Adaptive Optimal Control of Partially-unknown Constrained-input Systems using Policy Iteration with Experience Replay

Hamidreza Modares[1],
Ferdowsi University of Mashhad, Mashhad, Iran, 91775-1111

Frank L. Lewis[2],
University of Texas at Arlington, Arlington TX, USA, 7300

Mohammad-Bagher Naghibi-Sistani[3]
Ferdowsi University of Mashhad, Mashhad, Iran, 91775-1111

Girish Chowdhary[4]
Massachusetts Institute of Technology, Cambridge MA, USA, 02139-4307

and

Tansel Yucelen[5]
Missouri University of Science and Technology, Rolla, MO, USA 65409

This paper develops an online learning algorithm to find optimal control solutions for partially-unknown continuous-time systems subject to input constraints. The input constraints are encoded into the optimal control problem through a nonquadratic performance functional. An online policy iteration algorithm that uses integral reinforcement knowledge is developed to learn the solution to the optimal control problem online without knowing the full dynamics model. The policy iteration algorithm is implemented on an actor-critic structure, where two neural network approximators are tuned online and simultaneously to generate the optimal control law. A novel technique based on experience replay is introduced to retain past data in updating the neural network weights. This uses the recorded data concurrently with current data for adaptation of the critic neural network weights. Concurrent learning provides an easy-to-check real-time condition for persistence of excitation that is sufficient to guarantee convergence to a near optimal control law. Stability of the proposed feedback control law is shown and its performance is evaluated through simulations.

I. Introduction

Bellman's Principle of optimality has been widely used to design near-optimal controllers for both discrete-time and continuous-time systems, and it requires the solution of nonlinear and complicated Hamilton–Jacobi–Bellman (HJB) equations. Traditional methods for solving the HJB equation are offline and require complete knowledge of the system dynamics [1]. In practical applications, it is often desirable to design controllers conducive

[1] PhD-Student, Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.
[2] Professor, University of Texas at Arlington Research Institute, 7300 Jack Newell Blvd. S., Ft. Worth, TX 76118, USA.
[3] Assistant Professor, Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.
[4] Postdoctoral Associate, Laboratory of Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA.
[5] Assistant Professor, Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA.

to real time implementation and able to handle modeling uncertainties. Adaptive control theory consists of tools for designing stabilizing controllers which can adapt online to modeling uncertainty. However, classical adaptive control methods do not converge to the optimal feedback control solution, as they only minimize a norm of the output error. Indirect adaptive optimal controllers have been designed that first identify the system dynamics, and then solve optimal design equations.

Recently, reinforcement learning (RL) [2-4] as a learning methodology in machine learning has been used as a promising method to design of adaptive controllers that learn online the solutions to optimal control problems [1]. Considerable research has been conducted for approximating the HJB solution of discrete-time systems using RL algorithms. However, few results are available for continuous-time (CT) systems. Most of available RL algorithms for solving continuous-time optimal control problems are based on an iterative procedure, called policy iteration (PI) [5]. Using PI technique, the HJB nonlinear partial differential equation (PDE) is solved successively by breaking it into a sequence of linear PDEs that are considerably easier to solve. Beard [6] and Abu-Khalaf and Lewis [7] proposed iterative offline PI algorithms to solve the HJB equation. However, for real-time applications, online algorithms are often more desirable as can better handle sudden dynamic change, and do not require excessive offline data for training.

To overcome the limitations of offline solution for real-time applications, some online PI algorithms were presented [8-13]. However, none of these existing online PI algorithms take into account the input constraints due to actuator saturation. In practical control systems, the magnitude of the control signal is always bounded due to physical input saturation. Saturation is a common problem for actuators of control systems and ignoring this phenomenon often severely destroys system performance, or even may lead to instability [7]. Another problem related to the existing PI algorithm is that to ensure convergence of the critic to a near optimal value, a persistence of excitation (PE) condition is required to be satisfied which is which is often difficult or impossible to check. This occurs as a result of inefficiency of using the available data during learning, that is, existing online RL algorithms have high sample complexity [3, 4]. In particular it is well known that online policy iteration based algorithms, such as TD-λ are guaranteed to converge to an approximately optimal solution if and only if the markov chain induced by the closed loop system dynamics is guaranteed to revisit all states infinitely often (a condition known as ergodicity [14]). The ergodicity condition is closely related to that of persistency of excitation in traditional adaptive control.

Due to the requirements for PE-like conditions, existing PI algorithms are sample inefficient, that is, they require many samples from the real world in order to learn the optimal policy. In order to reduce sample complexity and use available data more effectively, experience replay technique [15-19] has been proposed in the context of RL. In this technique, a number of recent samples are stored in a database and they are presented repeatedly to the RL algorithm. However, there has been no result on how to use the experience replay technique to relax the PE condition in RL algorithms. In [20, 21], Chowdhary and Johnson introduced a related idea, called concurrent learning, for adaptive control of uncertain dynamical systems. They showed that the concurrent use of recorded and current data can lead to exponential stability of a model reference adaptive controller as long as the recorded data is sufficiently rich. They also showed that the richness of recorded data is guaranteed if it consists of as many linearly independent elements as the number of unknowns, this condition was termed the rank condition [20, 21]. However, their results were focused on direct adaptive control, and in particular, that work did not establish any optimality guarantees on the closed loop system. In this paper, we merge the ideas from concurrent learning adaptive control with the notion of experience replay in a policy-iteration based reinforcement learning framework to guarantee convergence to a near optimal control law subject also to the rank-condition. In that sense, not only does this paper contribute to the RL literature, as such guarantees are not available in existing experience replay literature [15-19], but it also contributes to adaptive control literature since direct adaptive optimal control has been argued to be equivalent to reinforcement learning [2].

In this paper we introduce the use of experience replay to the integral reinforcement learning (IRL) 0 approach and develop approximate online solutions for optimal control of CT systems in the presence of input constraints. Experience replay allows more efficient use of current and past data, and provides simplified conditions to check for PE-like requirements in real time. IRL allows applications to systems with unknown drift dynamics. A suitable nonquadratic functional is used to encode the input constraint into the optimization formulation. Then, an IRL algorithm is developed to solve the associated HJB equation online. The IRL allows development of a Bellman equation that does not contain the system dynamics. The optimal control law and optimal value function are approximated as the output of two NNs, namely an actor NN and a critic NN. To update the critic NN weights, the experience replay technique is employed. It is shown using the proof techniques from [20, 21], that using experience replay, or concurrent real-time learning, a simple and easily verifiable condition on the richness of the recorded data is sufficient to guarantee exponential convergence of the critic NN weights. The closed-loop stability of the overall system is assured.

II. Optimal Control of Constrained-input Systems

A. Constrained optimal control and policy iteration

In this section, the optimal control problem for affine-in-the-input nonlinear systems with input constraints is formulated and an offline PI algorithm is given for solving the related optimal control problem.

Consider the system dynamics be described by the differential equation

$$\dot{x} = f(x) + g(x)u(x) \tag{1}$$

where $x \in \Re^n$ is a measurable system state vector, $f(x) \in \Re^n$ is the drift dynamics of the system, $g(x) \in \Re^{n\times 1}$ is the input dynamics of the system, and $u(x) \in \Re$ is the control input. We denote $\Omega_u = \{ u \,|\, u \in \Re, |u(x)| \le \lambda \}$ as the set of all inputs satisfying the input constraints, where λ is the saturating bound for actuators. It is assumed that $f(x) + g(x)u$ is Lipchitz and that the system is stabilizable.

It is assumed that the drift dynamics $f(x)$ is unknown and $g(x)$ is known.

Define the performance index

$$V(x(t)) = \int_t^\infty \left(Q(x(\tau)) + U(u(\tau)) \right) d\tau \tag{2}$$

where $Q(x)$ is positive definite monotonically increasing function and $U(u)$ is a positive definite integrand function.

Assumption 1: The performance functional (2) satisfies zero-state observability.

Definition 1 (Admissible control) [6, 7]: A control policy $\mu(x)$ is said to be admissible with respect to (2) on Ω, defined by $\mu \in \pi(\Omega)$, if $\mu(x)$ is continuous on Ω, $\mu(0) = 0$, $u(x) = \mu(x)$ stabilizes system (1) on Ω, and $V(x_0)$ is finite $\forall x_0 \in \Omega$.

To deal with the input constraints, the following generalized nonquadratic functional can be used [7, 22].

$$U(u) = 2\int_0^u \left(\lambda \tanh^{-1}(v/\lambda) \right)^T R \, dv \tag{3}$$

Using the cost functional (3) in Eq. (2), the value function becomes

$$V(x(t)) = \int_t^\infty \left(Q(x) + 2\int_0^u \left(\lambda \tanh^{-1}(v/\lambda) \right)^T R \, dv \right) d\tau \tag{4}$$

Differentiating V along the system trajectories, the following Bellman equation is obtained

$$Q(x) + 2\int_0^u \left(\lambda \tanh^{-1}(v/\lambda) \right)^T R \, dv + \nabla V^T(x) \left(f(x) + g(x)\,u(x) \right) = 0 \tag{5}$$

where $\nabla V(x) = \partial V(x) / \partial x$. The optimal value function $V^*(x)$ satisfies the HJB equation [1]

$$\min_{u \in \pi(\Omega)} \left[Q(x) + 2\int_0^u \left(\lambda \tanh^{-1}(v/\lambda) \right)^T R \, dv + \nabla V^{*T}(x) \left(f(x) + g(x)\,u(x) \right) \right] = 0 \tag{6}$$

The optimal control for the given problem is obtained by differentiating Eq. (6) and is given as

$$u^*(x) = -\lambda \tanh\left(\frac{1}{2\lambda} R^{-1} g^T(x) \nabla V^*(x) \right) \tag{7}$$

Using Eq. (7) in Eq. (3), it yields

$$U\left(u^*\right)=\lambda\nabla V^{*T}\left(x\right)g\left(x\right)\tanh\left(D^*\right)+\lambda^2 R\ln\left(1-\tanh^2\left(D^*\right)\right) \tag{8}$$

where $D^*=\left(1/2\lambda\right)R^{-1}g\left(x\right)^T\nabla V^*\left(x\right)$. Substituting $u^*\left(x\right)$ (7) back into Eq. (5) and using $U\left(u^*\right)$(8), the following HJB equation is obtained

$$H\left(x,u^*,\nabla V^*\right)=Q\left(x\right)+\nabla V^{*T}\left(x\right)f\left(x\right)+\lambda^2 R\ln\left(1-\tanh^2\left(D^*\right)\right)=0 \tag{9}$$

In order to find the optimal control solution directly, first the HJB equation (9) must be solved for the optimal value function, then the optimal control input that achieves this minimal performance is obtained by Eq. (7). However, solving the HJB equation (9) requires solving a nonlinear PDE, which may be impossible to solve in practice.

Instead of directly solving the HJB equation, in [7] an iterative PI algorithm is presented. The PI algorithm starts with a given admissible control policy and then performs a sequence of two-step iterations to find the optimal control policy. In the policy improvement step, the Bellman equation (5) is used to find the value function for a given fixed policy and in the policy improvement step, using the value function found in the policy evaluation step, the algorithm finds an improved control policy of the form Eq. (7). However, to evaluate the value of a fixed policy using the Bellman equation (5), the complete knowledge of the system dynamics must be known a priori. In order to find an equivalent formulation of the Bellman equation in policy evaluation step that does not involve the dynamics, we use the integral reinforcement learning (IRL) idea as introduced in [11]. Note that for any time t and time interval $T>0$, the value function (4) satisfies

$$V\left(x_t\right)=\int_{t-T}^{t}\left(Q\left(x\right)+2\int_0^u\left(\lambda\tanh^{-1}\left(v/\lambda\right)\right)^T R\,dv\right)d\tau+V\left(x_{t-T}\right) \tag{10}$$

In [11], it is shown that Eq. (10) and Eq. (4) are equivalent and have the same solution. Therefore, Eq. (10) can be viewed as a Bellman equation for CT systems. Note that the IRL form of the Bellman equation does not involve the system dynamics. Using Eq. (10) instead of Eq. (5) to evaluate the value function, the following PI algorithm is obtained.

Algorithm 3.1: Integral Reinforcement Learning

1. (*policy evaluation)* given a control input $u^i\left(x\right)$, find $V^i\left(x\right)$ using the Bellman equation

$$V^i\left(x_t\right)=\int_{t-T}^{t}\left(Q\left(x\right)+2\int_0^{u^i}\left(\lambda\tanh^{-1}\left(v/\lambda\right)\right)^T R\,dv\right)d\tau+V^i\left(x_{t-T}\right) \tag{11}$$

2. (*policy improvement*) update the control policy using

$$u^{i+1}\left(x\right)=-\lambda\tanh\left(\frac{1}{2\lambda}R^{-1}g^T\left(x\right)\nabla V^i\left(x\right)\right) \tag{12}$$

The above PI algorithm only needs to have knowledge of the input dynamics. The online implementation of this PI algorithm is introduced in section III.

B. Value function approximation and the approximated HJB equation

In this subsection, we discuss the value function approximation to solve for the cost function $V(x)$ in policy evaluation (11). Assuming the value function is a smooth function, according to the Weierstrass high-order approximation Theorem [23], there exists a single-layer NN such that the solution $V\left(x\right)$ and its gradient can be uniformly approximated as

$$V(x)=W_1^T\phi\left(x\right)+\varepsilon_v\left(x\right) \tag{13}$$

$$\nabla V(x)=\nabla\phi^T\left(x\right)W_1+\nabla\varepsilon_v\left(x\right) \tag{14}$$

where $\phi\left(x\right)\in\Re^m$ provides a suitable basis function vector, $\varepsilon_v\left(x\right)$ is the approximation error, $W_1\in\Re^m$ is a constant parameter vector, l is the number of neurons.

Assumption 2 [9]. The NN reconstruction error and its gradient are bounded over a compact set. Also, the NN activation functions and their gradients are bounded.

Before presenting the actor and critic update laws, it is necessary to see the effect of the reconstruction error on the HJB equation. Assuming that the optimal value function is approximated by Eq. (13) and using its gradients Eq. (14) in the Bellman equation (10) it yields

$$\int_{t-T}^{t}\left(Q(x)+2\int_0^u\left(\lambda\tanh^{-1}(v/\lambda)\right)^T R\,dv\right)d\tau+W_1^T\,\Delta\phi(x(t))\equiv\varepsilon_B(t) \tag{15}$$

where

$$\Delta\phi(x(t))=\phi(x(t))-\phi(x(t-T)) \tag{16}$$

and $\varepsilon_B(t)$ is the Bellman approximation error and under Assumption 1 is bounded on the compact set Ω. Also, the optimal policy is obtained as

$$u=-\lambda\tanh\left(\frac{1}{2\lambda}R^{-1}g^T\left(\nabla\phi^T W_1+\nabla\varepsilon_v{}^T\right)\right) \tag{17}$$

Using Eq. (17) in Eq. (15), the following HJB equation is obtained.

$$\int_{t-T}^{t}\left(Q+W_1^T\nabla\phi\,f+\lambda^2R\ln\left(1-\tanh^2(D)\right)+\varepsilon_{HJB}\right)d\tau=0 \tag{18}$$

where $D=(1/2\lambda)R^{-1}g^T\nabla\phi^T W_1$, and ε_{HJB} is the residual error due to the function reconstruction error. In [7], the authors show that as for each constant vector ε_h, we can construct a NN so that $\sup_{\forall x}\|\varepsilon_{HJB}\|\le\varepsilon_h$. Note that in Eq. (18) and in the sequel, the variable x is dropped for ease of exposition.

III. Online Intergal Reinforcement Learning to Solve the Constrained Optimal Control Problem

An online IRL algorithm based on Policy Iteration (PI) algorithm is now given. The learning structure uses two NNs, i.e., an actor NN and a critic NN, which approximate the Bellman equation and its corresponding policy. The offline PI Algorithm 3.1 is used to motivate the structure of this online PI algorithm. Instead of sequentially updating the critic and actor NNs, as in Algorithms 3.1, both are updated simultaneously in real-time. We call this synchronized online PI. This is the continuous version of Generalized Policy Iteration (GPI) introduced in [2].

A. Critic NN and tuning using experience replay

This subsection presents tuning and convergence of the critic NN weights for a fixed admissible control policy, in effect designing an observer for the unknown value function for using in feedback.

Consider a fixed admissible control policy $u(x)$ and assume that its corresponding value function is approximated by Eq. (13). Then, the Bellman equation (15) can be used to find the value function related to this control policy. However, the ideal weights of the critic NN, i.e. W_1, which provide the best approximation solution for Eq. (15) are unknown and must be approximated in real-time. Hence, the output of the critic NN can be written as

$$\hat{V}(x)=\hat{W}_1^T\phi \tag{19}$$

where the weights $\hat{W}_1$ are the current estimated values of W_1 and then the approximate Bellman equation becomes

$$\int_{t-T}^{t}\left(Q(x)+2\int_0^u\left(\lambda\tanh^{-1}(v/\lambda)\right)^T R\,dv\right)d\tau+\hat{W}_1^T\Delta\phi(x(t))=e(t) \tag{20}$$

Equation (20) can be written as

$$e(t)=\hat{W}_1(t)^T\,\Delta\phi(t)+p(t) \tag{21}$$

where

$$p(t)=\int_{t-T}^{t}\left(Q(x)+2\int_0^u\left(\lambda\tanh^{-1}(v/\lambda)\right)^T R\,dv\right)d\tau \tag{22}$$

Note that the Bellman error e in Eqs. (20) and (21) is the continuous-time counterpart of the Temporal Difference (TD) [2]. The problem of finding the value function is now converted to adjusting the parameters of the critic NN such that the TD e is minimized.

In the following, a real-time learning algorithm based on the experience replay technique is applied for updating the critic NN weights. In contrast to traditional learning algorithms, in which only instantaneous Bellman equation error is used to update the critic weights, recorded data are used concurrently with current data for adaptation of the critic NN weights. Using this learning law, a simple condition on the richness of the recorded data is sufficient to guarantee exponential parameter estimation errors convergence.

The proposed experience replay-based update rule for the critic NN weights stores recent transition samples and repeatedly presents them to the gradient-based update rule. In can be interpreted as a gradient-descent algorithm that not only tries to minimize the instantaneous Bellman error, but also the Bellman equation error for the stored transition samples obtained by the current critic NN weights. These samples are stored in a history stack. To collect a history stack, let t_j , $j=1,...,l.$ denote some recorded times during learning. Let

$$\Delta\phi_j=\Delta\phi(t_j)=\phi(x(t_j))-\phi(x(t_j-T)) \tag{23}$$

and

$$p_j=p(t_j)=\int_{t_j-T}^{t_j}\left(Q(x)+2\int_0^u\left(\lambda\tanh^{-1}(v/\lambda)\right)^T R\,dv\right)d\tau \tag{24}$$

denote $\Delta\phi(t)$ and $p(t)$ evaluated at time t_j , $j=1,...,l$ and

$$e_j=\hat{W}_1(t)\Delta\phi_j+p_j \tag{25}$$

is the Bellman equation error at time t_j using the current critic NN weights. Note that using Eqs. (15), (21) and (25) we have

$$e_j=\tilde{W}_1(t)\Delta\phi_j+\varepsilon_B(t_j) \tag{26}$$

$$e(t)=\tilde{W}_1(t)\Delta\phi(t)+\varepsilon_B(t) \tag{27}$$

where $\varepsilon_B(t_j)$ is the reconstruction error obtained by Eq. (15) in time t_j and $\tilde{W}_1=W_1-\hat{W}_1$. The proposed learning gradient descent algorithm for the critic NN is now given as

$$\dot{\hat{W}}_1(t)=-\alpha_1\frac{\Delta\phi(t)}{\left(1+\Delta\phi(t)^T\Delta\phi(t)\right)^2}\left(p(t)+\Delta\phi(t)^T\hat{W}_1(t)\right)-\alpha_1\sum_{j=1}^{l}\frac{\Delta\phi_j}{\left(1+\Delta\phi_j{}^T\Delta\phi_j\right)^2}\left(p_j+\Delta\phi_j^T\hat{W}_1(t)\right) \tag{28}$$

Remark 2. Note that in this experience replay tuning law the last term depends on the history stack of previous activation function differences. Furthermore, note that the updates based on both current and recorded data use the current estimate of the weights

Using Eqs. (26), (27) and (28), and notations $\Delta\bar{\phi}(t)=\Delta\phi(t)/\left(1+\Delta\phi(t)^T\Delta\phi(t)\right)$ and $m_s=1+\Delta\phi(t)^T\Delta\phi(t)$, the critic NN weights error dynamics becomes

$$\dot{\tilde{W}}_1(t)=-\alpha_1\left[\Delta\bar{\phi}(t)\Delta\bar{\phi}(t)^T+\sum_{j=1}^{l}\Delta\bar{\phi}_j\Delta\bar{\phi}_j{}^T\right]\tilde{W}_1(t)+\alpha_1\left[\Delta\bar{\phi}(t)\frac{\varepsilon_B(t)}{m_s}+\sum_{j=1}^{l}\Delta\bar{\phi}_j\frac{\varepsilon_B(t_j)}{m_s}\right] \tag{29}$$

Condition 1. Let $Z=\left[\Delta\bar{\phi}_1,...,\Delta\bar{\phi}_l\right]$ be the history-stack. Then, Z in the recorded data contains as many linearly independent elements $\Delta\bar{\phi}\in\Re^{m\times 1}$ as the dimension of the basis of the uncertainty. That is $rank(Z)=m$.

Remark 3: Condition 1 is related to PE condition. However, instead of a traditional PE condition, which is often difficult or impossible to check online, this condition is easy to check online [20, 21].

It was shown in [20, 21] that the rank-condition in Condition 1 and a concurrent learning NN weight update law that uses current and recorded data concurrently (similar to Eq.(29)) leads to exponential stability of the closed loop system in the Model Reference Adaptive Control framework. In the following, we show that an experience replay enabled NN weight update law can lead to exponential weight convergence in PI algorithm provided that Conditions 1 is satisfied. The results presented in this paper make a significant contribution to RL literature, because existing work in experience replay [15-19] does not provide such guarantees nor uses the notion of a rank condition on the history-stack.

Theorem 1: Let the online critic tuning law is given by the weight update law of Eq. (28). If the recorded data points satisfy Condition 1, then

(a) for $\varepsilon_B(t)=0, \forall t, \hat{W}_1$ converges exponentially to W_1.

(b): For bounded $\varepsilon_B(t)$, i.e. $\sup_{\forall x}\|\varepsilon_B(t)\| \le \varepsilon_{\max}\ \forall t$, $\tilde{W}_1$ converges exponentially to a residual set.

Proof: (a): Consider a Lyapunov function as

$$V = \frac{1}{2}\tilde{W}_1(t)^T \alpha_1^{-1}\tilde{W}_1(t) \tag{30}$$

Differentiating Eq. (30) along the trajectories of Eq. (28), and considering $\varepsilon_B(t)=0$, we have

$$\dot{V} = -\tilde{W}_1(t)^T\left[\Delta\bar{\phi}(t)\Delta\bar{\phi}(t)^T + \sum_{j=1}^{l}\Delta\bar{\phi}_j\,\Delta\bar{\phi}_j^T\right]\tilde{W}_1(t) \tag{31}$$

If the condition 1 is satisfied, then $\left[\sum_{j=1}^{l}\Delta\bar{\phi}_j\,\Delta\bar{\phi}_j^T\right] > 0$ and hence $\dot{V} < 0$. This concludes that $\tilde{W}_1(t)$ converge to zero exponentially fast.

(b): Viewing Eq. (29) as a linear time-varying system, the solution $\tilde{W}_1$ is given by

$$\tilde{W}_1(t) = \boldsymbol{\varphi}(t,t_0)\tilde{W}_1(0) + \alpha_1\int_{t_0}^{t}\boldsymbol{\varphi}(\tau,t_0)\,\varepsilon_{GB}\,d\tau \tag{32}$$

where $\varepsilon_{GB} = \Delta\bar{\phi}(t)\dfrac{\varepsilon_B(t)}{m_s} + \sum_{j=1}^{l}\Delta\bar{\phi}_j\,\dfrac{\varepsilon_B(t_j)}{m_s}$ with the state transition matrix defined as

$$\frac{\partial\boldsymbol{\varphi}(t,t_0)}{\partial t} = -\alpha_1\left[\Delta\bar{\phi}\,\Delta\bar{\phi}(t)^T + \sum_{j=1}^{l}\Delta\bar{\phi}_j\,\Delta\bar{\phi}_j^T\right]\boldsymbol{\varphi}(t,t_0) \tag{33}$$

From proof of part (a), it can be concluded that $\boldsymbol{\varphi}(t,t_0)$ is exponentially stable provided that condition 1 is satisfied. Therefore, if condition 1 is satisfied, we obtain

$$\|\tilde{W}_1(t)\| \le \eta_1 e^{-\eta_2 t} + \alpha_1\int_0^t e^{-\eta_2(t-\tau)}\|\varepsilon_{GB}\|\,d\tau \tag{34}$$

for some $\eta_1, \eta_2 > 0$. Since $\sup_{\forall x}\|\varepsilon_B(t)\| \le \varepsilon_{\max}$ and $\left(\Delta\bar{\phi}(t)/m_s\right) < 1$, Eq. (34) can be written as

$$\|\tilde{W}_1\| \le \eta_1 e^{-\eta_2 t} + \frac{\alpha_1(l+1)}{\eta_2}\varepsilon_{\max} \tag{35}$$

The first term of the above equation converges to zero exponentially fast and this completes the proof of (b).

B. Tuning of the actor NN and the online PI algorithm

This section presents our main results. To solve the optimal control problem adaptively, an online PI algorithm is given which involves simultaneous and synchronous tuning of the actor and critic NNs.

As mentioned, in policy improvement step (12) of Algorithm 3.1, the actor finds an improved control policy according to the current estimated value function. Assume that $\hat{W}_1$ are the current estimation for the optimal critic NN weights. Then according to Eq. (12) one can update the control policy as

$$u_1 = -\lambda \tanh\left(\frac{1}{2\lambda} R^{-1} g^T \nabla\phi^T \hat{W}_1 \right) \tag{36}$$

However, the above standard policy improvement does not guarantee the stability of the system. Therefore, to assure stability, the following nonstandard policy update law is used

$$\hat{u}_1 = -\lambda \tanh\left(\frac{1}{2\lambda} R^{-1} g^T \nabla\phi^T \hat{W}_2 \right) \tag{37}$$

where $\hat{W}_2$ are considered as the current estimated values of the unknown optimal weights W_1.

Define the actor NN estimation errors as

$$\tilde{W}_2 = W_1 - \hat{W}_2 \tag{38}$$

Assumption 3. $g(x)$ is bounded by a constant.

We now present the main theorem which provides the tuning laws for the actor and critic NNs that assure convergence of the proposed PI algorithm to a near optimal control law, while guaranteeing stability. Define

$$\hat{U} = \hat{W}_2^T \nabla\sigma(\hat{\psi}\varsigma)\ \lambda\ \tanh\left(\hat{D}\right) + \lambda^2 R \ln\left(1 - \tanh^2\left(\hat{D}\right)\right)$$

$$\hat{D} = (1/2\lambda) R^{-1} (\hat{\psi}\varsigma)^T \nabla\phi^T \hat{W}_2$$

Theorem 2: Given the dynamical system (1) . Let the tuning for the critic NN be provided by

$$\dot{\hat{W}}_1 = -\alpha_1 \frac{\Delta\phi}{\left(1 + \Delta\phi^T \Delta\phi\right)^2} \left(\int_{t-T}^{t} \left(Q + \hat{U}\right) d\tau + \Delta\phi^T \hat{W}_1 \right) - \alpha_1 \sum_{j=1}^{l} \frac{\Delta\phi_j}{\left(1 + \Delta\phi_j^T \Delta\phi_j\right)^2} \left(\int_{t_j-T}^{t_j} \left(Q + \hat{U}\right) d\tau + \Delta\phi_j^T \hat{W}_1 \right) \tag{39}$$

Let Condition 1 be satisfied. Let the actor NN be tuned as

$$\dot{\hat{W}}_2(t) = -\alpha_2 \left[Y_1 \hat{W}_2 + \nabla\phi\ g\ \lambda\ \tanh\left(\hat{D}\right) + M_2 \frac{\Delta\bar{\phi}^T}{m} \hat{W}_1 + a\ M_2 \int_{t-T}^{t} M_2^T \hat{W}_2\, d\tau \right] \tag{40}$$

where $\Delta\bar{\phi} = \Delta\phi \big/ \left(1 + \Delta\phi^T \Delta\phi\right)$, $m = 1 + \Delta\phi^T \Delta\phi$ and Y_1 is a design parameter. Let Assumptions 1-3 hold. Let the sampling time be small enough. Let the control law be given by Eq. (37). Then the closed-loop system states, the critic NN error, and the actor NN error are UUB, for sufficiently large number of NN neurons provided that

$$Y_1 > \max_{\forall t} 0.5(1+a)\|M_2\|^2 \tag{41}$$

$$T < \sqrt[3]{\frac{3}{m_x^2}} \tag{42}$$

Proof: Proof is not provided due to the page limitation.

IV. Simulation Results

Consider the following nonlinear dynamics for the system (1).

$$f = \left(x_1 + x_2 - x_1\left(x_1^2 + x_1^2\right), -x_1 + x_2 - x_2\left(x_1^2 + x_1^2\right)\right)^T \tag{43}$$

$$g = [0, 1]^T \tag{44}$$

The aim is to control the system with control limits of $|u| \le 1$. The above system was previously employed by [7] to test their offline optimal control design algorithm.

The cost function is chosen as the nonquadratic cost function (4) with $Q(x) = x_1^2 + x_2^2$ and R=1 [7]. Also, similar to [7], the critic NN is chosen as a power series neural network with 24 activation functions containing powers of the state variable of the system up to the eight order. The critic and actor weights are initialized randomly. To perform simulations, the integral reinforcement interval T is considered as 0.05. During learning, the probing noise is added to the control input to ensure that Condition 1 is satisfied. Fig 1 shows the states of the system during online learning. The probing noise is no longer required and is thus removed after 150 seconds. After that, the states remain very close to zero, as required. Fig 2 shows convergence of the first 10 weights of the critic NN. In fact, the critic weights converge to W = [8.86, 4.60, 3.62, -2.80, 2.39, -1.15, 1.05, 2.89, -5.64, -0.54, 1.64, 1.60, 3.52, 2.67, 5.29, 0.29, 1.52, -0.22, -1.72, -1.00, 2.78, -0.72, 3.35, 2.22].

The performance of the final near-optimal controller which is found at the end of the learning process (control law 1) is compared with the performance of the near-optimal control law found in [7] (control law 2). Figs 3 and 4 depict the state x_1 and the control effort for control laws 1 and, starting the system from a specific initial condition. Comparing the results, it is obvious that the performance of the proposed optimal control law is better than those of [7], as both the control effort and the states for the proposed control law converges to zero faster.

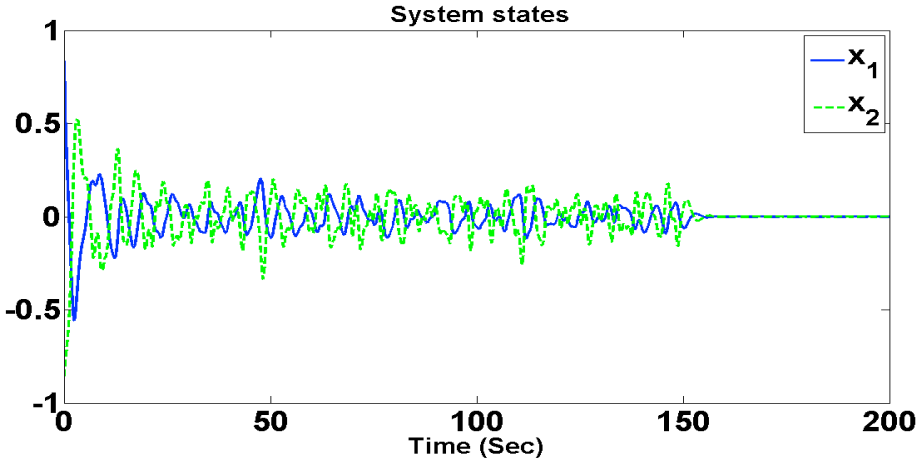

Fig 1. State trajectory during online learning

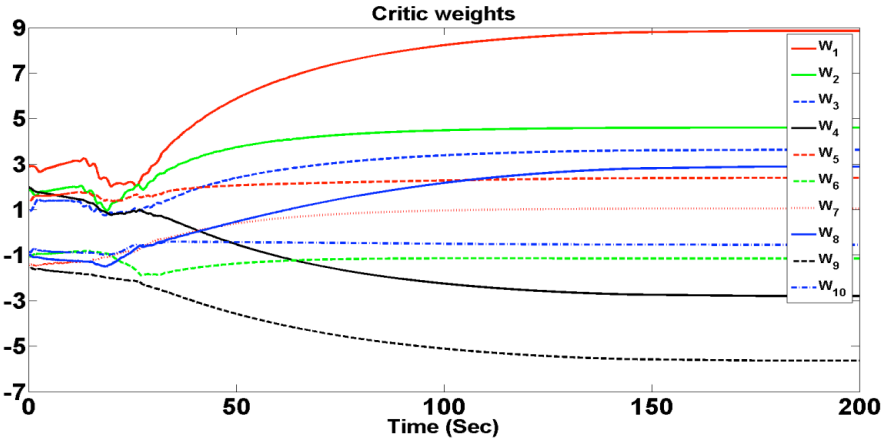

Fig 2. Convergence of the critic NN parameters

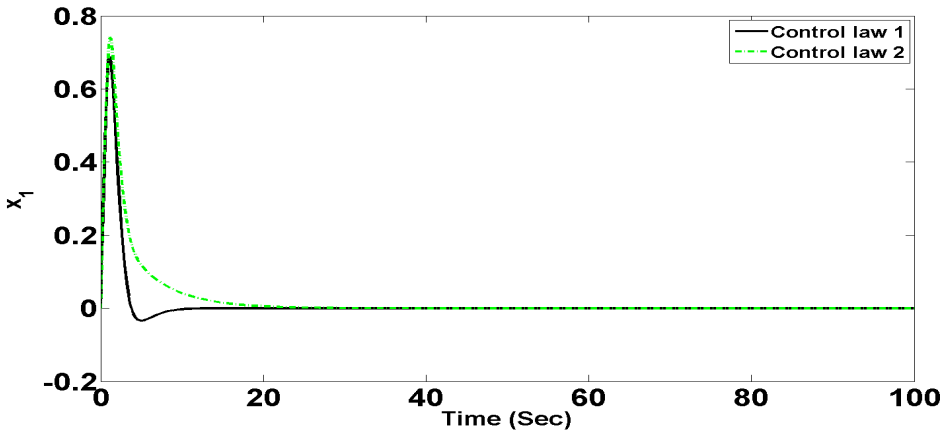

Fig 3. System state x_1 for control laws 1 and 2

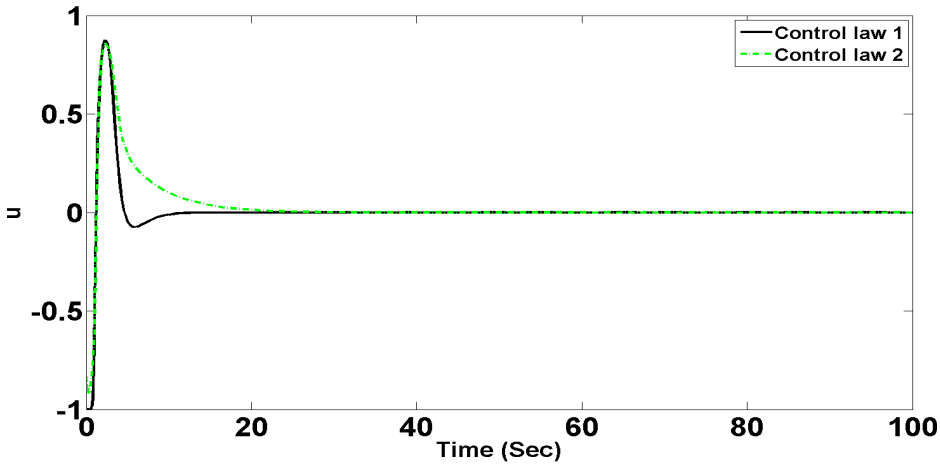

Fig 4. Control effort *u* for control laws 1 and 2

V. Conclusion

An adaptive algorithm converging to the optimal state feedback law for systems in the presence of input constraints was presented. Integral reinforcement knowledge was used for updating the value function. The presented approach is capable of learning the optimal policy without requiring knowledge of the drift dynamics. Using the experience replay technique for updating the value function, it was guaranteed under a simple condition on the richness of the recorded data, which can easily be checked online, is sufficient to guarantee exponential convergence to a near-optimal policy.

References

[1]Lewis, F. L., Vrabie, D., and Syrmos, V., *Optimal Control*, 3nd ed., Wiley, 2012.
[2]Sutton, R. S., and Barto, A. G., *Reinforcement learning – an introduction*, Cambridge, MA: MIT Press, 1998.
[3]Powell, W. B., *Approximate Dynamic Programming: solving the curses of dimensionality*, Wiley-Interscience, 2007.
[4]Bertsekas, D. P., *Dynamic Programming and Optimal Control: Approximate Dynamic Programming,* 4nd ed., MA: Athena Scientific, 2012.
[5]Howard, R. A., *Dynamic programming and markov processes.* Cambridge, MA: MIT Press, 1960.
[6]Beard, R. W., "Improving the Closed-loop Performance of Nonlinear Systems," Ph.D. Dissertation, Electrical Engineering Dep., Rensselaer Polytech Ins., Troy, New York, 1995.
[7]Abu-Khalaf, M., and Lewis, F. L., "Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach," Automatica, Vol. 41, 2005, pp. 779, 791.
[8]Doya, K., "Reinforcement Learning in Continuous-time and Space," Neural Computation, Vol. 12, No. 1, 2000, pp. 219, 245.
[9]Vamvoudakis, K., and Lewis, F. L., "Online Actor-critic Algorithm to Solve the Continuous Infinite-time Horizon Optimal Control Problem," Automatica, Vol. 46, 2010, pp. 878, 888.

[10]Murray, J. J., Cox, C. J., Lendaris, G. G., and Saeks, R., "Adaptive Dynamic Programming," IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev., Vol. 32, No. 2, 2002, pp. 140, 153.
[11]Vrabie, D., and Lewis, F. L., "Neural Network Approach to Continuous-time Direct Adaptive Optimal Control for Partially Unknown Nonlinear Systems," Neural Netw., Vol. 22, 2009, pp. 237, 246.
[12]Bhasin, S., "Reinforcement Learning and Optimal Control Methods for Uncertain Nonlinear Systems," Ph.D. Dissertation, Florida Univ, 2011.
[13]Vrabie, D., Pastravanu, O., Abu-Khalaf, M., and Lewis, F. L., "Adaptive Optimal Control for Continuous-time Linear Systems Based on Policy Iteration," Automatica, Vol. 45, No. 2, 2009, pp. 477, 484.
[14]Tsitsiklis, J. N., and Van Roy, B., "An Analysis of Temporal-Difference Learning with Function Approximation," IEEE Trans. Automatic Control, Vol. 42, 1997, pp. 674, 690.
[15]Wawrzynski, P., "Real-time Reinforcement Learning by Sequential Actor–critics and Experience Replay." Neural Netw., Vol. 22, 2009, pp. 1484, 1497.
[16]Dung, L. T., Komeda, T., and Takagi, M., "Efficient Experience Reuse in Non-Markovian Environments." *Proceeding of the Internatinal Conference Instrum*, Control Inf. Technol., Tokyo, Japan, 2008, pp. 3327–3332.
[17]Kalyanakrishnan, S., and Stone, P., "Batch reinforcement learning in a complex domain." *Proceeding of the 6th Internation Conference on Autoomus Agents and Multi-Agent Systms*, Honolulu, HI, pp. 650–657, 2007.
[18]Lin, L. J. , "Self-improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching." Machine Learning, Vol. 8, 1992, pp. 293, 321.
[19]Adam, S., Busoniu, L., and Babuska, R., "Experience Replay for Real-Time Reinforcement Learning Control." IEEE Trans. Syst. Man, Cybern., Part C: Appl. Rev., Vol. 42, 2012, pp. 201, 212.
[20]Chowdhary, G. V., "Concurrent learning for convergence in adaptive control without persistency of excitation," PhD. Dissertation, Georgia institute of technology, 2010.
[21]Chowdhary, G. V., and Johnson, E., "Concurrent Learning for Convergence in Adaptive Control without," IEEE CDC, Atlanta GA, 2010, pp. 3675-3679.
[22]Lyshevski, S. E., "Constrained optimization and control of nonlinear systems: New results in optimal control," *Proceeding of the IEEE Conference Decision and Control*, 1996, pp. 541–546.
[23]Finlayson, B. A., *The method of weighted residuals and variational principles*. New York: Academic Press, 1990.