. And we further assume:

(1) the instantaneous reinforcement signal (state transition cost) $r_{i,j}(t)$ satisfies

$$\forall i, j, t, \text{var}(r_{i,j}(t)) < \infty \quad (2)$$

(2) the discount factor $\gamma < 1$. (if $\gamma = 1$, all policies lead to a cost free absorbing terminal state w.p.1 [4], which we will not discuss in this paper).

Remark: This is similar to the assumption made in [10], however, Markov chain with absorbing state is not suitable for our discussion in this paper. This assumption places constraints on the underlying infinite-horizon discounted MDP. In essence, we assume that the MDP is irreducible and aperiodic, the steady-state variance of transition costs is finite, and that the cost-to-go from any state is well defined and finite.

We assume that the objective of the algorithm is to minimize a nonnegative cost-to-go function $J : H \to [0, \infty)$, or vice versa, maximize a negative counterpart $J : H \to (-\infty, 0]$.

Assumption 3: (Value function property) [11]: J is continuously differentiable and its derivative satisfies the Lipschitz condition

$$\|\nabla J(x) - \nabla J(x')\| \le K \|x - x'\|, \forall x, x' \in H \quad (3)$$

Where K is some nonnegative constant.

We will now move on with our assumptions. We assume that all random variables introduced so far are viewed as defined on a probability space (Ω, F, P). We introduce $\{P_t\}$, an incremental sequence of σ-fields contained in subfields of P_t , which is used to describe the history of the algorithm up to time t . Before we introduce the next assumption, we would like to note the measure-theoretic terminology that a random variable Z is P_t measurable meaning that Z is completely determined by the history depicted by P_t [11].

Assumption 4: [11] Let

$$P_t = \{r(t-1), r(t-2), \cdots, l_c(t-1), l_c(t-2), \cdots, l_a(t-1), l_a(t-2), \cdots\} \quad (4)$$

Where $(t-1)$, $(t-2), \cdots$ represent different time steps of the past history. All the random variables are allowed to be P_t measurable, $l_c(t)$, $l_a(t)$ are assumed to be non-negative and mutually conditionally independent given P_t.

Remark: This assumption allows for the possibility of deciding whether to update a particular component at time t, based on the past history of the process [11].

To examine the learning process taking place in the critic network, we define the following objective function for the critic network

$$\tilde{E}_c = \frac{1}{2}\sum_i p_i \left[\alpha J(t) - J(t-1) + r(t)\right]^2 \quad (5)$$
$$= \frac{1}{2} E_o \left[\alpha J(t) - J(t-1) + r(t)\right]^2$$

It can be seen that (5) is an "averaged" square error in a statistical sense. To obtain a local minimum of the "averaged" square error, the Robbins-Monro algorithm can be applied here by first taking a derivative of this error with respect to (w.r.t) the parameters, which are the weights in the critic network in this case, Let

$$e_c = \alpha J(t) - \left[J(t-1) - r(t)\right] \quad (6)$$

Since J is smooth in $\tilde{w}_c$, and $\tilde{w}_c$ is bounded, then we have

$$\frac{\partial \tilde{E}_c}{\partial \tilde{w}_c} = E_o\left[e_c \frac{\partial \tilde{e}_c}{\partial \tilde{w}_c}\right] \quad (7)$$

According to Robbins-Monro algorithm, the root (can be global or local) of $\frac{\partial \tilde{E}_c}{\partial \tilde{w}_c}$ as a function of $\tilde{w}_c$ can be obtained by the following recursion, on condition that the root exists and that the stepsizes meet the standard requirements, i.e. Assumption 1. And Δw_c will converge to a stationary minimum.

$$\Delta w_c(t) = -l_c(t)\nabla_{w_c} J(t) \quad (8)$$
$$= -l_c(t)\left\{\alpha J(t) - \left[J(t-1) - r(t)\right]\right\}\frac{\partial J(t)}{\partial w_c(t)}$$

Since the action network is almost the identical as above, we do not take detailed procedure for its convergence proof. Interested readers can refer to [9] for details.

B. Convergence Of The Whole System: Two-time-scale Problem

As for the asynchronous implementation, the update of the value of a particular state may proceed independently of the updates of the values of other states. [11] showed that as long as each state is updated infinitely often, i.e., each state is visited with probability 1 (w.p.1), and each action is tried infinitely number of times in each state, then the asynchronous algorithm eventually converges to the optimal value function. Also, the relation of time scale of different network is a big issue. As pointed out in [3], in this work we adopt the assumption that the policy will only change slightly. This assumption's merit lies in that it makes possible that the learner can assess the effect of the policy refinement on the average reward since the Bellman Error term contains information not available to the learner.

There are two aspects of views of the actor-critic time scale problem. The first holds that the whole structure can be viewed as a loop diagram in Fig. 2. From this perspective, we can consider the internal loop includes the system model and critic network, and the outer loop includes the action network and the internal loop. To be stabilized, the internal loop must operate at a higher frequency than the outer loop, according to feedback theory [12]. The second aspect is that the actor-critic can be viewed as a kind of adaptive tracking control [13]. It

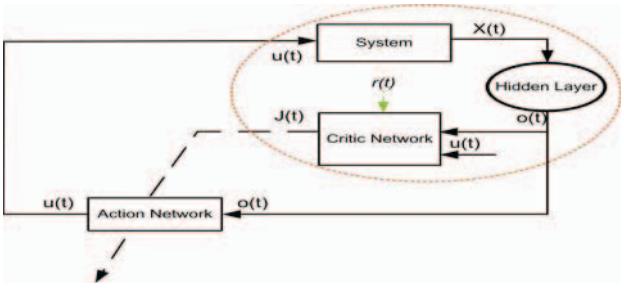

Fig. 2. The framework from a control community perspective

can be understood as the critic is trying to track the actor's policy adaptively. The two-time-scale involves a tradeoff, or pessimistically speaking, a dilemma. Let's talk about the learning rate of the critic first. On one hand, the learning rate should satisfy Assumption 1, namely, non-increasing and decaying with time. On the other hand, to track changes of the statistics of the environment that are slow relative to the step-size of the algorithm convergence of stochastic approximation driven by stationary or asymptotically stationary ergodic noise, constant step-sizes should be used [14].

However, to assume that the environment changes slowly relative to the step sizes employed by users is such a restrictive assumption that is difficult to be satisfied in many applications. To this end, in our current work we simply assume the learning rate of the actor decays slowly compared with the learning rate of the critic. If the statistics of the actor is changed at a rate slower than that of the critic in stochastic approximation, it will be practical to study the tracking ability of stochastic approximation with decreasing stepsizes,

Assumption 5: (Two-time-scale) [13],[15] Let $l_c(t)$ and $l_a(t)$ be the learning rate of the critic network and action network of the POMDP system, respectively. Then $\exists d > 0$,

$$\sum_t \left(\frac{l_a(t)}{l_c(t)}\right)^d < \infty \quad (9)$$

Remark: For finite MDPs, according to the convergence law of positive term series, the assumption can be weakened as

$$\lim_{t\to\infty} \frac{l_a(t)}{l_c(t)} = 0 \quad (10)$$

Equations (1) and (9) indicate this is a contraction mapping with radius of convergence < 1, *i.e.* $l_a(t)$ should be faster than $l_c(t)$, so $|l_a(t)| < |l_c(t)| < 1$.

C. Convergence of POMDP Problem

Here we present the analysis of the actor-critic paradigm driven by POMDP environment. Our idea is originated in [4], which set a rough theoretical framework for semi-batched Q-learning without state estimators.

Theorem 1: (The online Action-dependent version) Given Assumption 1, 2, 3, 4, the gradient-based action-dependent actor-critic paradigm using nonlinear function approximator will converge to the optimal value of the following system equations w.p.1, i.e., $\forall o \in \mathrm{O}$

$$V(o)=\sum_{x}P^{\pi}(x|o)\left[r_{x}^{\pi}(t)+\gamma\sum_{o'}P^{\pi}(x,o')V(o')\right] \tag{11}$$

$$P^{\pi}(x,o')=\sum_{x'}\left(P^{\pi}(x,x')P(o'|x')\right) \tag{12}$$

Further, if Assumption 5 is satisfied, then there exists a persistent policy π so that during learning, the actor-critic algorithm will converge to the solution of the following system equation w.p.1, i.e.,$\forall o\in \mathrm{O}$

$$\begin{aligned}&J(t)=\sum_{x}P^{\pi}(x|o,u)\\&\cdot\left[r_{x}^{u}(t)+\gamma\sum_{o'}P^{u}(x,o')\max_{u'}J_{o'}^{u'}(t+1)\right]\end{aligned} \tag{13}$$

For the online version, the estimated value function is the following

$$J(t)=\sum_{x}P^{\pi}(x|o,u)\left[r_{x}^{u}(t)+\gamma\sum_{o'}P^{u}(x,o')J(t+1)\right] \tag{14}$$

$P^{\pi}(x|o,u)$ is the asymptotic probability under optimal policy π , and $P^{u}(x,o')$ is computed in equation 12.

Proof: The (state, action) pair becomes (state, observation, action) pair in the POMDP situation. First we introduce some indicator functions. Let

$$\chi_{t}(o,u)=\begin{cases}1 & \text{at time } t \text{ action } u \text{ was}\\ & \text{executed in observation } o\\ 0 & \text{otherwise}\end{cases}$$

$$\chi_{t}(x|o,u)=\begin{cases}1 & \text{at time } t \text{ the actual hidden was } x\\ & \text{when the obsevable pair was } (o,u)\\ 0 & \text{otherwise}\end{cases}$$

$$\chi_{t}(o,o'|u)=\begin{cases}1 & \text{at time } t \text{ transition took}\\ & \text{place from } o \text{ to } o' \text{ if } u \text{ was taken}\\ 0 & \text{otherwise}\end{cases}$$

$$\begin{aligned}&J(t+1)=(1-\chi_{t}(o,u)l_{c}(t))J(t)\\&+\chi_{t}(o,u)l_{c}(t)\left[\sum_{x}\tfrac{\chi_{t}(x|o,u)}{\chi_{t}(o,u)}r_{x}^{u}(t)+\gamma\sum_{o'}J(t)\tfrac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}\right]\end{aligned} \tag{15}$$

Also the following assumption holds

$$V_{t}(o)=\max_{u}J^{u}(t) \tag{16}$$

The bias can be expressed as

$$\begin{aligned}&F_{t}(o,u)=\sum_{x}\tfrac{\chi_{t}(x|o,u)}{\chi_{t}(o,u)}r_{x}^{u}(t)\\&+\gamma\sum_{o'}\left\{V_{t+1}(o',u')\tfrac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}\right\}-J(t)\\&=\sum_{x}\tfrac{\chi_{t}(x|o,u)}{\chi_{t}(o,u)}r_{x}^{u}(t)\\&+\gamma\sum_{o'}\left\{\max_{u'}J^{u'}(t+1)\tfrac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}\right\}-J(t)\end{aligned} \tag{17}$$

Then we have

$$\begin{aligned}&F_{t}(o,u)=\sum_{x}\left(\tfrac{\chi_{t}(x|o,u)}{\chi_{t}(o,u)}r_{x}^{u}(t)-P^{\pi}(x|o,u)r_{x}^{\pi}(t)\right)\\&+\gamma\sum_{o'}\tfrac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}\left[V_{t}(o')-\hat{V}_{t}(o')\right]\\&+\gamma\sum_{o'}\left[\tfrac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}-P^{u}(o,o'|\pi)\right]\hat{V}_{t}(o')\end{aligned} \tag{18}$$

Where

$$P^{u}(o,o'|\pi)=\sum_{x}P^{\pi}(x|o,u)\left[\sum_{x'}\left(P^{u}(x,x')P(o'|x')\right)\right] \tag{19}$$

And the $\hat{V}_{t}$ is the observation of V_{t}.
We further define the following two terms:

$$P_{t}(x|o,u):=E\left\{\frac{\chi_{t}(x|o,u)}{\chi_{t}(o,u)}\right\} \tag{20}$$

$$P_{t}(o,o'|o,u):=E\left\{\frac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}\right\} \tag{21}$$

Therefore, the expected value of $F_{t}(o,u)$ can be bounded as

$$\begin{aligned}&\|E\{F_{t}(o,u)\}\|\\&\leq\left\|E\left\{\sum_{x}\left(\tfrac{\chi_{t}(x|o,u)}{\chi_{t}(o,u)}-P^{\pi}(x|o,u)\right)r_{t}^{\pi}(x)\right\}\right\|\\&+\gamma\left\|V_{t}-\hat{V}_{t}\right\|\\&+\gamma\left\|E\left\{\sum_{o'}\left[\tfrac{\chi_{t}(o,o'|u)}{\chi_{t}(o,u)}-P^{u}(o,o'|\pi)\right]\hat{V}_{t}(o')\right\}\right\|\\&\leq\gamma\left\|V_{t}-\hat{V}_{t}\right\|+c\varepsilon_{t}\end{aligned} \tag{22}$$

Where ε_{t} satisfies

$$\varepsilon_{t}=\max\left(\varepsilon_{t}^{1},\varepsilon_{t}^{2}\right) \tag{23}$$

$$\begin{aligned}\varepsilon_{t}^{1}&=\max_{(x,o,u)}|P_{t}(x|o,u)-P^{\pi}(x|o,u)|\\\varepsilon_{t}^{2}&=\gamma\max_{(o,o',u)}|P_{t}(o,o'|o,u)-P^{u}(o,o'|\pi)|\end{aligned} \tag{24}$$

Thus the bias $F_{t}(o,u)$ is a contraction mapping.

Next we will show that the variance of $F_{t}(o,u)$ is also bounded. Since the discounted factor γ is smaller than 1, then the variance is bounded since the variance of the sample probabilities is bounded. Therefore, by theorem 1 of [10], for $\forall\varepsilon>0$, with probability $1-\varepsilon$, $\forall\delta>0$, $\lim_{t\to\infty}|J(t)-J(\infty)|<\delta$,

thus the online action-dependent version converges w.p.1 [4]. In brief, it is proved that the difference caused by the online updating asymptotically diminishes to zero in the time limit sense almost surely (a.s).

III. Simulation Analysis

A. System Model

In this section, we will demonstrate the performance of the proposed approach in a Furuta pendulum swing-up and balancing task. In this paper, the case under study is identical to the one in [16]. The model is described with the differential equations as follows.

$$M(q)\frac{\partial^2 q}{\partial t^2} + P\left(q, \frac{\partial q}{\partial t}\right) = Bu \tag{25}$$

$$M(q) = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{12}(q) & M_{22}(q) \end{bmatrix} \tag{26}$$

$$P\left(q, \frac{\partial q}{\partial t}\right) = \begin{bmatrix} P_1\left(q, \frac{\partial q}{\partial t}\right) \\ P_2\left(q, \frac{\partial q}{\partial t}\right) \end{bmatrix} \tag{27}$$

$$M_{11}(q) = J_{eq} + M_p r^2 \cos^2(q_1) \tag{28}$$

$$M_{12}(q) = -\frac{1}{2} M_p r l_p \cos(q_1)\cos(q_2) \tag{29}$$

$$M_{22}(q) = J_p + M_p l_p^2 \tag{30}$$

$$\begin{aligned} P_1\left(q, \tfrac{\partial q}{\partial t}\right) &= -2M_p r^2 \cos(q_1)\sin(q_1)\left(\tfrac{\partial q_1}{\partial t}\right)^2 \\ &+ \tfrac{1}{4} M_p r l_p \cos(q_1)\sin(q_2)\left(\tfrac{\partial q_2}{\partial t}\right)^2 \end{aligned} \tag{31}$$

$$\begin{aligned} P_2\left(q, \tfrac{\partial q}{\partial t}\right) &= \tfrac{1}{2} M_p r l_p \sin(q_1)\cos(q_2)\left(\tfrac{\partial q_1}{\partial t}\right)^2 \\ &+ M_p g r l_p \sin(q_2) \end{aligned} \tag{32}$$

- $M_p = 0.027$ mass of the pendulum
- $l_p = 0.153$ the length of the pendulum center to the pivot
- $L_p = 0.191$ total length of pendulum
- $r = 0.0826$ the length of arm pivot to pendulum pivot
- $g = 9.81$ the gravitational acceleration constant
- $J_P = 1.23 \times 10^{-4}$ the pendulum moment of inertia about its pivot axis
- $Mag = 10$ analog control conversion gain (in Newton/volt)

This model consists of four state variables:

$q_1(t)$ angle of the driving arm w.r.t the horizontal position;
$\frac{\partial q_1(t)}{\partial t}$ angular velocity of the driving arm;
$q_2(t)$ angle of the pole w.r.t the vertical position;
$\frac{\partial q_2(t)}{\partial t}$ angular velocity of the pole;

In the simulation, a run consists of a maximum of 1000 consecutive trials. Note that only angle of the driving arm and angle of the pole are observable whereas angular velocity of the driving arm and angular velocity of the pole are not. That is to say $q_1(t)$ and $q_2(t)$ constitute the observable layer.

TABLE I
Summary of parameters used in simulation

Parameters	$l_c(0)$	$l_a(0)$	$l_c(f)$	$l_a(f)$	*
Value	0.6	0.3	0.005	0.001	*
Parameters	N_c	N_a	T_c	T_a	N_h
Value	50	100	0.05	0.005	6

A run is considered successful if the last trial (trial number less than 1000 in this case) of the run lasted 100,000 time steps. Otherwise, the run is considered unsuccessful. All the results are based on 20 random runs with zero initial states and random initial weight matrices. In our simulation, the time step was set 0.02s; a pole is considered failure once the pole is outside the range of $[-12°, 12°]$ or the driving arm is beyond the range of $[-170°, 170°]$ in reference to the central position. The control u generated by the action network is converted into force by an analog amplifier through a conversion gain Mag = 10 (in Newton/volt). Other parameters are shown shown in Table I.

B. Result Analysis

The network's parameters are demonstrated as follows.

$l_c(0)$ initial learning rate of the critic network
$l_a(0)$ initial learning rate of the action network
$l_c(t)$ learning rate of the critic network at time t which is decreased by 0.05 every 5 time steps until it reaches $l_c(f)$ and stays at $l_c(f)$ thereafter
$l_a(t)$ learning rate of the action network at time t which is decreased by 0.05 every 5 time steps until it reaches $l_a(f)$ and stays at $l_a(f)$ thereafter
N_c internal backpropagation cycle of the critic network
N_a internal backpropagation cycle of the action network
T_c internal training error threshold of the critic network
T_a internal training error threshold of the action network
N_h total number of hidden nodes

The details of these parameters are summarized in Table I. To be more applicable, we also conducted experiments of additive noise both on sensor and actor, i.e., to the state measurements and the action network output. Identical to the noise level setting in [9], we implemented the actor noise through $u = u \times (1 + \ noise \ \ percentage)$. For the sensor noise, the experiments were carried on with adding both uniform and Gaussian random variables to the angle measurement. The uniform state sensor noise was implemented through $\theta = \theta \times (1 + \ noise \ \ percentage)$. Gaussian sensor noise was zero mean with specified variance.

The results under different noise levels are shown in Table II. Fig. 3 and Fig. 4 show the trajectory and histogram of pendulum angle in a typical run under noise free situation, respectively. Fig. 5 shows the performance of the pole angular velocity (note that this variable is not available to actor and

TABLE II
SUMMARY OF PARAMETERS USED IN SIMULATION

Noise type	Result
Noise free	40.5
Uniform 10% actor	46.2
Uniform 10% sensor	44.1
Gaussian $\sigma^2 = 0.1$ sensor	59.3
Gaussian $\sigma^2 = 0.2$ sensor	76.4

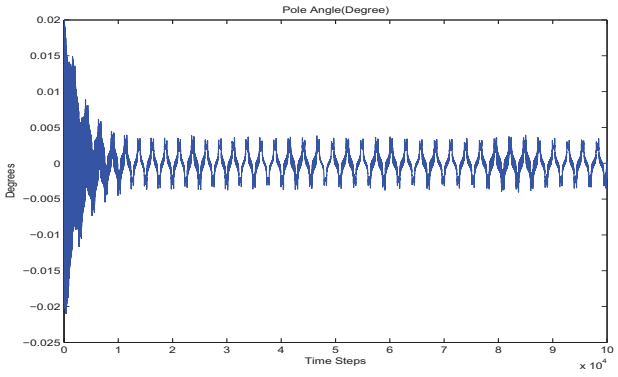

Fig. 3. Pole angle trajectory of the pendulum

critic). From Fig. 5 we can see that the actor-critic not only balance the pole angle, but can also stabilize the hidden state $\frac{\partial q_2(t)}{\partial t}$ as well. This is within our expectation in that the pole angular velocity converges since the pole is balanced in the steady state. From these results one can see, the proposed framework is able to fulfill the balancing task of Furuta pendulum. This indicates that the proposed approach may provide important theoretical suggestions of the ADP research as well as powerful practical techniques. Preliminary results are promising and effectively validate the framework proposed in this paper under POMDP environment.

IV. CONCLUSION

In this paper, we develop the idea originated in [4] into an online action-dependent gradient-based actor-critic paradigm

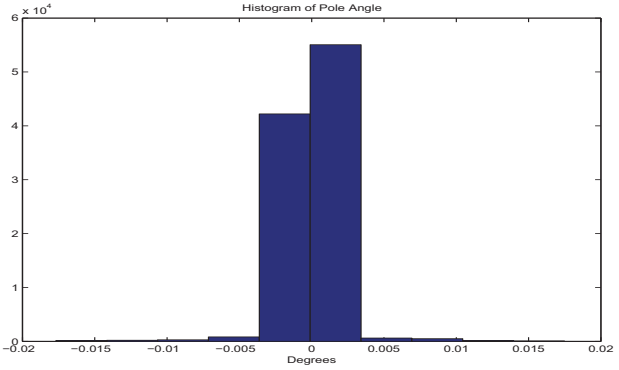

Fig. 4. Histogram of pole angle trajectory of the pendulum

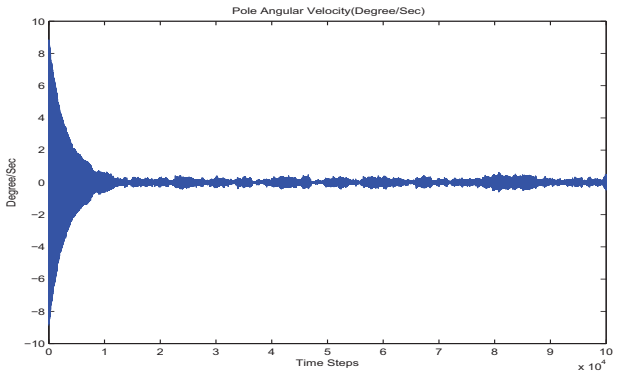

Fig. 5. Pole angular velocity trajectory of the pendulum

[9]. This is an important topic since it explores the ability of the adaptive dynamic programming approach in POMDP environment without building up state estimators to output belief states. Both theoretical research and detailed simulation results based on the Furuta pendulum balancing task are used to demonstrate the effectiveness of this approach.

REFERENCES

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[2] E. Alpaydin, *Introduction to Machine Learning*. MIT Press, October 2004.

[3] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement learning algorithm for partially observable markov decision problems," in *Advances in Neural Information Processing Systems 7*. MIT Press, 1995, pp. 345–352.

[4] S. P. Singh, T. Jaakkola, and M. I. Jordan, "Learning without state-estimation in partially observable markovian decision processes," in *In Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 1994, pp. 284–292.

[5] S. D. Whitehead and L. J. Lin, "Reinforcement learning in non-markov environments," *Artificial Intelligence*, vol. 8, pp. 3–4, 1992.

[6] L. ji Lin and T. M. Mitchell, "Memory approaches to reinforcement learning in non-markovian domains," Tech. Rep., 1992.

[7] M. Hauskrecht, "Value-function approximations for partially observable markov decision processes," *Journal of Artificial Intelligence Research*, vol. 13, pp. 33–94, 2000.

[8] K. P. Murphy, "A Survey of POMDP Solution Techniques," Tech. Rep., 2000.

[9] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 264–276, Mar 2001.

[10] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," *Neural Computation*, vol. 6, pp. 1185–1201, 1994.

[11] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, Sep 1986.

[12] K. Miroslav Krstic, Petar V. Kokotovic, *Nonlinear and Adaptive Control Design*. Wiley-Interscience, 1995.

[13] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *SIAM Journal on Control and Optimization*. MIT Press, 2000, pp. 1008–1014.

[14] E. Eweda and O. Macchi, "Convergence of an adaptive linear estimation algorithm," *IEEE Transactions on Automatic Control*, vol. 29, no. 2, pp. 119–127, Feb 1984.

[15] V. R. Konda and J. N. Tsitsiklis, "Linear stochastic approximation driven by slowly varying markov chains," *Systems and Control Letters*, vol. 50, pp. 95–102, 2003.

[16] L. Aguilar, I. Boiko, L. Fridman, and R. Iriarte, "Generating self-excited oscillations via two-relay controller," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 416–420, Feb. 2009.