Online Learning of Optimal Control Solutions Using Integral Reinforcement Learning and Neural Networks

[1]Kyriakos G. Vamvoudakis, [2]Draguna Vrabie and [1]Frank L. Lewis, *Fellow, IEEE*

[1]Automation and Robotics Research Institute,
University of Texas at Arlington,
Texas, USA
{kyriakos,lewis@uta.edu}

[2]United Technologies Research Center,
Connecticut, USA
vrabiedl@utrc.utc.com

***Abstract*— In this paper we introduce an online algorithm that uses integral reinforcement knowledge for learning the continuous-time optimal control solution for nonlinear systems with infinite horizon costs and partial knowledge of the system dynamics. This algorithm is a data based approach to the solution of the Hamilton-Jacobi-Bellman equation and it does not require explicit knowledge on the system's drift dynamics. The adaptive algorithm use the structure of policy iteration, and it is implemented on an actor/critic structure. Both actor and critic neural networks are adapted simultaneously and a persistence of excitation condition is required to guarantee convergence of the critic to the actual optimal value function. Novel tuning algorithms are given for both critic and actor networks, with extra terms in the actor tuning law being required to guarantee closed-loop dynamical stability. The convergence to the optimal controller is proven, and stability of the system is also guaranteed. Simulation example support the theoretical result.**

I. INTRODUCTION

AT the very top in the hierarchy of a complex integrated control applications stand the money-making loops that require cost effective control solutions. At this level the desired control solution is often associated with an optimal control problem. Such optimal control policies must satisfy the specified system performances while minimizing a structured cost index that describes the balance between desired performances and available control resources.

From a mathematical standpoint the solution of the optimal control problem is the solution of the underlying Hamilton-Jacobi-Bellman (HJB) equation. Until recently, due to the intractability of this nonlinear differential equation for continuous-time (CT) systems, which form the object of interest in this paper, only particular solutions were available (e.g. for the linear time-invariant case, the HJB becomes the Riccati equation). For this reason considerable effort has been devoted to developing algorithms which approximately solve this equation (e.g. [1], [3], [12]). Far more results are available for the solution of the discrete-time HJB equation. Good overviews are given in [4], [16].

In this paper we use Reinforcement Learning (RL) methods, specifically a new Integral Reinforcement Learning (IRL) approach, to provide an online learning solution to optimal control problem that does not require knowledge of the system drift dynamics.

The algorithm that we introduce herein is conceptually based on the Policy Iteration (PI) technique [8]. The PI algorithm is an iterative approach to solve the HJB equation by constructing a sequence of admissible control policies that converges to the optimal control solution. We will provide a characterization for an admissible policy. The algorithm starts by evaluating the cost of a given initial admissible control policy and then uses this information to obtain a new and improved control policy, i.e. the new policy that will have a lower associated cost compared to the previous control law. These two steps of policy evaluation and policy improvement are repeated until the policy improvement step no longer changes the actual policy, thus convergence to the optimal controller is achieved. One must note that the cost can be evaluated only in the case of admissible control policies, admissibility being a condition for the control policy which is used to initialize the algorithm.

Actor/critic structures based on Value Iteration have been introduced and further developed by Werbos [20], [21], [22] with the purpose of solving the optimal control problem online in real-time. Werbos defined four types of actor-critic algorithms based on value iteration, subsumed under the concept of Approximate or Adaptive Dynamic Programming (ADP) algorithms. Adaptive critics have been described in [14] for discrete-time systems and [2], [7], [18], [19] for continuous-time systems.

In the linear CT system case, when quadratic indices are considered for the optimal stabilization problem, the HJB equation becomes the well known Riccati equation and the policy iteration method is in fact Newton's method [9] which requires iterative solutions of Lyapunov equations. In the nonlinear systems case, successful application of the PI method was limited until [3], where Galerkin spectral

This work was supported by the National Science Foundation ECS-0801330, the Army Research Office W91NF-05-1-0314 and the Air Force Office of Scientific Research FA9550-09-1-0278.

approximation methods were used to solve the nonlinear Lyapunov equations describing the policy evaluation step in the PI algorithm. Such methods are known to be computationally intensive and cannot handle control constraints.

The key to solving practically the CT nonlinear Lyapunov equations was in the use of neural networks (NN) [1] which can be trained to become approximate solutions of these equations. In fact the PI algorithm for CT systems can be built on an actor/critic structure which involves two neural networks: one, the critic NN, is trained to become an approximation of the Lyapunov equation solution at the policy evaluation step, while the second one is trained to approximate an improving policy at the policy improving step.

Reinforcement learning (RL) is a class of methods used in machine learning to methodically modify the actions of an agent based on observed responses from its environment ([8], [17]). The RL methods have been developed starting from learning mechanisms observed in mammals. Every decision-making organism interacts with its environment and uses those interactions to improve its own actions in order to maximize the positive effect of its limited available resources; this in turn leads to better survival chances. RL is a means of *learning optimal behaviors by observing the response from the environment to non-optimal control policies*. In engineering terms, RL refers to the learning approach of an actor or agent which modifies its actions, or control policies, based on stimuli received in response to its interaction with its environment. This learning can be extended along two dimensions: i) nature of interaction (competitive or collaborative) and ii) the number of decision makers (single or multi agent).

Advances in RL for continuous-time systems have been hampered by the fact that the Bellman equation (Hamiltonian equation) for CT systems depends on the full system dynamics. In [19] was developed an online PI algorithm for continuous-time systems which converges to the optimal control solution without making explicit use of any knowledge on the internal dynamics of the system. The algorithm was based on the idea of Integral Reinforcement Learning (IRL), which allows the development of a Bellman equation that does not contain the system dynamics. That algorithm used *sequential updates* of the critic (policy evaluation) and actor (policy improvement) neural networks (i.e. while one is tuned the other one remains constant).

This paper is concerned with developing approximate online solutions, based on the structure of PI, for the infinite horizon optimal control problem for continuous-time (CT) nonlinear systems. We present an online integral reinforcement algorithm that combines the advantages of [18] and [19]. These include *simultaneous tuning* of both actor and critic neural networks [18] (i.e. both neural networks are tuned at the same time) and no need for the drift term in the dynamics [19]. Simultaneous tuning idea was first introduced by [20], [21] and has been the idea of recent papers in the area, however in most of these papers the authors either designed model-based controllers [6], [18] or used dynamic neural networks to identify the nonlinear plant [5]. Our algorithm avoids partial knowledge of the plant and uses only two neural networks by designing a hybrid controller as in [19].

The contributions in this paper are i) provide a new online continuous time algorithm that converge to the solution of HJB and Bellman equation without solving them, ii) partial need of dynamics, iii) update actor and critic neural networks simultaneously in real time and iv) only two approximators (neural networks) are used.

The paper is organized as follows. Section II provides the formulation of the optimal control problem followed by the general description of neural network value function approximation (VFA). Section III introduces the online synchronous integral reinforcement learning algorithm for the actor and critic networks based on PI. Results for convergence and stability are given. Section IV presents a simulation example that show the effectiveness of the online integral reinforcement learning algorithm.

II. The Optimal Control Problem and the Policy Iteration Algorithm

A. Optimal control and the continuous-time HJB equation

Let the system dynamics be described by the differential equation

$$\dot{x}(t) = f(x(t)) + g(x(t))\,u(x(t))\,;\, x(0) = x_0 \qquad (1)$$

with state $x(t) \in \mathbb{R}^n$, $f(x(t)) \in \mathbb{R}^n$, $g(x(t)) \in \mathbb{R}^{n\times m}$ and control input $u(t) \in U \subset \mathbb{R}^m$. We assume that $f(0) = 0$, $g(0) = 0$, $f(x) + g(x)u$ is Lipschitz continuous on a set $\Omega \subseteq \mathbb{R}^n$ that contains the origin. We assume that the dynamical system is stabilizable on Ω, i.e. there exists a continuous control function $u(t) \in U$ such that the system is asymptotically stable on Ω, and that $f(x) + g(x)u$ is Lipschitz continuous on Ω.

Define the infinite horizon integral cost

$$V(x_0) = \int_0^\infty r(x(\tau), u(\tau))\,d\tau \qquad (2)$$

where $r(x,u) = Q(x) + u^T R u$ with $Q(x)$ positive definite, *i.e.* $\forall x \neq 0, Q(x) > 0$ and $x = 0 \Rightarrow Q(x) = 0$, and $R \in \mathbb{R}^{m\times m}$ a positive definite matrix.

Definition 1. [1] (Admissible policy) A control policy $\mu(x)$ is defined as admissible with respect to (2) on Ω, denoted by $\mu \in \Psi(\Omega)$, if $\mu(x)$ is continuous on Ω, $\mu(0) = 0$, $\mu(x)$ stabilizes (1) on Ω and $V(x_0)$ is finite $\forall x_0 \in \Omega$.

For any admissible control policy $\mu \in \Psi(\Omega)$, if the associated cost function

$$V^\mu(x_0) = \int_0^\infty r(x(\tau), \mu(x(\tau)))\,d\tau \qquad (3)$$

is C^1, then an infinitesimal version of (3)is

$$0 = r(x, \mu(x)) + V_x^{\mu T}(f(x) + g(x)\mu(x)),\ V^{\mu}(0) = 0 \quad (4)$$

where V_x^{μ} denotes the partial derivative of the value function V^{μ} with respect to x. (Note that the value function does not depend explicitly on time). Equation (4) is a Lyapunov equation for nonlinear systems which, given a controller $\mu(x) \in \Psi(\Omega)$, can be solved for the value function $V^{\mu}(x)$ associated with it. Given that $\mu(x)$ is an admissible control policy, if $V^{\mu}(x)$ satisfies (4), with $r(x, \mu(x)) \geq 0$, then $V^{\mu}(x)$ is a Lyapunov function for the system (1) with control policy $\mu(x)$.

The optimal control problem can now be formulated: Given the continuous-time system (1), the set $\mu \in \Psi(\Omega)$ of admissible control policies and the infinite horizon cost functional (2), find an admissible control policy such that the cost index (2) associated with the system (1) is minimized.

Define the Hamiltonian of the problem

$$H(x, u, \nabla V_x) = r(x(t), u(t)) + \nabla V_x^{T}(f(x(t)) + g(x(t))u(t)) \quad (5)$$

the optimal cost function $V^*(x)$ satisfies the HJB equation

$$0 = \min_{u \in \Psi(\Omega)} [H(x, u, \nabla V_x^*)] \quad (6)$$

where $\nabla V_x \equiv \frac{\partial V}{\partial x}$ is disabused here as a column vector.

Assuming that the minimum on the right hand side of (6) exists and is unique then the optimal control function for the given problem is

$$u^*(x) = -R^{-1} g^T(x) \nabla V_x^* \quad (7)$$

Inserting this optimal control policy in the Hamiltonian we obtain the formulation of the HJB equation in terms of V_x^*

$$0 = Q(x) + \nabla V_x^{*T} f(x) - \frac{1}{4} \nabla V_x^{*T} g(x) R^{-1} g^T(x) \nabla V_x^*,\ V^*(0) = 0 \quad (8)$$

This is a necessary and sufficient condition for the optimal value function [11]. For the linear system case, considering a quadratic cost functional, the equivalent of this HJB equation is the well known Riccati equation.

In order to find the optimal control solution for the problem one only needs to solve the HJB equation (8)for the value function and then substitute the solution in (7) to obtain the optimal control. However, solving the HJB equation is generally difficult as it is a nonlinear differential equation, quadratic in the cost function, which also requires complete knowledge of the system dynamics (*i.e.* the system dynamics described by the functions $f(x), g(x)$ need to be known). The next section provides the policy iteration algorithm and the value function approximation of the critic network.

B. Policy Iteration

Policy iteration (PI) is an iterative method of reinforcement learning [16] for solving (8), and consists of policy improvement based on on (7) and policy evaluation based on (4).

In the actor/critic structure the Critic and the Actor functions are approximated by neural networks, and the PI algorithm consists in tuning alternatively each of the two neural networks. The critic neural network is tuned to evaluate the performance of the current control policy.

Policy Iteration Algorithm:

Step 1. Given policies $\mu^{(i)}(x)$, solve for the value $V^{\mu^{(i)}}(x(t))$ using

$$\begin{aligned} 0 &= r(x, \mu^{(i)}(x)) + (\nabla V_x^{\mu^{(i)}})^T (f(x) + g(x)\mu^{(i)}(x)) \\ V^{\mu^{(i)}}(0) &= 0 \end{aligned} \quad (9)$$

Step 2. Update the control policy using

$$\mu^{(i+1)} = \arg\min_{u \in \Psi(\Omega)} [H(x, u, \nabla V_x^{(i)})] \quad (10)$$

which explicitly is

$$\mu^{(i+1)}(x) = -\tfrac{1}{2} R^{-1} g^T(x) \nabla V_x^{(i)} \quad (11)$$

To ensure convergence of the PI algorithm an initial admissible policy $\mu^{(0)}(x(t)) \in \Psi(\Omega)$ is required. It is in fact required by the desired completion of the first step in the policy iteration: i.e. finding a value associated with that initial policy (which needs to be admissible to have a finite value and for the nonlinear Lyapunov equation to have a solution). The algorithm then converges to the optimal control policy $\mu^* \in \Psi(\Omega)$ with corresponding cost $V^*(x)$. Proofs of convergence of the PI algorithm have been given in several references. See [1], [2], [3], [7], [8], [12], [18], and [19].

Policy iteration is a Newton method. In the linear time-invariant case, it reduces to the Kleinman algorithm [9]for solution of the Riccati equation, a familiar algorithm in control systems. Then, (9) become a Lyapunov equation.

A major problem with this formulation of PI for CT systems is that the full system dynamics must be known as both *f(x)* and *g(x)* appear in the Bellman equation (9).

C. Value function approximation (VFA)

A practical method for implementing PI for CT systems is presented in this section. This involves two aspects: value function approximation (VFA) and integral reinforcement learning (IRL). The critic NN is based on value function approximation (VFA). Thus, assume there exist weights W_1 such that the value $V(x)$ is approximated by a neural network as

$$V(x) = W_1^T \phi(x) + \varepsilon(x) \quad (12)$$

where $\phi(x) : \mathbb{R}^n \to \mathbb{R}^N$ is the activation functions vector, N the number of neurons in the hidden layer, and $\varepsilon(x)$ the NN approximation error. It is known that $\varepsilon(x)$ is bounded by a constant on a compact set. Select the activation functions to provide a *complete* basis set such that $V(x)$ and its derivative

$$\frac{\partial V}{\partial x} = \nabla \phi^T W_1 + \frac{\partial \varepsilon}{\partial x} \tag{13}$$

are uniformly approximated. According to the Weierstrass higher-order approximation theorem [1], such a basis exists if $V(x)$ is sufficiently smooth. This means that, as the number of hidden-layer neurons $N \to \infty$, the approximation error $\varepsilon \to 0$ uniformly.

D. Integral Reinforcement Learning

The PI algorithm given above requires full system dynamics, since both *f(x)* and *g(x)* appear in the Bellman equation (9). In order to find an equivalent formulation of the Bellman equation that does not involve the dynamics, we note that for any time t_0 and time interval *T* the value function (3) satisfies

$$V^{\mu}(x_{t_0}) = \int_{t_0-T}^{t_0} r(x(\tau), \mu(x(\tau)))d\tau + V^{\mu}(x_{t_0-T}) \tag{14}$$

In [19] it is shown that (14) and (9) are equivalent, i.e., they both have the same solution. Therefore, (14) can be seen as a Bellman equation for CT systems. Note that this form does not involve the system dynamics. We call this the integral reinforcement learning (IRL) form of the Bellman equation.

Therefore, by using a critic NN for VFA, the Bellman error based on (14) becomes [19]

$$\int_{t-T}^{t} \left(Q(x) + \mu^T R \mu\right) d\tau + W_1^T \phi(x(t)) - W_1^T \phi(x(t-T)) = \varepsilon_B \tag{15}$$

We define the integral reinforcement as

$$p = \int_{t-T}^{t} \left(Q(x) + \mu^T R \mu\right) d\tau \tag{16}$$

Now (15) can be written as

$$\varepsilon_B - p = W_1^T \Delta\phi(x(t)) \tag{17}$$

where $\Delta\phi(x(t)) \equiv \phi(x(t)) - \phi(x(t-T))$.

Under the Lipschitz assumption on the dynamics, this residual error is bounded on a compact set. Moreover, in [1] it has been shown that, under certain assumptions, as the number of hidden layer neurons $N \to \infty$, one has $\varepsilon_B \to 0$.

III. ONLINE INTEGRAL REINFORCEMENT LEARNING ALGORITHM WITH SYNCHRONOUS TUNING OF ACTOR AND CRITIC NEURAL NETWORKS

Standard PI algorithms for CT systems are offline methods that require complete knowledge on the system dynamics to obtain the solution (*i.e.* the functions $f(x), g(x)$ in (1) need to be known). In order to change the offline character of PI for CT systems, and thus make it consistent with online learning mechanisms in the mammal brain, we present an adaptive learning algorithm that uses simultaneous continuous-time tuning for the actor and critic neural networks and does not need the drift term $f(x)$ in the dynamics. We term this *online integral reinforcement learning algorithm*.

A. Critic NN and Bellman equation solution

The weights of the critic NN, W_1, which solve (15) are unknown. Then the output of the critic neural network is

$$\hat{V}(x) = \hat{W}_1^T \phi(x) \tag{18}$$

where $\hat{W}_1$ are the current known values of the critic NN weights. Recall that $\phi(x): \mathbb{R}^n \to \mathbb{R}^N$ is the activation functions vector, with *N* the number of neurons in the hidden layer. The approximate Bellman error is then

$$\int_{t-T}^{t} \left(Q(x) + u^T R u\right) d\tau + \hat{W}_1^T \phi(x(t)) - \hat{W}_1^T \phi(x(t-T)) = e_1 \tag{19}$$

which according to (16) can be written as

$$\hat{W}_1^T \Delta\phi(x(t)) = e_1 - p \tag{20}$$

It is desired to select $\hat{W}_1$ to minimize the squared residual error

$$E_1 = \tfrac{1}{2} e_1^T e_1 \tag{21}$$

Then $\hat{W}_1(t) \to W_1$. We select the tuning law for the critic weights as the normalized gradient descent algorithm

$$\dot{\hat{W}}_1 = -a_1 \frac{\Delta\phi(x(t))^T}{\left(1 + \Delta\phi(x(t))^T \Delta\phi(x(t))\right)^2} \left[\int_{t-T}^{t} \left(Q(x) + u^T R u\right) d\tau + \Delta\phi(x(t))^T \hat{W}_1\right] \tag{22}$$

Note that the data required in this tuning algorithm at each time are $\left(\Delta\phi(t), p(t)\right)$.

Define the critic weight estimation error $\tilde{W}_1 = W_1 - \hat{W}_1$ and substitute (15) in (22) and, with the notation $\bar{\Delta}\phi(t) = \Delta\phi(t) / (\Delta\phi(t)^{\mathrm{T}} \Delta\phi(t) + 1)$ and $m_s = 1 + \Delta\phi(t)^T \Delta\phi(t)$, we obtain the dynamics of the critic weight estimation error as

$$\dot{\tilde{W}}_1 = -a_1 \bar{\Delta}\phi(t) \bar{\Delta}\phi(t)^{\mathrm{T}} \tilde{W}_1 + a_1 \bar{\Delta}\phi(t) \frac{\varepsilon_B}{m_s} \tag{23}$$

Though it is traditional to use critic tuning algorithms of the form (22), it is not generally understood when convergence of the critic weights can be guaranteed. In this paper, we address this issue in a formal manner. To guarantee convergence of $\hat{W}_1$ to W_1, the next Persistence of Excitation (PE) assumption is required.

Note that:

$$\Delta\phi(x(t)) = \int_{t-T}^{t} \nabla\phi(x)\dot{x}d\tau = \int_{t-T}^{t} \nabla\phi(f + gu)\, d\tau \tag{24}$$

It is obvious to see from (20) that the regression vector $\bar{\Delta}\phi(t)$ must be persistently exciting to solve for $\hat{W}_1$ in a least squares sense.

Persistence of Excitation (PE) Assumption. Let the

signal $\bar{\Delta}\phi(t)$ be persistently exciting over the interval $[t-T,t]$, *i.e.* there exist constants $\beta_1 > 0$, $\beta_2 > 0$, $\mathrm{T} > 0$ such that, for all *t*,

$$\beta_1 \mathrm{I} \le S_0 \equiv \int_{t-T}^{t} \bar{\Delta}\phi(\tau)\bar{\Delta}\phi^{\mathrm{T}}(\tau)d\tau \le \beta_2 \mathrm{I} \tag{25}$$

Remark 1. Note that, as $N \to \infty$, $\varepsilon_B \to 0$ uniformly [1].

B. Action NN and online adaptive optimal control

The policy improvement step in PI is given by substituting (13)into (7)as

$$u(x) = -\tfrac{1}{2} R^{-1} g^T(x) \nabla\phi^T W_1 \tag{26}$$

with critic weights W_1 unknown. Therefore, define the control policy in the form of an action neural network which computes the control input in the structured form

$$u_2(x) = -\tfrac{1}{2} R^{-1} g^T(x) \nabla\phi^T \hat{W}_2 \tag{27}$$

where $\hat{W}_2$ denotes the current known values of the actor NN weights.

Based on (8) and (15), define the approximate HJB equation

$$\int_{t-\mathrm{T}}^{t} \left(-Q(x) - \frac{1}{4} W_1^T \bar{D}_1(x) W_1 + \varepsilon_{HJB}(x) \right) d\tau = W_1^T \Delta\phi(x(t)) \tag{28}$$

with the notation $\bar{D}_1(x) = \nabla\phi(x) g(x) R^{-1} g^T(x) \nabla\phi^T(x)$, where W_1 denotes the ideal unknown weights of the critic and actor neural networks which solve the HJB.

We now present the main Theorems, which provide the tuning laws for the actor and critic neural networks that guarantee convergence to the optimal controller along with closed-loop stability. The next notion of practical stability is needed.

Definition 2. [10] (UUB) A time signal $\zeta(t)$ is said to be uniformly ultimately bounded (UUB) if there exists a compact set $S \subset \mathbb{R}^n$ so that for all $\zeta(0) \in S$ there exists a bound *B* and a time $T(B,\zeta(0))$ such that $\|\zeta(t)\| \le B$ for all $t \ge t_0 + T$.

Theorem 1. Let tuning for the critic NN be provided by

$$\dot{\hat{W}}_1 = -a_1 \frac{\Delta\phi(x(t))^T}{\left(1 + \Delta\phi(x(t))^T \Delta\phi(x(t))\right)^2} \left(\Delta\phi(x(t))^T \hat{W}_1 + \int_{t-T}^{t} \left(Q(x) + \frac{1}{4} \hat{W}_2^T \bar{D}_1 \hat{W}_2 \right) d\tau \right) \tag{29}$$

where $\Delta\phi(x(t)) = \int_{t-T}^{t} \nabla\phi(f + g u_2)\, d\tau$ and assume that $\bar{\Delta}\phi(t)$ is persistently exciting (which means u_2 is persistently exciting). Let the actor NN be tuned as

$$\dot{\hat{W}}_2 = -a_2 \left(F_2 \hat{W}_2 - F_1 \Delta\phi(x(t))^T \hat{W}_1 \right) - \tfrac{1}{4} a_2 \bar{D}_1(x) \hat{W}_2 \frac{\Delta\phi(x(t))^T}{\left(1 + \Delta\phi(x(t))^T \Delta\phi(x(t))\right)^2} \hat{W}_1 \tag{30}$$

Then the closed-loop system state is UUB, the critic parameter error $\tilde{W}_1 = W_1 - \hat{W}_1$ and the actor parameter error $\tilde{W}_2 = W_1 - \hat{W}_2$ are UUB.

Proof:

The convergence proof is based on Lyapunov analysis. For space reasons we will present the details of this proof in a future paper.

■

Theorem 2. Optimal solution. Suppose the hypotheses of Theorem 1 hold. Then:

a. $H(\hat{u}, \hat{W}_1, x) \equiv \int_{t-T}^{t} \left(Q(x) + \hat{u}^T R \hat{u} - \varepsilon_{HJB} \right) d\tau + \hat{W}_1^T \phi(x(t)) - \hat{W}_1^T \phi(x(t-T))$ is UUB, where $\hat{u} = -\tfrac{1}{2} R^{-1} g^T(x) \nabla\phi^T(x) \hat{W}_1$.

That is, $\hat{W}_1$ converge to the approximate HJB solution.

b. $\hat{u}_2(x)$ converges to the optimal solution, where $\hat{u}_2 = -\tfrac{1}{2} R^{-1} g^T(x) \nabla\phi^T(x) \hat{W}_2$.

Proof:

For space reasons we will present the details of this proof in a future paper.

■

Remark 2. The positive tuning parameters F_1, F_2 are selected appropriately to ensure stability.

IV. Simulation Results

To support the new synchronous online integral reinforcement learning algorithm for CT systems, we offer a simulation example of a nonlinear system. We observe convergence to the actual optimal value function and control. In these simulations, exponentially decreasing noise is added to the control inputs to ensure PE until convergence is obtained.

Consider the following affine in control input nonlinear system, with a quadratic cost constructed as in [13]

$$\dot{x} = f(x) + g(x)u,\ x \in R^2$$

where

$$f(x) = \begin{bmatrix} -x_1 + x_2 \\ -x_1^3 - x_2 - \dfrac{x_1^2}{x_2} + 0.25 x_2 (\cos(2x_1 + x_1^3) + 2)^2 \end{bmatrix}$$

$$g(x) = \begin{bmatrix} 0 \\ \cos(2x_1 + x_1^3) + 2 \end{bmatrix},$$

One selects $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 1$ and $T = 0.01$.

The optimal value function is $V^*(x) = \frac{1}{4}x_1^4 + \frac{1}{2}x_2^2$

the optimal control signal is

$$u^*(x) = -\tfrac{1}{2}(\cos(2x_1 + x_1^3) + 2)x_2$$

One selects the critic NN vector activation function as

$$\phi(x) = [x_1^2 \quad x_2^2 \quad x_1^4 \quad x_2^4]$$

Figure 1 shows the critic parameters, denoted by

$$\hat{W}_1 = [W_{c1} \quad W_{c2} \quad W_{c3} \quad W_{c4}]^T$$

After the simulation by using the integral reinforcement learning algorithm we have

$$\hat{W}_1(t_f) = \hat{W}_2(t_f) = [0.0033 \quad 0.4967 \quad 0.2405 \quad 0.0153]^T$$

The actor NN is given by

$$\hat{u}_2(x) = -\frac{1}{2}\begin{bmatrix} 0 \\ \cos(2x_1 + x_1^3) + 2 \end{bmatrix}^T \begin{bmatrix} 2x_1 & 0 & 4x_1^3 & 0 \\ 0 & 2x_2 & 0 & 4x_2^3 \end{bmatrix} \hat{W}_2(t_f)$$

The evolution of the system states is presented in Figure 2. One can see that after 80s convergence of the NN weights in both critic and actor has occurred. This shows that the probing noise effectively guaranteed the PE condition.

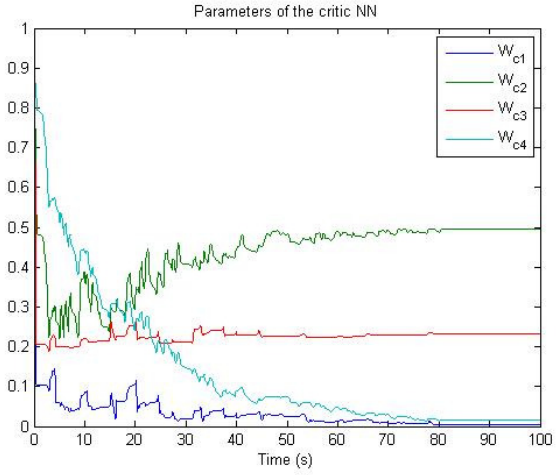

Fig. 1. Convergence of the critic parameters to the parameters of the optimal critic.

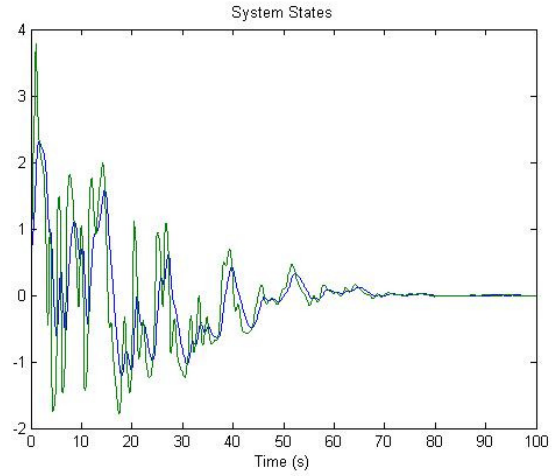

Fig. 4. Evolution of the system states for the duration of the experiment.

V. CONCLUSION

In this paper we have proposed a new adaptive algorithm which solves the continuous-time optimal control problem for affine in the inputs nonlinear systems. The importance of this algorithm relies on the partial need of dynamics, only $g(x)$ is needed, the simultaneous tuning of the actor and critic neural networks and the convergence to HJB and Bellman equation solution without solving these equations.

REFERENCES

[1] M. Abu-Khalaf, F. L. Lewis, "Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach" , *Automatica*, vol. 41, no. 5, pp. 779-791, 2005.

[2] L. C. Baird III, "Reinforcement Learning in Continuous Time: Advantage Updating", *Proc. Of ICNN*, Orlando FL, June 1994.

[3] R. Beard, G. Saridis, J. Wen, "Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation", *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

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

[5] S. Bhasin, M. Johnson, W. E. Dixon, "A model free robust policy iteration algorithm for optimal control of nonlinear systems", *Proc. 49th IEEE Conference on Decision and Control*, pp. 3060-3065, Atlanta, 2010.

[6] T. Dierks, S. Jagannathan, "Optimal Control of Affine Nonlinear Continuous-time systems Using an Online Hamilton-Jacobi-Isaacs Formulation", *Proc. 49th IEEE Conference on Decision and Control*, pp. 3048-3053, Atlanta, 2010.

[7] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-Time Adaptive Critics", *IEEE Transactions on Neural Networks*, 18(3), 631-647, 2007.

[8] R. A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, Massachusetts, 1960.

[9] D. Kleinman, "On an Iterative Technique for Riccati Equation Computations", *IEEE Trans. on Automatic Control*, vol. 13, pp. 114-115, February, 1968.

[10] F.L. Lewis, S. Jagannathan, A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis 1999.

[11] F. L. Lewis, V. L. Syrmos, *Optimal Control*, John Wiley, 1995.

[12] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive Dynamic Programming", *IEEE Trans. on Systems, Man and Cybernetics*, vol. 32, no. 2, pp 140-153, 2002.

[13] V. Nevistic, J. A. Primbs, "Constrained nonlinear optimal control: a converse HJB approach," *Technical Report 96-021*, California Institute of Technology, 1996.

[14] D. Prokhorov, D. Wunsch, "Adaptive critic designs," *IEEE Trans. on Neural Networks*, vol. 8, no 5, pp. 997-1007, 1997.

[15] B. Stevens, F. L. Lewis, *Aircract Control and Simulation,* 2nd edition, John Willey, New Jersey, 2003.

[16] J. Si, A. Barto, W. Powel, D. Wunch, *Handbook of Learning and Approximate Dynamic Programming*, John Wiley, New Jersey, 2004.

[17] R. S. Sutton, A. G. Barto, *Reinforcement Learning – An Introduction*, MIT Press, Cambridge, Massachusetts, 1998.

[18] Kyriakos G. Vamvoudakis, and F. L. Lewis, "Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem," *Automatica*, vol. 46, no. 5, pp. 878-888, 2010.

[19] D. Vrabie, O. Pastravanu, F. Lewis, M. Abu-Khalaf, "Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration," ,*Automatica*, vol.42, no. 2, pp. 477-484, 2009.

[20] P.J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences*, Ph.D. Thesis, 1974.

[21] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of Intelligent Control*, ed. D.A. White and D.A. Sofge, New York: Van Nostrand Reinhold, 1992.

[22] P. Werbos, "Neural networks for control and system identification", *IEEE Proc. CDC89,* IEEE, 1989.